

# Routing-based Sequencing Applied to Shuttle Systems

Thomas Lienert, Johannes Fottner  
 Chair of Materials Handling, Material Flow, Logistics  
 Technical University of Munich, Germany  
 {lienert, fottner}@fml.mw.tum.de

**Abstract**—Shuttle systems are automated storage and retrieval systems based on autonomous vehicles moving within a storage-rack system. Various shuttle system configurations exist, which differ by the vehicles’ movement axes. Some configurations allow stored units to be retrieved in a given sequence directly from the storage system. Supplying picking areas in customer-order sequence or production areas in production sequence exemplify the requirement to supply storage items in a particular sequence.

In this contribution, we apply a routing-based algorithm to shuttle systems, which ensures that routed vehicles arrive in a desired sequence. We conduct a series of simulation experiments to analyze the impact of retrieval-in-sequence on throughput and discuss measures to mitigate the ensuing loss of throughput.

**Keywords**—shuttle systems, automated storage and retrieval systems, routing, sequencing, discrete event simulation

## I. INTRODUCTION

Shuttle systems are autonomous vehicle-based storage and retrieval systems (AVS/RSs) used to store small unit loads for supplying picking or production areas based on the goods-to-person principle [1]. According to Malmborg, shuttle systems are characterized by horizontally operating vehicles, which are moving within a rail system on every storage tier, completing storage and retrieval transactions [2]. They offer greater throughput and better scalability than conventional stacker-crane-based automated storage and retrieval systems (AS/RSs) [3].

Different system configurations have evolved over the course of recent developments. These can mainly be divided into configurations with captive vehicles and those with roaming vehicles [4]. A single vehicle moves linearly along every storage aisle on every storage tier in systems with captive vehicles. A lift transports storage units from the input/output (I/O) location to the various tiers, where they are handed over to a buffer location. The vehicles load the storage unit from that buffer and bring it to the designated storage location. Retrieval proceeds vice versa. By contrast, the lifts move the vehicles vertically in systems with roaming vehicles. A vehicle can therefore operate on different storage tiers within the same aisle. Fewer vehicles are needed to serve all storage locations. These types of vehicles are also known as tier-to-tier vehicles.

Degree of freedom	Characteristics			
Change of an aisle	not possible		possible	
Change of a tier	not possible		possible	

Configuration	aisle-captive tier-captive	aisle-to-aisle tier-captive	aisle-captive tier-to-tier	aisle-to-aisle tier-to-tier
Movement axes	x	x / y	x / z	x / y / z

Fig. 1. Shuttle system configurations resulting from the vehicles’ movement space

Another configuration allows vehicles to not only change storage tier, but to change the storage aisle on a given storage tier. The vehicles can turn 90 degree and use cross-aisles, which are orthogonal to the storage racks. Figure 1 provides an overview of the four system configurations that emerge from the vehicles’ different movement spaces. The x-axis coincides with the storage aisle, the y-axis with the lifts, and the z-axis with the cross-aisles.

Systems with tier- and aisle-captive vehicles are the most common. These configurations attain the greatest throughput, as the vertical and horizontal movements are completely decoupled from each other. Since every shuttle operates within its own rail system, it needn’t share resources with the others. In contrast, several vehicles move within the same rail system in configurations deploying tier-to-tier and aisle-to-aisle vehicles (figure 2). The main characteristic of a system configured this way is that every shuttle can reach every storage location within the system. The system can therefore operate with a single vehicle. More vehicles can be added to achieve greater throughput, if needed. Should a single vehicle fail, those still operating continue responding to storage and retrieval requests and the system’s throughput is only slightly affected. A disadvantage of these configurations is that systems incorporating them require more complex control strategies to run robustly and efficiently. As every vehicle can reach every storage location, storage and retrieval requests need to be assigned to vehicles. Traffic on the storage tiers needs to be managed to avoid congestion, blocking, and collisions among vehicles moving within a shared rail system.

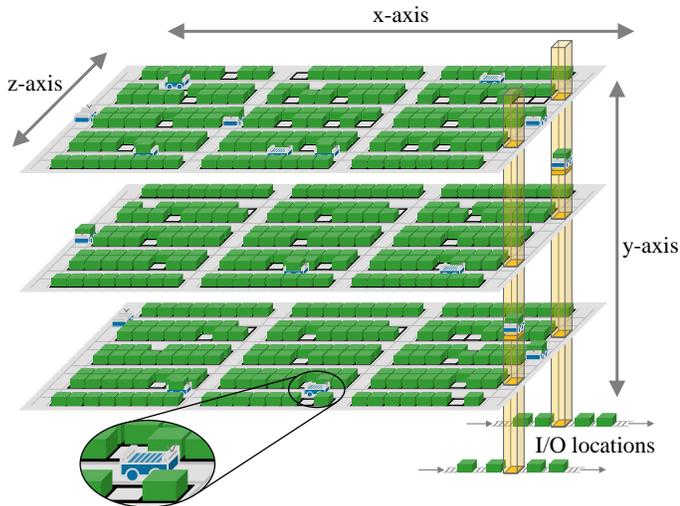


Fig. 2. Example of a system with tier-to-tier and aisle-to-aisle vehicles

In systems with aisle-to-aisle vehicles, every storage unit can be moved to every connected lift and therefore be provided at every I/O location. This enables retrieval in a desired sequence directly from the storage system. No additional conveyance technology within the pre-storage area is required. In this paper, we focus on systems with aisle-to-aisle and tier-to-tier vehicles. However, the presented approach is generic; it can also be applied to systems with tier-captive vehicles.

This contribution extends our previous work as follows: We describe an improved algorithm for routing vehicles that must arrive at a destination in a given sequence, compared to the inverse routing presented in [5]. We therefore use the time window routing method, which provides efficient deadlock-free routes for shuttles on every storage tier. We apply the generic approach presented in [6], extended by the robust integration of acceleration and deceleration processes of the vehicles within the model as explained in [7]. Finally, we use the routing-based sequencing algorithm for retrieval-in-sequence with shuttle systems, modelling the storage system as [8] describes in detail.

## II. PROBLEM STATEMENT

AS/RSs are warehousing systems used to store and retrieve products in both distribution and production environments [9]. The need to retrieve stored items in a given sequence diminishes throughput [10].

Figure 3 explains how that throughput is lost. It provides a schematic layout of a conventional stacker-crane-based AS/RS (left), which could also be interpreted as a shuttle system with aisle-captive vehicles. Retrieval is realized aisle by aisle. In the pre-storage area, a conveyor loop connects different aisles and points of use. Two independent sequences (sequence *A* and sequence *B*) need to be provided at two different points of use (marked as *Seq A* and *Seq B* in figure 3). We denote the place where the storage units must be in the correct sequence as the sequence point (SP). The storage units' order can no longer be manipulated after the sequence point.

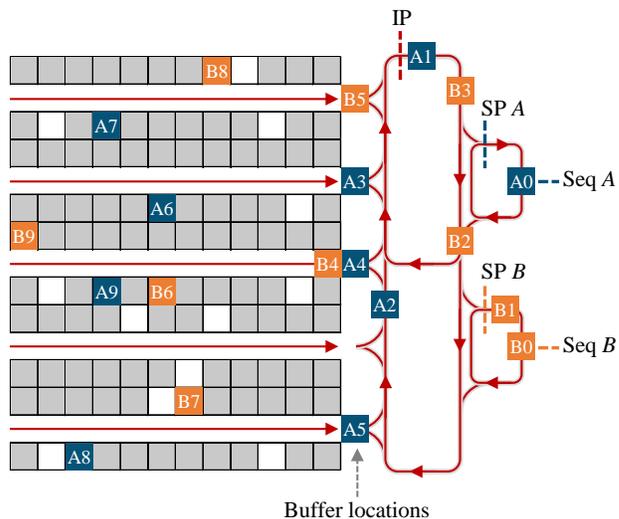


Fig. 3. Retrieval-in-sequence with a stacker-crane-based AS/RS

Storage units within a sequence are labeled with increasing sequence numbers. Once a retrieval request is released, the corresponding stacker-crane retrieves the storage unit from its storage location and hands it over to a buffer location between storage aisle and conveyor loop. The storage unit waits there until it is allowed to enter the conveyor loop. Before the latter happens, the storage unit with the preceding sequence number must have passed the identification point (IP), where a scanner detects it and sends its sequence number to the material-flow computer. The buffer can provide space for several storage units. However, sooner or later waiting storage units will fill it if its size is feasible, preventing the stacker-crane from working. The consequence is decreased throughput.

Although sequences are independent, they do affect each other. For instance, as sequence number *B4* has been retrieved from the same storage aisle as sequence number *A4*, the former has to wait until sequence number *A4* has been released before it can enter the conveyor loop. This delays all of the following sequence numbers within sequence *B*. In a worst case scenario, the sequences permanently block each other, and a deadlock emerges, which order-releasing management needs to avoid (figure 4).

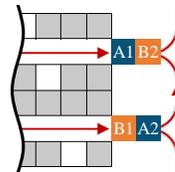


Fig. 4. Two independent sequences permanently block each other; a deadlock arises that has to be avoided.

Geinitz investigated the impact of retrieval-in-sequence with a stacker-crane-based AS/RS. Dependencies among sequences being hard to determine analytically, he conducted a simulation study and showed that considerably more throughput is lost than for chaotic retrieval (up to 30%). However, this decreases when fewer aisles are involved and more buffer locations are available in front of the aisles [10].

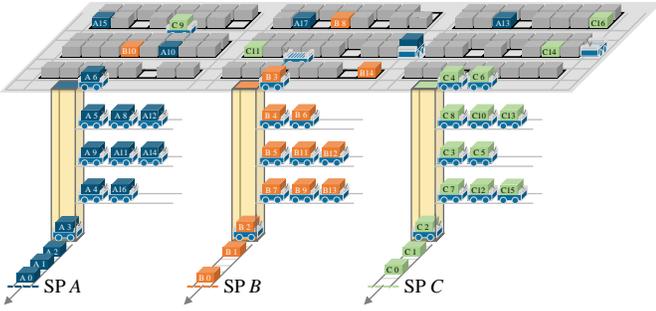


Fig. 5. Shuttle system with three lifts and three independent sequences

Establishing sequence within shuttle systems featuring aisle-to-aisle vehicles is a two-stage process. First, the sequence numbers retrieved from a single storage tier need to be provided in ascending order on that storage tier at the lift, which supplies a point of use. As different routes to reach a lift exist, vehicles can overtake each other establishing the desired tier-based sequence. Second, that lift moves waiting sequence numbers on different tiers to the I/O location to establish the final sequence. After having finished a transportation task, the material-flow computer will search for the next following sequence number to transport the carrying vehicle to the I/O location, where the storage unit is handed over. Figure 5 shows an example with three lifts, each supplying a point of use. Hence there are three independent sequences. In the sequel, we refer to a vehicle loaded with a storage unit with respective sequence number by the sequence number itself.

Most of the literature concerning shuttle systems deals either with performance analysis of different layout configurations, for instance identifying ideal rack geometry [11], or with developing control algorithms as in [12], where lifts with two independent platforms are scheduled. For evaluation, both analytical models, generally based on queuing networks [13], and simulation-based approaches [14] can be found. Apart from our previous work, [5], no further investigation analyzes the impact of sequencing with shuttle systems, although this problem often arises in operations. Applications are the supply of picking locations in sequence of customer orders, the supply of production areas in the production sequence, and the supply of gates in sequence of the delivery order of trucks.

Below, we briefly introduce the time window routing method and describe an optimal routing-based approach that ensures a given sequence at the destination. We implement the algorithm in a simulation environment, conduct a series of simulation experiments to quantify the loss of throughput caused by the retrieval-in-sequence with shuttle systems, and show how that loss can be reduced.

### III. ROUTING-BASED SEQUENCING ALGORITHM

Traffic management must not only calculate routes taken by vehicles to reach their destinations, but also handle deadlocks. One possibility for avoiding deadlocks is routing based on time windows. This concept was first introduced by Kim and Tanchocco [15] and subsequently used to route vehicles robustly and efficiently in different applications such

as aircraft taxi traffic at airports [16], automated guided vehicles within container terminals [17] and warehouses [18] or in general agents moving over an infrastructure [19].

To apply the time window routing method, the infrastructure needs to be modelled as a graph. The nodes of this graph represent layout sections; the edges only represent the successor relations. For each node, the algorithm maintains a list of free time windows through which vehicles can be routed (figure 6).

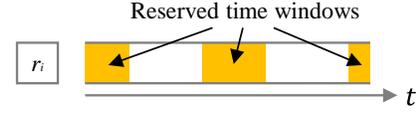


Fig. 6. Reserved and free time windows on node  $r_i$

By applying an  $A^*$  algorithm, the routing procedure yields the fastest path from the start node to the destination node through the free time windows (figure 7) as long as the heuristic for estimating the remaining travel time is consistent [20]. Note that the whole route will be calculated in advance before a vehicle starts traveling it. We refer to [6] and [7] for a detailed and formal description of the routing procedure.

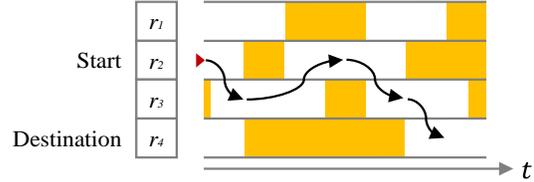


Fig. 7. Fastest path from start to destination.

To ensure a desired sequence at a destination toward which several vehicles are routed, the algorithm checks the correctness of that sequence whenever the search through free time windows reaches a free time window on a destination node. The algorithm therefore maintains a list containing the routed sequence numbers and their arrival times at the sequence point for every sequence point. If a preceding sequence number arrives in a reserved time window later than the currently examined free time window, then the sequence is incorrect, the goal conditions are not satisfied, and the search through free time windows continues.

Figure 8 provides an example of such a situation. The lift is accessed from node  $r_2$ , which is the sequence point for the tier-based sequence. The search for the route for sequence number  $A4$  reaches a free time window on that node. Since sequence number  $A3$  arrives in a later reserved time window, the search must continue.

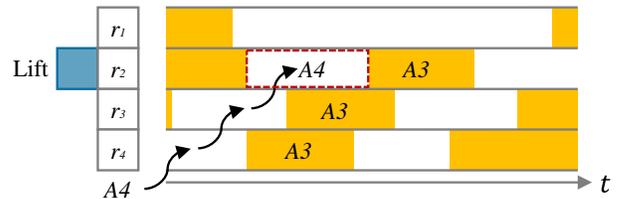


Fig. 8. The first free time window on node  $r_2$  does not lead to a correct sequence.

In the example, the algorithm will return to node  $r_2$ , but within a later free time window as shown in figure 9. As in the former free time window, the sequence was incorrect. The vehicle will pass this node and wait on node  $r_1$  until the next free time window on node  $r_2$  begins. This time the sequence is correct and the goal conditions fulfilled. Node  $r_1$  can be seen as a buffer location next to the lift.

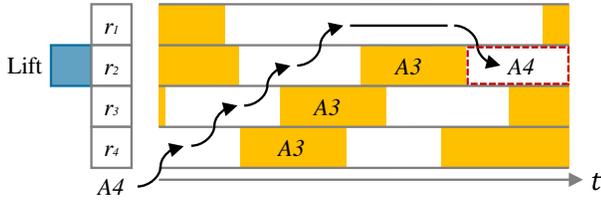


Fig. 9. Arriving in the second free time window on node  $r_2$  guarantees the correct sequence.

Time window routing provides the fastest possible route that preserves the desired sequence, taking into account the prevailing reservations of vehicles already routed. In concluding, we would like to emphasize that the routing procedures must be executed in the same sequence as that of the vehicles' intended arrival at the sequence point.

#### IV. SIMULATION STUDY

We conduct a simulation study using the *Tecnomatix Plant Simulation* discrete-event simulation environment to quantify the loss of throughput that retrieval-in-sequence with shuttle systems causes. We use several types of nodes, such as nodes representing sections of storage aisles, sections of cross aisles, and crossings, to model a storage tier. [8] provides details about modeling the shuttle system.

We investigate a shuttle system comprising ten storage tiers each with 2400 storage locations, which are arranged along twelve storage aisles. There are two cross-aisles on each tier for changing the storage aisle: one between the lifts and the storage racks, the other in the middle of the storage aisles. There are three lifts, each supplying a single point of use, hence there are three independent sequences. Ten retrieval tasks are initially generated for each point of use and gathered into a pool. Storage tiers and locations from which units have to be retrieved are chosen randomly. Whenever a storage unit has been handed over at the I/O location, the retrieval request with the oldest timestamp in the pool is assigned to the vehicle and a new retrieval request is generated for the point of use that has been served. Furthermore, vehicles work constantly in double cycles. After unloading the storage unit, another storage unit is loaded, which needs to be stored in the system. An empty storage location is therefore chosen randomly on the same tier as that on which the next retrieval takes place.

As soon as a loaded vehicle arrives at a lift, it registers as waiting vehicle and a transportation task for the lift is generated. If retrieval is chaotic, the lift completes transportation tasks using the FIFO method. If retrieval is in sequence, then transportation tasks are completed in ascending order of sequence numbers.

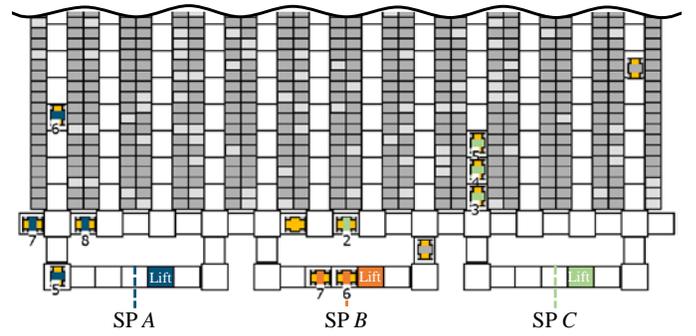


Fig. 10. Screenshot of a storage tier in the simulation model. Every white rectangle represents a node in the underlying layout graph. The marked sequence points refer to the tier-based sequence.

Figure 10 shows a screenshot of a part of a storage tier and the connections to the lifts in the simulation model. As mentioned, sequence numbers need to be provided in ascending order at the corresponding lifts, which are entered from the left-hand side and leftward toward the right-hand side. Sequence numbers  $A_7$  and  $A_8$  can be observed waiting at their current locations in sequence A until approaching sequence number  $A_6$  has passed them.

We assume that throughput lost by retrieval-in-sequence depends on the number of vehicles operating in the system. We therefore vary the numbers of vehicles in increments of two until reaching maximum throughput, which is bounded by constantly working lifts. We conduct a retrieval-in-sequence and a chaotic retrieval with each number of vehicles. We run five iterations of twelve simulation hours each.

Figure 11 shows the throughput, measured in double cycles per hour, evolving with increasing number of vehicles for chaotic retrieval and retrieval-in-sequence. Almost no loss of throughput is detectable for up to ten vehicles. Since the vehicles are spread across the ten storage tiers, interference between them is relatively small. More vehicles engender greater loss of throughput; however, the gap closes as soon as chaotic retrieval reaches saturation.

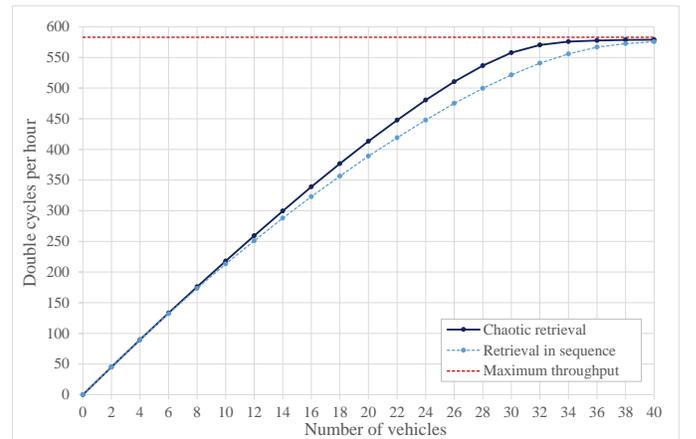


Fig. 11. Throughput obtained by chaotic retrieval and by retrieval-in-sequence

The chart not only shows the loss of throughput caused by the retrieval-in-sequence, it also gives information about how adding more vehicles can reduce the loss. For instance, 22 vehicles are necessary to reach a throughput of 450 double cycles per hour with chaotic retrieval. Retrieval-in-sequence needs two more vehicles to reach the same throughput.

Another option for reducing the loss of throughput is to deploy buffers in front of the sequence points. Prematurely arriving sequence numbers would be buffered until their correct position within the sequence is reached. Since establishing the sequence is a two-stage process, there are two options for placing those extra buffers: an additional sequence buffer on each storage tier close to the lift and sequence buffers on the I/O level. On the storage tier shown in figure 12, sequence number A6 arrives at the lift before sequence number A5 does. Hence it will be held in the sequence buffer next to the lift. The described routing procedure completely organizes this process, as shown in the example of figure 8 and figure 9.

On the I/O level, there is a sequence buffer with two places, which can be accessed individually. Sequence number A3 is waiting in one of these places until sequence number A2, which the lift is currently transporting, has passed.

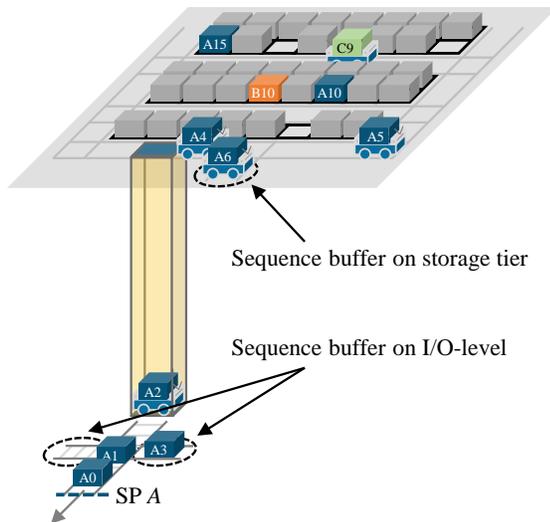


Fig. 12. Options for placing sequence buffers

A sequence buffer on the I/O level allows a lift to transport a higher sequence number if the next sequence number is not yet waiting on a storage tier. The flowchart in figure 13 describes the procedure for selecting a lift's next transportation task. If the transportation task with the next sequence number is not yet available, the lift will complete the transportation task with the smallest available sequence number if any transportation task is available and there is still capacity in the sequence buffer.

We conduct another series of simulation experiments and analyze the following scenarios: A layout with an extra buffer on each tier next to each lift and a layout with an extra buffer having one or two places, which can be accessed individually on the I/O level (as shown in figure 12).

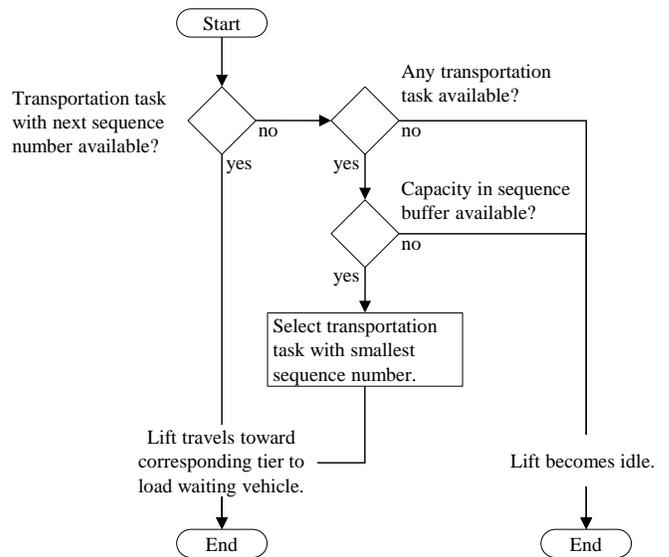


Fig. 13. Procedure for selecting the next transportation task for a lift

Figure 14 shows the associated throughput. Using sequence buffers on the storage tier doesn't influence throughput. Hence, only the lifts cause loss of throughput. If the next sequence number is still unavailable, then the lift is prevented from working and becomes idle, although other vehicles on other tiers are already waiting. If the additional buffer is used, the lift can continue completing transportation tasks with higher sequence numbers. Using a single sequence buffer on the I/O level significantly reduces loss of throughput, whereas using two buffer locations nearly allows the throughput obtained by chaotic retrieval to be reached.

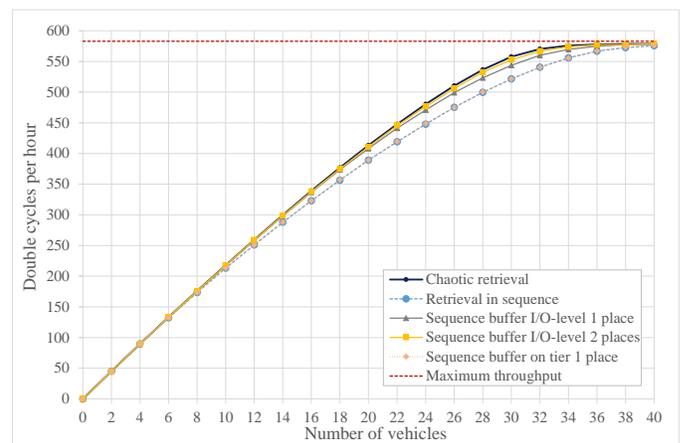


Fig. 14. Throughput obtained by chaotic retrieval, by retrieval-in-sequence, and with options using sequence buffers

The chart in figure 15 provides a closer look at the loss of throughput. Loss of throughput rises nearly linearly and peaks with 24 to 28 vehicles. As can be seen, adding one additional buffer on the I/O level reduces the maximum loss of throughput from nearly 7% to 2.5%. Adding two additional buffer locations limits the loss of throughput to under 1% regardless of the number of vehicles operating within the system.

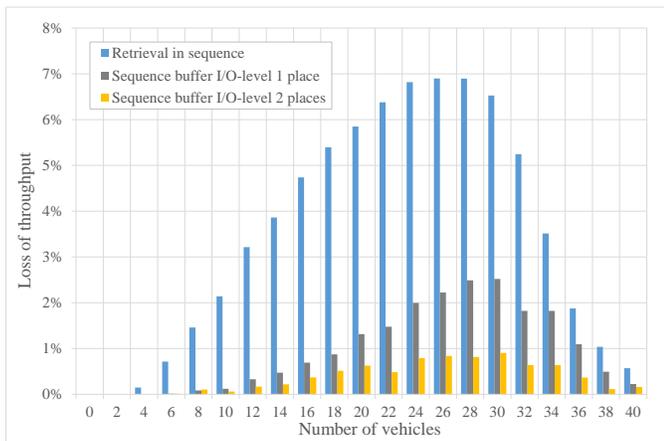


Fig. 15. Loss of throughput compared to chaotic retrieval

In summary, the loss of throughput caused by retrieval-in-sequence depends on the number of vehicles operating in the system. Regarding the considered layout, it climbs to 7% of the throughput obtained by chaotic retrieval. However, this loss can be eliminated by adding more resources (vehicles) or by providing a sequence buffer comprising two places, which require additional space in the pre-storage area.

## V. CONCLUSION

In this paper, we applied a routing-based sequencing algorithm to shuttle systems with tier-to-tier and aisle-to-aisle vehicles. This routing algorithm is based on time windows and identifies the fastest path from the current location to a destination at which the vehicles must arrive in a given sequence. The algorithm enables retrieval-in-sequence directly from the shuttle system. Hence storage items can be provided in a desired sequence at the point of use without using further conveyor technology.

We conducted a series of simulation experiments and gained the following insights:

- Retrieval-in-sequence with shuttle systems incurs a loss of throughput similar to what has been observed while investigating stacker crane-based AS/RSs.
- The loss of throughput depends on the number of vehicles operating in the system.
- Using more vehicles can reduce loss of throughput.
- Using sequence buffers on the I/O level can reduce loss of throughput.

If loss of throughput has to be reduced, then adequate measures taking costs and space into account must be individually initiated. However, if retrieval-in-sequence is necessary only temporary, it is reasonable to take advantage of the flexibility of shuttle systems with tier-to-tier and aisle-to-aisle vehicles and use more vehicles. In contrast, if retrieval-in-sequence is a permanent requirement, the use of sequence buffers is advisable.

As the throughput lost by retrieval-in-sequence not only depends on the number of vehicles, but also on the storage

systems' layout, we suggest varying the number of tiers, aisles, cross-aisles, and lifts to obtain further information about the impact of sequencing on shuttle systems.

## REFERENCES

- [1] VDI guideline 2692 part 1, "Automated vehicle storage and retrieval systems for small uni loads," Berlin:Beuth, 2015.
- [2] C. J. Malmberg, "Conceptualizing tools for autonomous vehicle storage and retrieval systems," *International Journal of Production Research*, vol. 40, no. 8, pp. 1807–22, 2002
- [3] G. Kartnig, B. Grösel, N. Zrnica, "Past, state-of-the-art and future of intralogistics in relation to megatrends," *FME Transactions*, vol. 40, pp. 193–200, 2012
- [4] S. S. Heragu, C. Cai, A. Krishnamurthy and C. J. Malmberg, "Analytical models for analysis of automated warehouse material handling systems," *International Journal of Production Research*, vol. 49, no. 22, pp. 6833–61, 2011
- [5] T. Lienert and W. Günthner, "Development and simulation-based evaluation of an algorithm for the retrieval-in-sequence for shuttle systems," in *Proceedings of the European Modeling and Simulation Symposium*, pp. 15–24, September 2016.
- [6] T. Lienert and Fottner, "Development of a generic simulation method for the time window routing of automated guided vehicles," *Logistics Journal: Proceedings*, vol. 2017, 2017.
- [7] T. Lienert and Fottner, "Robust integration of acceleration and deceleration processes into the time window routing method, in *Proceedings of the 9th International Scientific Symposium on Logistics*, in press.
- [8] T. Lienert and Fottner, "No more deadlocks—applying the time window routing method to shuttle systems," in *Proceedings of the 31st European Conference on Modelling and Simulation*, pp. 169–75, May 2017.
- [9] K. J. Roodbergen and I. F. A. Vis, "A survey on literature on automated storage and retrieval systems," *European Journal of Operations Research*, vol. 194, no. 2 pp. 343–62, 2009
- [10] J. Geinitz, "Unerkannte Abhängigkeiten mindern die Leistungs-fähigkeit automatisierter Lager", in *Marktbild Lager* no. 18, pp. 16–8.
- [11] M. Eder and G. Kartnig, "Throughput analysis of S/R shuttle systems and ideal geometry for high performance," in *Proceedings of the XXI International Conference on Material Handling, Construction and Logistics*, pp. 193–8, 2015
- [12] H. J. Carlo and I. F. A. Vis, "Sequencing dynamic storage systems with multiple lifts and shuttles," *International Journal of Production Economics*, vol. 140, no. 2, pp. 844–53, 2012
- [13] D. Roy, A. Krishnamurthy, S. Heragu and C. Malmberg, "Queuing models to analyze dwell-point and cross-aisle location in autonomous vehicle-based warehouse systems", *European Journal of Operational Research*, vol. 242, no. 1, pp. 72–87, 2015
- [14] T. Lerher, Y. B. Ekren, Z. Sari and B. Rosi, "Simulation analysis of shuttle based storage and retrieval systems," *International Journal of Simulation Modelling* vol. 14, no. 1, pp. 48–59, 2015
- [15] C. W. Kim and J. M. A. Tanchocco, "Conflict-free shortest-time bi-directional AGV routing," *International Journal of Production Research*, vol. 29, no. 12, pp. 2377–91, 1991
- [16] T. Busacker, "Steigerung der Flughafen-Kapazität durch Modellierung und Optimierung von Flughafen-Boden-Rollverkehr," Ph. D. dissertation, Technische Universität Berlin, 2005
- [17] B. Stenzel, "Online Disjoint Vehicle Routing with Application to AGV Routing," Ph. D. dissertation, Technische Universität Berlin, 2008
- [18] N. Smolic-Rocak, S. Bogdan, Z. Kovacic and T. Petrovic, "Time Windows Based Dynamic Routing in Multi-AGV Systems", in *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 1, pp. 151–155, 2010
- [19] A. W. ter Mors, "The world according to MARP," Ph. D. dissertation, Technische Universiteit Delft, 2010
- [20] S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach", Global Edition, Pearson Education Limited, 2016