

# SIMULATION-BASED EVALUATION OF RESERVATION MECHANISMS FOR THE TIME WINDOW ROUTING METHOD

Thomas Lienert

Florian Wenzler

Johannes Fottner

Chair of Materials Handling, Material Flow, Logistics

Department of Mechanical Engineering

Technical University of Munich

Boltzmannstrasse 15, 85748 Garching, Germany

Email: [thomas.lienert@tum.de](mailto:thomas.lienert@tum.de), [florian.wenzler@tum.de](mailto:florian.wenzler@tum.de), [j.fottner@tum.de](mailto:j.fottner@tum.de)

## KEYWORDS

Automated Warehouses, Mobile Robots, Time Window Routing Method, Discrete Event Simulation

## ABSTRACT

Automated warehouses operated by a fleet of robots offer great flexibility, since fleet size can be adjusted easily to throughput requirements. Furthermore, they provide higher redundancy compared to common solutions for automated storage and retrieval systems.

On the other hand, these systems require more complex control strategies to run robustly and efficiently. Special routing and deadlock handling strategies are necessary to avoid blocking and collisions among the robots.

In this contribution, we focus on the time window routing method, an approach for avoiding deadlocks by reserving routes in advance. We present and discuss different reservation mechanisms that are evaluated by the means of simulation.

## INTRODUCTION

Automated warehouses that are run with a fleet of mobile robots for the part-to-picker order picking have become the subject of intensive research since different decision problems need to be resolved during operation of these systems.

Basically, these systems consist of a rack system containing storage items and a fleet of robots moving within the storage area. The robots use a rectangular grid of paths to fulfil storage and retrieval requests. We refer to these systems as mobile-robot-based warehouses. There are several types of mobile-robot-based warehouses that differ in regard to their storage systems.

Robotic mobile fulfilment systems (RMFS) consist of a single storage tier, where items are stored on shelves on the ground. Robots travel underneath these shelves, lift them and bring them to the picking zone that is located somewhere near the storage area (Azadeh et al. 2018a).

In contrast, shuttle-systems consist of several tiers that are connected by lifts. These lifts link the storage system to the picking zone and enable vertical movements of the robots that are denominated as shuttles in this context (Tappia et al. 2018).

In another type, robots move horizontally as well as vertically within an aisle placed between two single-deep storage racks. Picking zones are located at one or at both ends of each aisle (Azadeh et al. 2018b).

Although these types differ slightly, they provide the same benefits relative to common stacker-crane-based storage systems. They are easily scalable. The whole system can theoretically be run with a single robot and if a higher throughput is needed, more and more robots can be added. The layout can be changed flexibly and the system can be enlarged easily. Furthermore, a required sequence can be established within the storage system (Lienert and Fottner 2018) and high redundancy can be provided as long as suitable failure-handling strategies are used (Lienert et al. 2019). In the literature, mobile-robot-based warehouses are widely discussed. Among others, storage assignment (Boysen et al. 2019), order batching (Boysen et al. 2017), dispatching (Yuan and Gong 2017), battery charging and swapping (Zou et al. 2017) as well as dwelling strategies for idle robots (Roy et al. 2016) are addressed.

In this contribution, we focus on the routing and deadlock-handling, more precisely on the time window routing method. This approach enables conflict-free routing of robots by reserving the path to be travelled in advance. Acceleration and deceleration processes are usually neglected when the time window routing method is applied. We present different reservation mechanisms that include acceleration and deceleration processes and fit different requirements regarding communication between the robots and the material flow control. The remainder of this paper is organized as follows. In the next section, we briefly introduce the time window routing method. We subsequently describe and discuss three different reservation mechanisms that are compared with a simulation and taking into account an RMFS before we conclude our work.

## TIME WINDOW ROUTING METHOD

In mobile-robot-based warehouses, robots move using the layout given by storage-aisles and cross-aisles. Since several robots are operating in the system at the same time, traffic must somehow be controlled and deadlocks or even worse, collisions must be avoided. In general a deadlock describes a situation where one or more

processes are blocked forever because the requests for resources by the processes can never be satisfied (Kim et al. 1997). In the context of routing robots, the processes correspond to the execution of the routes and the resources for the layout of segments along these routes. For example, a deadlock occurs if two robots driving in opposite directions meet each other within a storage aisle.

The approach of the time window routing method avoids deadlocks by reserving the path for a robot from its current situation to the destination in advance. On each layout segment that needs to be travelled along this path, a time window is blocked, during which the layout segment is claimed exclusively by a robot and during which the layout is not available for the movement of any other robot. Since time windows on neighbouring layout segments of a route overlap each other, robots can move safely through the layout.

To apply this method, the layout is represented by a graph. Each node corresponds to a layout segment, whereas the edges give information about predecessors and successors of the nodes. For each node, there is a time line with reserved and free time windows (figure 1).  $f_{i,l}$  denominates the  $l^{\text{th}}$  free time window on node  $i$ .

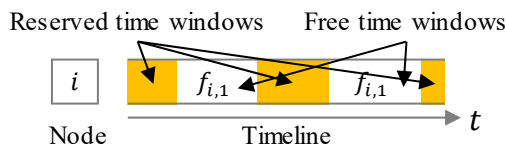


Figure 1: Reserved and free time windows on a node  $i$

In the case that a robot has to be routed from its current position to a given destination, the method searches for a route through the free time windows on the nodes using an A\*-algorithm. Once a conflict-free route is found, the corresponding time windows are reserved and the robot can start moving. From the point of time that the idea of this method was introduced first by Kim and Tanchocco (Kim and Tanchocco 1991), it has been applied in different contexts as the routing of automated guided vehicles in container terminals (Stenzel 2008) for the organization of taxi traffic at airports (Bussacker 2005) and for managing a fleet of robots in an RMFS (Hvězda et al. 2018). For more detailed insights into the modelling of the layout as a graph we refer to (Lienert and Fottner 2017).

The time window routing method can be applied in systems that use centralized material flow control as well as in systems that are operated by a multi agent system.

When it comes to the execution of a reserved route, deadlocks or collisions among the robots threaten to occur even though the routes are theoretically conflict-free. Robots might be delayed – due to several reasons – and not match their reserved time windows. Therefore it is essential that the node’s crossing order of the robots based on the conflict-free schedule is maintained (Maza and Castagna 2005). Hence, a robot is only allowed to travel the next node along its reserved route if it has reserved the next time window on that node.

Figure 2 shows an example that clarifies this approach. At timestamp  $T$  the reserved time window of robot  $r_2$  begins on node  $j$ . Robot  $r_1$  is delayed. According to the planning, it should reside on node  $i$ , but it has not yet passed node  $j$ ,  $k$  and  $l$ .

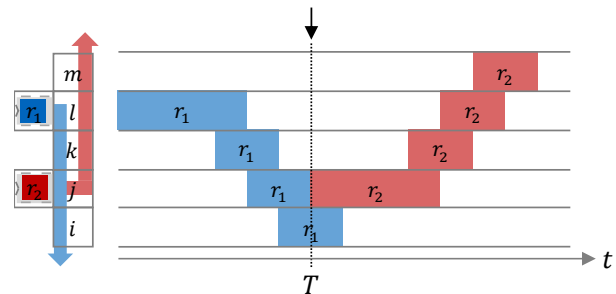


Figure 2: Robot  $r_1$  is delayed and a deadlock might occur

If robot  $r_2$  nevertheless continues with the execution of its route, both robots will face each other somewhere in between the nodes  $j$  and  $l$  and a deadlock will occur.

Note that robots must also not enter a node before their reserved time window has started. In the example in figure 3, robot  $r_2$  has just entered node  $j$  at timestamp  $T$  and before the corresponding time window has started. Routing robot  $r_1$  will lead to a feasible route traversing nodes  $l$ ,  $k$  and  $j$  before the reservation of robot  $r_2$  begins (as in figure 2). Once again, a deadlock is likely to happen.

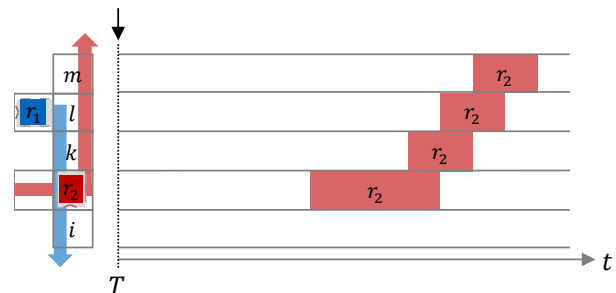


Figure 3: Robot  $r_2$  is early and a deadlock might occur

Maintaining the node’s crossing order is easily realizable if acceleration and deceleration processes are neglected since a robot can stop immediately in case it is not allowed to enter the next node. However, if acceleration and deceleration processes are taken into account, the robust execution of a route must be implemented with some sort of lookahead.

In a previous work, we modified the time window routing method to incorporate acceleration and deceleration processes. During the planning, so-called “segments” are created that describe movement of a robot over several nodes in a straight line. The computed route is executed segment by segment, respecting the node’s crossing order (Lienert et al. 2018a). Figure 4 shows the creation of a segment during the planning phase. Starting with a free time window on the node  $i$ , the algorithm checks whether the free time window on node  $j$  is reachable. In that case, the segment is – if possible – extended node by node.

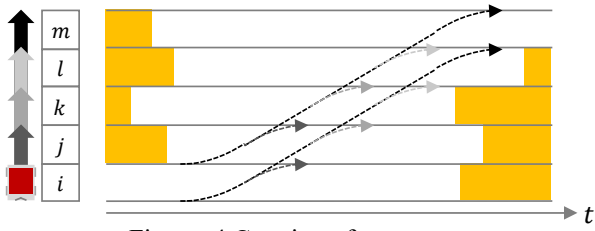


Figure 4: Creation of a segment.

To extend a segment, none of the time windows within the segment can violate any existing reservation. In the example in figure 4, all free time windows are reachable since no trajectory touches any existing reservation. Each trajectory is described by a pair of arrows that represent the movements of the front and rear of a robot. Free time windows that are reachable are candidates for the first node of a new segment in a later iteration. For a comprehensive description of the overall routing procedure, we refer to (Lienert et al. 2018a).

### RESERVATION MECHANISMS

In this section, we focus on the part of the algorithm that examines the reachability of free time windows from a free time window at a specific time stamp, taking into account three different reservation mechanisms. These mechanisms differ on the one hand in regard to the requirements of the communication between robots and centralized material flow control. On the other hand, the required length of the reserved time windows of a segment differ, which leads to varying resource utilizations.

#### Mechanism 1: Triangle

First, we assume that once a robot starts with the execution of a segment, it has to be ensured that the robot can finish that segment without any interference due to other robots that are late. In settings where communication between robots and central material flow control cannot be guaranteed at any given time in real-time, this procedure is necessary to avoid collisions and deadlocks.

The reservations of a segment are preponed so they start with the departure of the robot from the first node of the segment. Time windows are deleted as soon as a robot has left a node completely. As a result, reservations of a segment form a triangle (see figure 5).

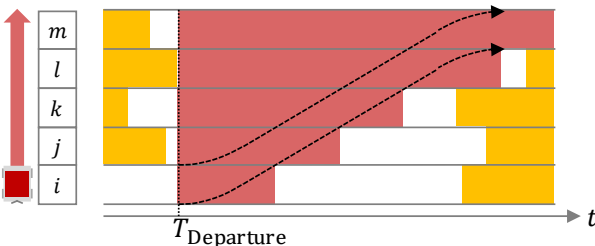


Figure 5: Reservations of a segment display the form of a triangle

We assume free time window  $f_{i,l}$  to be the first time window of a segment and free time window  $f_{k,p}$  the free time window whose reachability is analyzed (figure 6).

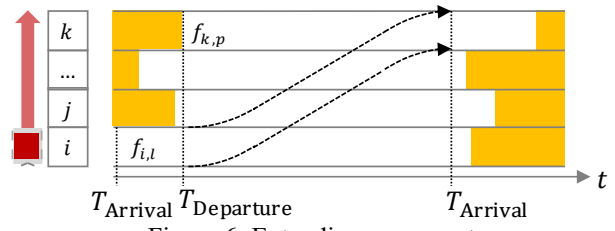


Figure 6: Extending a segment

The flowchart in figure 7 shows the procedure of checking the reachability of a free time window (FTW) in detail.

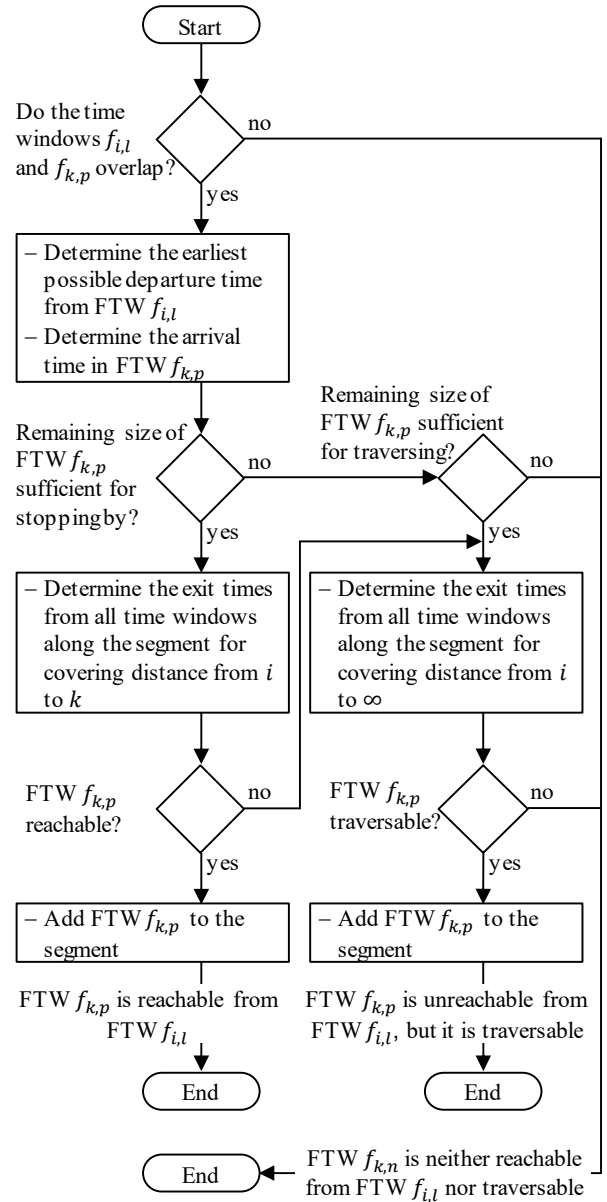


Figure 7: Reachability check of free time window  $f_{k,p}$

First of all, it is mandatory that both of the free time windows  $f_{i,l}$  and  $f_{k,p}$  overlap each other. In that case, the earliest departure time from free time window  $f_{i,l}$  can be determined. The departure cannot take place before the robot arrives on node  $i$  and not before any of the free time windows of the segment including the free

time window  $f_{k,p}$  start. Next, the arrival time in free time window  $f_{k,p}$  can be calculated taking into account the distance as well as the robot's parameters.

The remaining size of free time window  $f_{k,p}$  after the arrival must be sufficient to leave the node completely before the next reservation starts. In that case, all of the required reservations on all nodes of the segment can be determined. If no existing reservation on any node of the segment is violated, the free time window  $f_{k,p}$  is reachable and the segment can be extended. In a later iteration, free time window  $f_{k,p}$  serves as a starting time window of another segment.

If the remaining size of free time window  $f_{k,p}$  is not sufficient for stopping by, node  $k$  might be traversed by the robot before the next reservation starts. In that case, all required time windows on all nodes of the segment can once again be determined assuming a movement of infinite length. If no existing reservation on any node of the segment is violated, free time window  $f_{k,p}$  is traversable and the segment can be extended (as in figure 8).

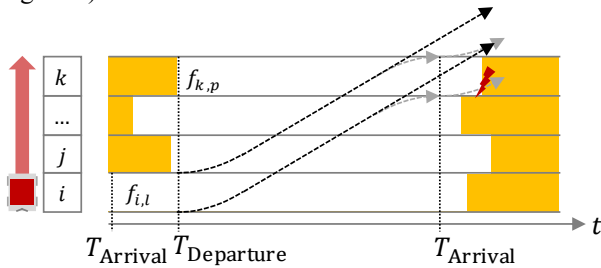


Figure 8: Remaining size of free time window  $f_{k,p}$  does not allow an intermediate stop on node  $k$ , but traversing the node.

In case the remaining size of the free time window is sufficient for stopping by but a previous reservation is violated, traversing the node without violating any reservation might once again be possible (as in figure 9).

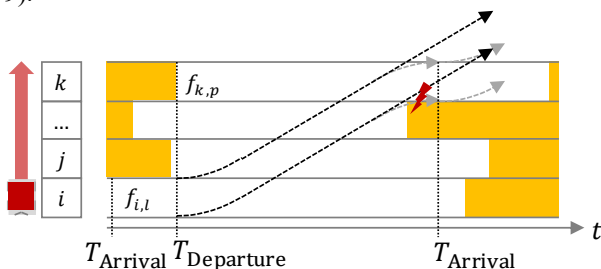


Figure 9: An existing reservation does not allow an intermediate stop on node  $k$ , but does allow traversing the node.

Note that in both cases (figure 8 and figure 9), free time window  $f_{k,p}$  is not a potential candidate for the first time window of another segment.

During execution of a reserved route, a robot is only allowed to start a segment if the robot is not early and if the robot is to travel all nodes of the segment next, taking into account the node's crossing order.

## Mechanism 2: Rectangle

The second reservation mechanism is even more restrictive. Time windows are deleted only after a segment is finished and the robot comes to a standstill. As a result, reservations of a segment form a rectangle (see figure 10).

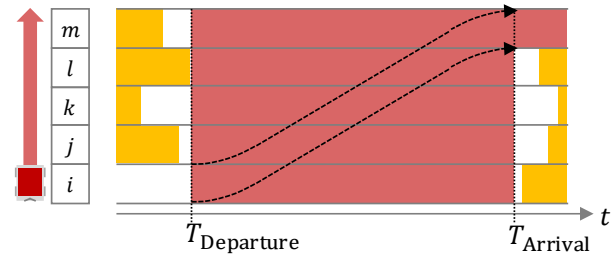


Figure 10: Reservations of a segment display the form of a rectangle

This approach is suitable for applications that do not allow the deletion of time windows in real-time. Once a robot starts with the execution of a segment, no information of the robot's position is available until the robot finishes the segment.

The procedure to check reachability corresponds to the flow chart in figure 7 apart from determining the necessary reservations. These once again begin with the departure, but end with the arrival on the last node of the segment.

## Mechanism 3: Stairs

The third reservation mechanism enables a more efficient use of resources but requires communication between robots and centralized material flow control more often. Time windows do not begin at the departure, but there is a buffer before each reservation that enables a safety deceleration in case a robot with a previous reservation is delayed. As a result, reservations of a segment form a stairs pattern (see figure 11).

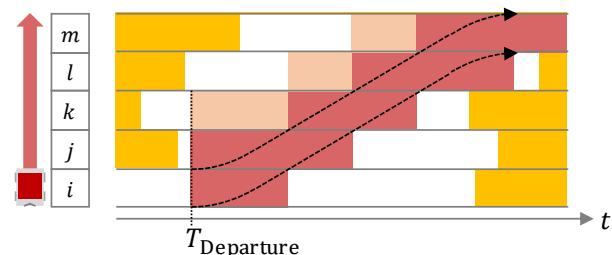


Figure 11: Reservations of a segment display the form of a stairs pattern.

To analyze reachability, the procedure differs slightly and is shown in the flow chart in figure 14. Note that it is not necessary that both free time windows  $f_{i,j}$  and  $f_{k,p}$  overlap each other.

First of all, the start of the reservation of the free time window  $f_{k,p}$  has to be determined. Starting from the penultimate node of the segment on which the robot has to stop in case a preceding robot is delayed, the

distances between the nodes are added up until the deceleration distance is reached. Let the node  $a$  be the node in which this summed distance reaches or exceeds the deceleration distance. The reservation of the free time window  $f_{k,p}$  must begin as soon as the stopping position of the node  $a$  has been passed (figure 12).

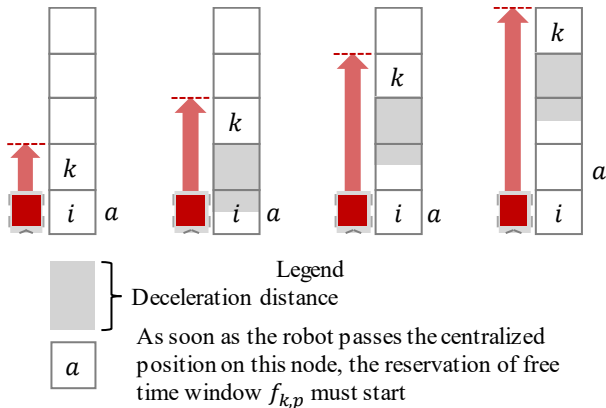


Figure 12: Determination of the start of the reservation of the free time-window  $f_{k,p}$

If the calculated start does not fall in the free time-window, it is checked whether it is possible to postpone the departure, so that the start coincides with the start of the free time window. Otherwise the free time window  $f_{k,p}$  is not reachable.

The arrival time can be determined next. The remainder of the procedure corresponds to the one described in the flowchart in figure 7 apart from the determination of the necessary reservations. For each free time window in the segment, the start of the reservations needs to be recalculated (as described above) since the departure from the first free time window  $f_{i,j}$  of the segment might have been shifted. Reservations end as soon as a robot has completely left a node.

In case a robot that reserved a preceding time window on any node of a segment is delayed, the additional buffer allows the robot to decelerate and to stop safely. In the example in figure 13, robot  $r_1$  is delayed at timestamp  $T$  and its reserved time window on node  $m$  is not yet deleted, robot  $r_2$  is not allowed to enter node  $m$  and starts decelerating, coming to a standstill on node  $l$ .

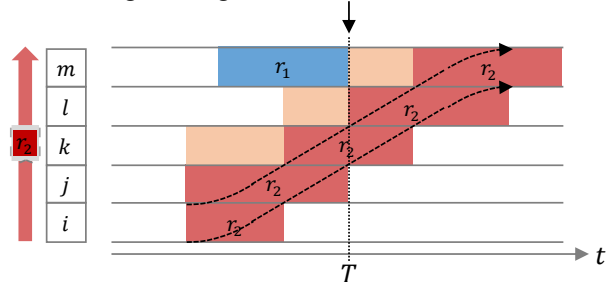


Figure 13: Robot  $r_1$  being late leads to an unplanned intermediate stop on node  $l$  for robot  $r_2$ .

Note that if acceleration and deceleration processes are neglected, additional buffers disappear.

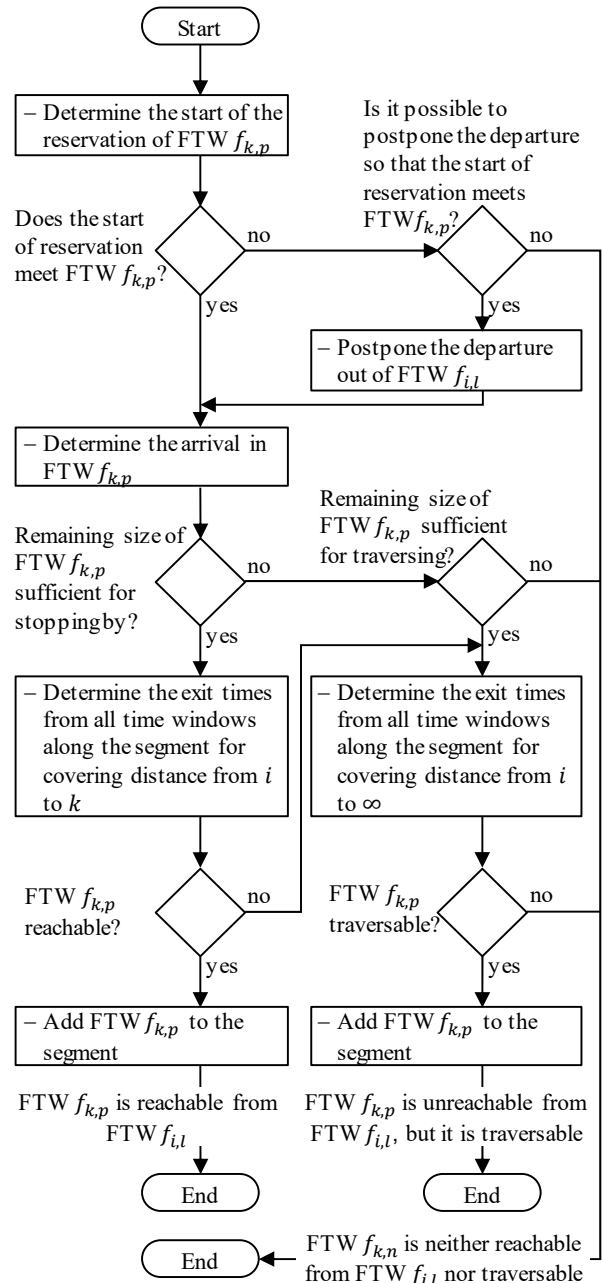


Figure 14: Reachability check of free time window  $f_{k,p}$

Theoretically there is another reservation mechanism where reservations start with an additional buffer but end with the arrival on the last node of the segment. However, we assume that if real-time communication is enabled (which is necessary for the stairs mechanism), time windows can be deleted immediately after a node has been left.

## SIMULATION STUDY

In this section, we compare the previously described reservation mechanisms by performing a simulation study, considering an RMFS.

There is a trade-off regarding the segment length using the triangle and rectangle reservation mechanisms. Short segments entail frequent stops and accelerations of the robots. Longer segments enable robots 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. As a consequence, the optimal maximal segment length has to be determined for each number of robots. This is done first before the mechanisms are compared to each other.

### Considered System

We apply the strategies to a fleet of robots moving within an RMFS with 336 storage locations that are arranged in seven double rows divided by storage aisles. There are two cross-aisles located at one third and at two thirds of the aisle length. All aisles can be used for bi-directional traffic. There are four picking areas with five picking places, with each arranged in front of the storage system. In front of these places, there are two unidirectional cross-aisles. A replenishment area, where empty racks are refilled, is located on the opposite side of the storage area. Robots are dedicated to a picking zone and perform three different cycles to maintain the material flow between storage locations, picking area and replenishment area. For a more detailed description of the system, we refer to (Lienert et al. 2018b). We implemented the RMFS using the Tecnomatix Plant Simulation discrete event simulation environment.

### Parameter Settings

We vary the number of robots, starting with four robots (one for each picking zone) and increasing this up to 60 robots working in the system in steps of four, and repeat the experiments for each reservation mechanism. With both the triangle and rectangle reservation mechanisms, we limit the segment length to a certain number of nodes. We start with a maximum segment length of only one node, meaning that robots only move node by node, stopping at every single node along their routes. We vary the maximum segment length increasing it up to 30 nodes. All of the remaining parameters, such as the robot's acceleration and maximum speed, remain the same. Simulation time is set to 24 hours. No warm-up time is taken into account, as the goal of the simulation study is to compare the reservation mechanisms. We conduct five replications for each parameter setting.

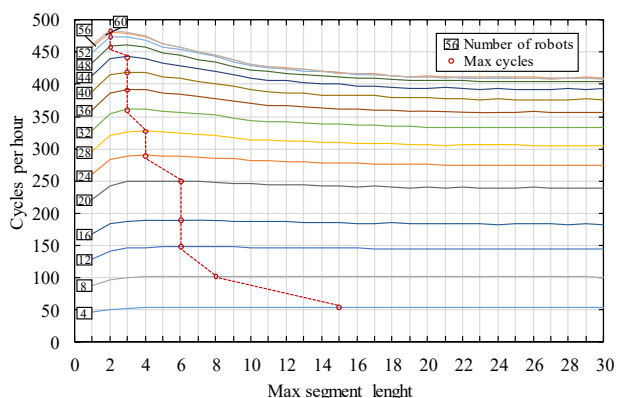


Figure 15: Throughput using the triangle reservation mechanism

### Results

First we analyse the trade-off between shorter and longer segments. Figure 15 shows the throughput reached with the corresponding parameter setting using the triangle reservation mechanism.

As can be seen, the more robots that are operating in the system, the shorter the optimal segment length becomes. Figure 16 shows the throughput reached using the rectangle reservation mechanism.

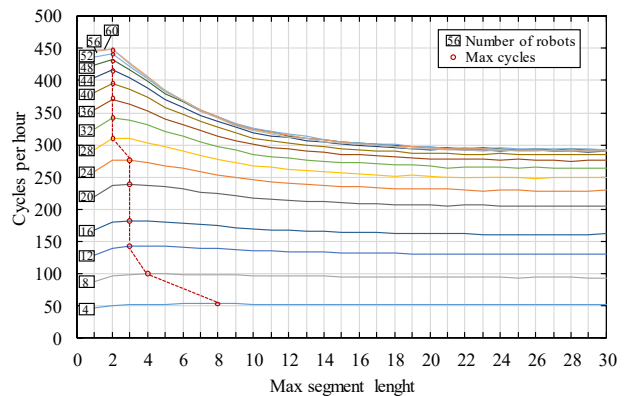


Figure 16: Throughput using the rectangle reservation mechanism

A similar behavior can be observed. However, the optimal maximum segment length is even shorter. That is expectable, since nodes are reserved for a longer time periods than using the triangle mechanisms.

Finally we compare these maximums to the stairs reservation mechanisms. Figure 17 shows the throughput reached.

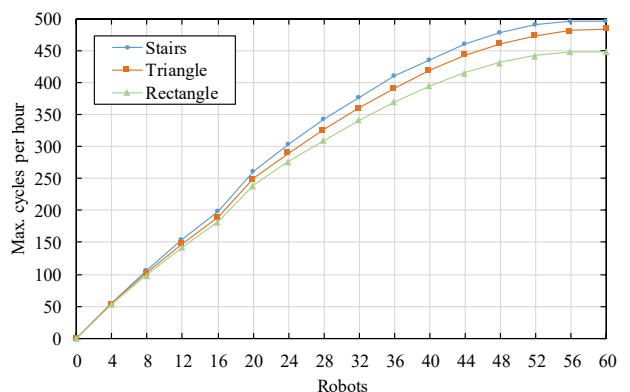


Figure 17: Throughput for varying number of robots

All curves look similar with a linear increase and a small knee between 16 and 20 vehicles before reaching saturation with 56 robots. Since each robot is assigned to a certain picking zone, the number of robots per picking zone with 20 robots equals the number of picking places, and a different strategy is used for the supply of the picking zone (see Lienert et al. 2018b).

For a small number of robots, the throughput of the different mechanisms is equal. In fact, using a single robot in the whole system yields the same throughput, no matter which reservation mechanism is used.

However, the more robots that are operating in the system, the more evident the difference becomes. As expected, the stairs mechanism reaches the highest throughput followed by the triangle one. Reaching a throughput of 400 cycles per hour using the stairs mechanism requires 36 robots, 40 robots using the triangle mechanism and as many as 44 robots using the rectangle mechanism.

Note that saturation is reached at different levels. Using the stairs reservation mechanism yields the highest throughput, whereas the triangle mechanism reaches 97.6% and the rectangle mechanism 90.4% of that throughput.

## CONCLUSION

In this contribution, we considered mobile-robot-based warehouses. We presented three different reservation mechanisms for the time window routing method. These mechanisms require different levels of communication and differ regarding the resource utilization while executing a calculated route. We conducted a simulation study to compare the performance of these mechanisms considering an RMFS. As expected, the stairs mechanism enables the highest throughput followed by the rectangle mechanism.

In our consideration, robots accelerate, move with constant maximum speed or decelerate. For future work, we suggest enabling robots to move with a constant, but reduced speed to avoid intermediate stops. This behavior must be modelled and taken into account for the analysis of reachability of free time windows.

## REFERENCES

- Azadeh, K.; de Koster, R. and Roy, D., 2018 (a), "Robotized and Automated Warehouse Systems: Review and Recent Developments." *Transportation Science* 53 No.4, 917–945.
- Azadeh, K.; Roy, D.; De Koster, R., 2018 (b): "Design, Modeling, and Analysis of Vertical Robotic Storage and Retrieval Systems." *Transportation Science* 53 No.5, 1–22.
- Boysen, N.; Briskorn, D. and Emde, S., 2017, "Parts-to-picker based order processing in a rack-moving mobile robots environment." *European Journal of Operational Research* 262, No.2, 550-562.
- Boysen, N.; de Koster, R. and Weidinger, F., 2019, "Warehousing in the e-commerce era: A survey." *European Journal of Operational Research* 227 No.2, 396–411.
- Busacker, T. 2005. *Steigerung der Flughafen-Kapazität durch Modellierung und Optimierung von Flughafen-Boden-Rollverkehr – Ein Beitrag zu einem künftigen Rollführungssystem*. Dissertation. Technische Universität Berlin.
- Havězda, J.; Rybecký, T.; Kulich, M. and Přeučil, L., 2018, "Context-Aware Route Planning for Automated Warehouses." In *Proceedings of the 21st International Conference on Intelligent Transportation Systems*, 2955-2960.
- Kim C. W. and Tanchoco J. M. A., 1991, "Conflict-free shortest-time bi-directional AGV routing." *International Journal of Production Research* 29, No.12, 2377-2391.
- Kim C. W., Tanchoco J. M. A. and Koo P., 1997 "Deadlock Prevention in Manufacturing Systems with AGV Systems: Banker's Algorithm Approach." *Journal of Manufacturing Science and Engineering* 119, No.4, 849-854.
- Lienert, T. and Fottner, J., 2017, "Development of a generic simulation method for the time window routing of automated guided vehicles." *Logistics Journal: Proceedings*, Vol. 2017.
- Lienert, T.; Wenzler, F. and Fottner, J., 2018 (a), "Robust integration of acceleration and deceleration processes into the time window routing method." In *Proceedings of the 9th International Scientific Symposium on Logistics*, 66-86.
- Lienert, T.; Staab, T.; Ludwig, C. and Fottner, J., 2018 (b), "Simulation-based Performance Analysis in Robotic Mobile Fulfillment Systems." In *Proceedings of the 8th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, 383-390.
- Lienert, T. and Fottner, J., 2018, "Routing-based Sequencing Applied to Shuttle Systems." In *Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2949-2954.
- Lienert, T., Stigler, L. and Fottner, J., 2019, "Failure-handling strategies for mobile robots in automated warehouses." In *Proceedings of the 33rd European Conference on Modelling and Simulation*, 199-205.
- Maza, S. and Castagna, P., 2005, "A performance-based structural policy for conflict-free routing of bi-directional automated guided vehicles." *Computers in Industry* 56, No.7, 719-733.
- Roy, D.; Krishnamurthy, A.; Heragu, S. and Malmborg, C., 2015, "Queuing models to analyze dwell-point and cross-aisle location in autonomous vehicle-based warehouse systems", *European Journal of Operational Research* 242, No. 1, 72-87.
- Stenzel, B. 2008. *Online Disjoint Vehicle Routing with Application to AGV Routing*. Dissertation. Technische Universität Berlin.
- Tappia, E.; Roy, D. de Koster, R. and Melacini, M., 2018, "Modeling, Analysis, and Design Insights for Shuttle-Based Compact Storage Systems", *Transportation Science* 51, No.1.
- Yuan, Z.; Gong, Y.Y., 2017, "Bot-In-Time Delivery for Robotic Mobile Fulfillment Systems." *IEEE Transactions on Engineering and Management* 64, No.1, 83-93.
- Zou, B.; Xu, X.; Gong, Y.Y., de Koster, R., 2017, "Evaluating battery charging and swapping strategies in a robotic mobile fulfillment system." *European Journal of Operational Research* 267, No. 2, 733-753.

**THOMAS LIENERT** has been working as a research assistant at the Chair of Materials Handling, Material Flow and Logistics, Technical University of Munich, since 2014. His research deals with the simulation of mobile-robot-based warehouses. His email address is: thomas.lienert@tum.de.

**FLORIAN WENZLER** has been working as a research assistant at the Chair of Materials Handling, Material Flow and Logistics, Technical University of Munich, since 2013. His research deals with the optimization of project plans in construction. His email address is: florian.wenzler@tum.de.

**JOHANNES FOTTNER** is professor and head of the Chair of Materials Handling, Material flow, Logistics at the Technical University of Munich. His email address is: j.fottner@tum.de.