Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo and M. Loper, eds.

SCHEDULING AND CONTROLLING MULTIPLE VEHICLES ON A COMMON TRACK IN HIGH-POWERED AUTOMATED VEHICLE STORAGE AND RETRIEVAL SYSTEMS

Andreas Habl
Andrei Evseev
Johannes Fottner

Chair of Materials Handling, Material Flow, Logistics
Technical University of Munich
Boltzmannstraße 15
85748 Garching bei München, GERMANY

ABSTRACT

The ongoing trend towards increased throughput capacity and scalability of automated warehouses is pushing the application of rail-guided automated vehicle storage and retrieval systems (AVSRSs). By deploying more vehicles on each tier and lifting system, the performance and adaptability of these systems can be further increased. However, the transformation into high-powered AVSRSs requires advanced algorithms to run the system in a robust and efficient manner. In this work, an approach to schedule and control multiple vehicles on a common track by using a tier of a high-powered AVSRS is presented. By running computational experiments, it can be shown that the developed algorithm enables a collision-free and fast execution of transport tasks. By deploying several vehicles on a tier, the completion time of the transport tasks can be decreased and, hence, the throughput capacity of the tier can be increased.

1 INTRODUCTION

Automated vehicle storage and retrieval systems (AVSRSs), also known as shuttle systems, represent a scalable and powerful way to store small unit loads and to supply picking or manufacturing areas on the basis of goods-to-person principle. This warehouse technology leverages the separation of horizontal and vertical transportation of units such as containers, boxes or trays in order to achieve high throughput capacities. Hence, vehicles are restricted to a single tier or lifting system to execute transport tasks in horizontal and vertical direction, respectively (FEM 2017).

In the case of storage, a lift vehicle of the lifting system picks up the transport unit (TU) to be stored at the input position and transports it to the respective tier. After laying the TU at the transfer buffer, a shuttle vehicle of the tier picks it up and moves the TU to the storage location. For a retrieval, the operations are reversed and the TU arrives at the output position. In order to carry out storage and retrieval jobs in an AVSRS, usually one shuttle vehicle operates at each tier and one lift vehicle moves in each lifting system. However, to further increase throughput capacity as well as to provide enhanced scalability and adaptability of new or already existing AVSRSs, more than one vehicle on each tier or lifting system could be deployed (Figure 1). This transformation into a high-powered AVSRS requires extended control algorithms to achieve a robust and efficient execution of transport tasks on tiers and in lifting systems. To ensure robust transportation, the vehicles have to be coordinated in order to prevent block events and collisions. Beyond that, the applied scheduling of the vehicles needs to minimize waiting times as well as driving routes, thus making the transportation efficient.

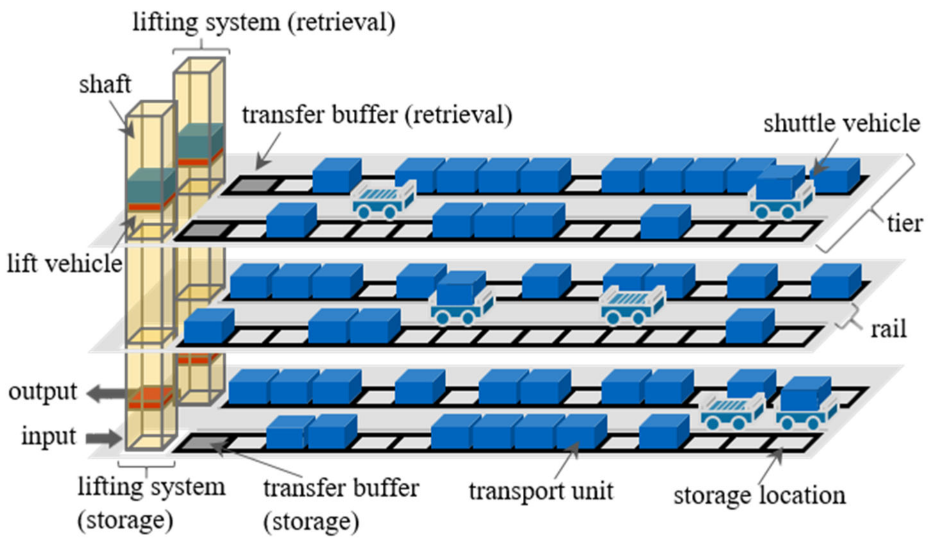

Figure 1: Segment of a high-powered AVSRS with two lift vehicles per lifting system as well as two shuttle vehicles per tier.

The resulting lift and shuttle vehicle scheduling problem is comparable to the crane scheduling problem, which has already been discussed in literature. It can be regarded as a special case of general machine scheduling and, thus, be assigned to NP-hard problems. The problem can be stated as follows: n transport tasks must be processed, whereby m vehicles are available. In this sense, the tasks have to be assigned to the vehicles with the goal to minimize the total completion time of the tasks. For this optimal assignment of tasks, the optimal trajectories of all vehicles have to be determined. However, since in an AVSRS lifting systems and tiers are interconnected via transfer buffers for TUs to be stored or retrieved, the interaction between these dynamic systems has to be considered, dynamically adapting the schedule and control of vehicles on a tier or lifting system to incoming transport tasks. Another feature of AVSRSs is to follow a given order of task execution when sequenced retrievals of TUs have to be provided.
This paper investigates the vehicle scheduling problem that occurs in a high-powered AVSRS with multiple lift and shuttle vehicles that operate on the same shaft and rail, respectively, by generalizing the problem to multiple vehicles that move along a common track. Therefore, the developed approach consists of a scheduling part, i.e., assigning transport tasks to the vehicles, as well as a controlling part, i.e., planning the trajectories of the vehicles. The developed algorithm takes the dynamic interaction of vertical and horizontal transportation into account and allows for retrievals in sequence. In order to demonstrate the feasibility and efficiency of the approach to tackle the mentioned challenges, a tier of a high-powered AVSRS is exemplarily observed.

The remainder of the paper is organized as follows. In Section 2, related work on scheduling and controlling AVSRS as well as on similar problems in other logistics applications is presented. Section 3 describes the proposed algorithm to assign the tasks to the vehicles and to plan the trajectories for execution. In Section 4, a series of computational experiments is carried out to evaluate the efficiency of the algorithm and to analyze different layout configurations. Section 5 presents and discusses the obtained results of the experiments. Finally, in Section 6, a conclusion of this contribution and an outlook for future research is provided.

2 RELATED WORK

In a high-powered AVSRS, multiple lift or shuttle vehicles move on a common track, i.e., in a shaft or on a rail, to execute transport jobs. Within vertical transportation, Carlo and Vis (2012) schedule two vehicles in a common shaft of a lifting system via a block-sequencing algorithm that optimizes the sequence of the transport jobs with the help of heuristics. Zhao et al. (2018) supplement this approach by taking into consideration the acceleration and deceleration of the vehicles. Habl et al. (2019) apply several block-sequencing algorithms to schedule up to five vehicles in a common shaft and evaluate the efficiency and robustness by conducting a simulation study.

For horizontal transportation, Habl et al. (2020) present different approaches to coordinate multiple vehicles that move along the same rail on a tier by introducing the shuttle vehicle scheduling problem. A block-sequencing algorithm is applied to schedule up to five vehicles on a common rail, and it is evaluated by simulation. By integrating relocation strategies, Habl et al. (2019) extend the application of the block-sequencing algorithm to multi-deep storage racks. By applying a dynamic control of multiple vehicles on a common rail, Habl et al. (2020) present and analyze a control algorithm based on reservation of sections within the tier and priority rules for job allocation. When it comes to the application of AVSRSs with unit lifts in practice, usually only one vehicle operates on each tier and lifting system. Nevertheless, if several vehicles move along the same rail or shaft in practice, the vehicles mostly are restricted to an allocated zone in order to simplify the control.

Besides AVSRSs, multiple vehicles with non-crossing constraints can be found in a variety of other logistics applications. Kung et al. (2014) propose an algorithm to schedule multiple stacker cranes in an automated storage and retrieval system (ASRS) by using dynamic programming. Kress et al. (2019) also present a dynamic programming algorithm and a related beam search heuristics to schedule two gantry cranes that move along a common rail. Peterson et al. (2014) present an algorithm to schedule several factory cranes on a common track by using a block-sequencing method and heuristics. To increase the efficiency of elevators, Takahashi et al. (2003) deploy two vehicles in a common shaft. In order to avoid collisions between vehicles, a zoning approach is applied and the optimal partitioning is identified by utilizing a genetic algorithm. Similarly, Ishihara and Kato (2013) apply a zoning approach that dynamically assigns a zone to each vehicle. Beyond that, Valdivielso and Miyamoto (2011) present and evaluate a control strategy based on a scheduling algorithm without zoning. In the field of robotics, Erdoğan et al. (2014) present heuristics and optimal approaches to schedule two robots on a common rail by introducing the twin robots scheduling problem.

The scheduling and controlling of several vehicles that move along a common track occurs in many applications such as AVSRSs, ASRS, cranes, elevators, and robots. In the field of AVSRSs, different algorithms have been developed to accomplish safe and fast execution of transport jobs either in a lifting system or on a tier. However, in order to holistically optimize the transport processes in a high-powered AVSRS and to push the application of such systems in practice, the interaction between lifting systems and tiers as well as retrievals in sequence need to be considered by the control. The other logistics applications have already been discussed in literature. But, since in AVSRSs tiers and lifting systems are interconnected, existing solutions of other isolated applications cannot be completely adapted. Hence, this contribution extends the investigations on the control of a high-powered AVSRS from a holistic perspective by developing an approach that is able to integrate the interface between tier and lifting system and that enables sequenced retrievals.

3 ALGORITHM

The overall objective of the algorithm is to provide an assignment of a block of jobs to the vehicles that enables the execution of the jobs in a preferably small amount of time. Therefore, different assignments are investigated by searching the solution space. For every assignment, the target coordinates of every vehicle are determined and, subsequently, trajectories for every vehicle are planned to reach the targets. By means of the trajectories, the completion time for a block of jobs can be computed. Thus, the assignments can be compared to each other.

3.1 Task Assignment

Each job contains a task to move to the pickup position of a TU and a task to move the TU to the delivery position. Basically, an assignment is composed of pairs (task, vehicle) that describe which task has to be executed by which vehicle. Those pairs are ordered to set the priority of each task, which defines the sequence of the tasks to be performed, allowing for a proper order of pickup and delivery as well as sequencing of TUs for retrieval. The best assignment is the one that enables the smallest completion time of all tasks by the available vehicles. In this context, each vehicle receives a set of trajectories that are divided in parts with uniformly accelerated movement. On the basis of the computed trajectories and, therefore, completion times for any assignment, the best assignment with the smallest completion time can be searched. This discrete optimization problem can be formulated as an assignment A with

$$A = \begin{pmatrix} A_1 \\ \dots \\ A_N \end{pmatrix} = \begin{pmatrix} T_1 & V_1 \\ \dots & \dots \\ T_N & V_N \end{pmatrix}$$

where N is the total number of tasks T that are assigned to vehicles V. The assignment also includes the priority through the order of the tasks.

3.2 Coordinate Determination

In any assignment, each task has been assigned to a vehicle and it has a certain priority, which defines the order of execution. When an assignment is forwarded to execution, the tasks are processed one by one according to the priority. As a consequence, at each point in time a task is prioritized and it is executed by a vehicle, whereas all the other vehicles have to avoid interferences with the prioritized vehicle. In order to minimize driving routes, the avoidant vehicles keep as close to their next task destinations as possible and, thus, at each priority every vehicle has a target coordinate to which it moves. To determine the target coordinates, two different operation modes of the vehicles are distinguished. Every vehicle that still has pending tasks for execution operates in fast mode, which means that the vehicle needs to move to the target coordinate as fast as possible. If a vehicle has already finished its assigned tasks, the vehicle operates in lazy mode, which means that it waits as long as possible.

As an example, Figure 2 shows the computation of target coordinates for the two vehicles R (red circle) and G (green rectangle) and one assignment (2 R, 1 G, 3 G, 3 R), where the number represents the task coordinate and the letters stands for the vehicle. At the beginning (Initial), the two vehicles are located at the initial coordinates 1 and 2 and operate in fast mode. It is iterated over all priorities. At priority 1, task 2 R is forwarded and vehicle R instantly gets the coordinate of its current task as a target coordinate (marked with a black dot). At priority 2, task 1 G is forwarded and vehicle G receives its target coordinate 1 at the current priority. The target coordinates for previous priorities (marked with a white dot) are computed backwards. Since there is no interference with vehicle R, vehicle G simply stays at coordinate 1. At priority 3, task 3 G is forwarded and vehicle G receives the target coordinate 3. At priority 4, task 3 R is forwarded and vehicle R receives the target coordinate 3. Since an interference between the vehicles is possible, the coordinates are constrained by the already computed target coordinates. The resulting target coordinates are (1, 1, 3, 2) for vehicle G and (2, 4, 4, 3) for vehicle R.

3.3 Trajectory Planning

Each movement to a target coordinate contains a period of constant acceleration, maximum speed, and constant deceleration. Depending on the current position and speed of the vehicle, these periods differ from each other and some periods may be absent, respectively. The time needed to move to a target coordinate is denoted as t_{move}.

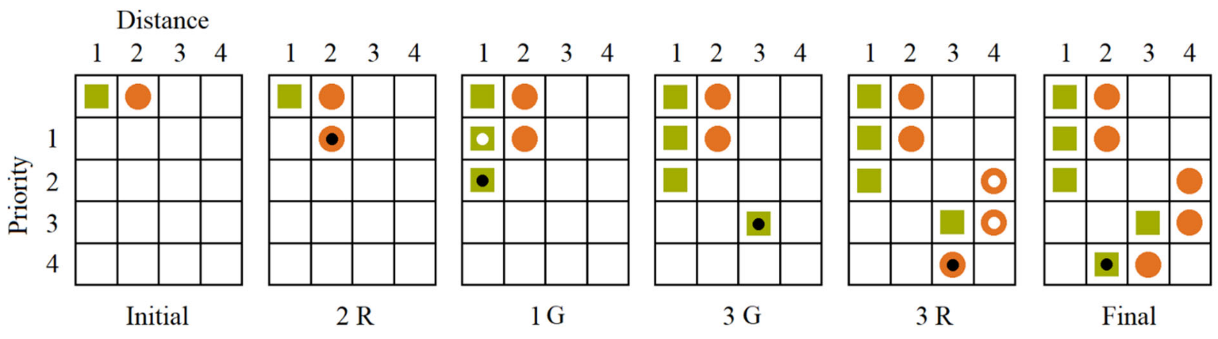

Figure 2: Determination of target coordinates of two vehicles for an exemplary assignment.

If the vehicle is able to reach the maximum speed, the time t_{move} consists of the time needed to reach the maximum speed t_{accel}, the time needed to move with the maximum speed t_{const} and the time needed to stop t_{decel}.

$$t_{move} = t_{accel} + t_{const} + t_{decel} = \frac{v_{max} - |v|}{a_{max}} + \frac{s_{const}}{v_{max}} + \frac{v_{max}}{a_{max}}$$

In case the vehicle only reaches an intermediate speed v_{int}, t_{move} consists of an acceleration and deceleration period.

$$t_{move} = t_{accel} + t_{decel} = \frac{v_{int} - |v|}{a_{max}} + \frac{v_{int}}{a_{max}}$$

If the vehicle needs only to stop, t_{move} consists just of a deceleration period.

$$t_{move} = t_{decel} = \frac{|v|}{a_{max}}$$

Based on the initial and computed target coordinates for every vehicle at each priority, the trajectories are planned by looping over all priorities. For each vehicle, the fastest trajectory to the target coordinate is constructed and loading or unloading parts are appended. Collisions between vehicles are avoided due to the way the target coordinates are constructed. As an example, Figure 3 shows the planning of the trajectories for the two vehicles R and G and the assignment (2 R, 1 G, 3 G, 3 R) from above. The solid parts correspond to loading or unloading processes, the dotted parts to an empty vehicle, and the dashed parts to a vehicle that transports a TU. It is iterated over all priorities, again. At priority 1, the prioritized task is at coordinate 2 and it is performed by vehicle R. Since vehicle R is already at its target coordinate, it can perform the task (solid line). Vehicle G waits (dotted line) as it is also at its target coordinate (see Figure 3a). At priority 2, the prioritized task is at coordinate 1 and it is performed by vehicle G. Since vehicle G has already been at the target coordinate, the waiting part is replaced by the loading part (see Figure 3b). At priority 3, the prioritized task is at coordinate 3 and it is performed by vehicle G, so it needs to move to coordinate 3 and, then, can perform its task. Vehicle R moves to its target coordinate 4 and waits (see Figure 3c). At priority 4, the prioritized task is at coordinate 3 and it is performed by vehicle R, so it needs to move to coordinate 3 and then can perform its task. Vehicle G moves to its target coordinate 2 and waits. The job block is finished and the completion time can be computed (see Figure 3d).

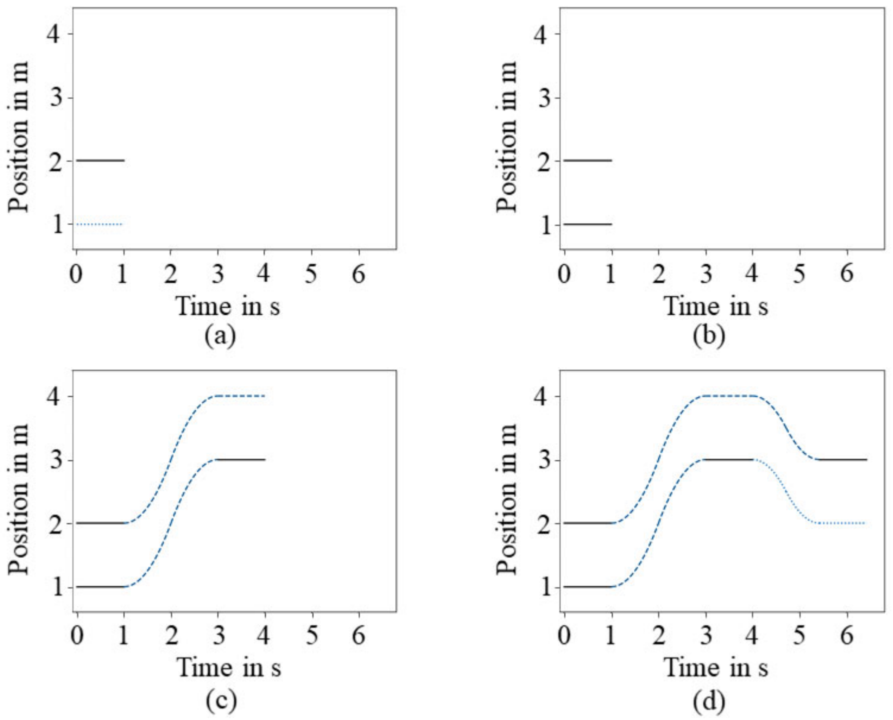

Figure 3: Trajectory planning of two vehicles.

3.4 Solution Space Search

The solution space consists of all possible assignments. It is explored by using a genetic algorithm, for which mutations or moves on the solution space need to be defined in order to get to the next solution. When a move is attempted, it is checked for validity, i.e., it must not break the sequence or assign a task to an occupied vehicle. The following three variants of moves are defined in order to traverse the solution space:

- T-move: Swap priorities of two consecutively prioritized tasks.
- V-move: Swap jobs between two neighboring vehicles.
- J-move: Swap priority numbers of two jobs that are performed consecutively by the same vehicle.

The search algorithm starts the exploration of the solution space with a greedy assignment. It is determined on the basis of the distances between vehicles and tasks. Each vehicle receives the task, whose start location is closest to the vehicle position. The corresponding priority is defined by the length of the distance, whereas a shorter distance means a higher priority. Then, the moves are performed within branches (branch number). Each branch describes a sequence of moves with a defined length (step number). Subsequently, the completion times for the assignments in the branches are computed based on the calculated trajectories. It is defined as the time at which all tasks are performed. After sorting of the assignments by completion time, the best assignments (cut number) with the smallest completion times are extracted. This process is repeated multiple times (iteration number) and the most promising assignment is selected. Figure 4 summarizes the process of getting the best achieved assignment of tasks to the vehicles.

4 COMPUTATIONAL EXPERIMENTS

The developed algorithm is implemented in a Python application that enables the analysis of the performance in different parameter settings. Every experiment contains 10,000 transport jobs that are generated randomly and are composed of storages and retrievals. The jobs are divided into job blocks with a defined number of jobs per block (job block number). For every block, the algorithm accomplishes the assignment of tasks, determination of target coordinates as well as planning of trajectories. On the basis of the trajectories of the vehicles, the completion time for each job block can be computed. The applied parameter specifications of model and algorithm are shown in Table 1. In the following experiments, the number of vehicles deployed as well as the configuration of the lifting systems are varied in order to test the algorithm in different model specifications. Besides, the number of job blocks, i.e., the total number of transport jobs is analyzed regarding the quality of the results.

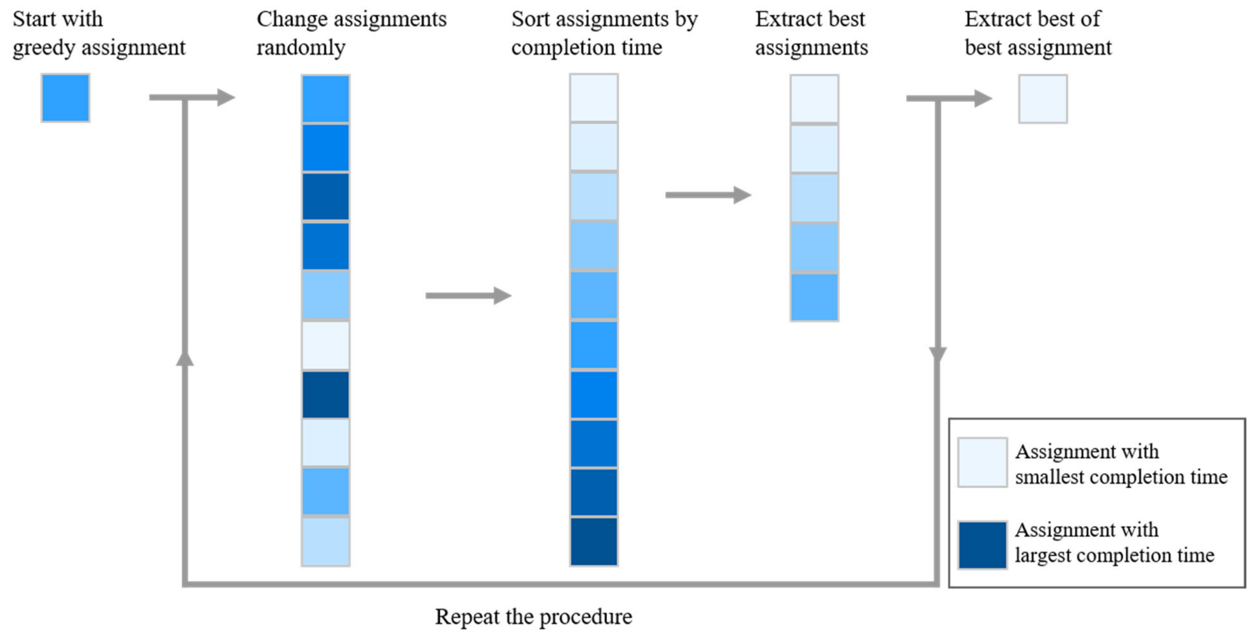
Start with greedy assignment
Change assignments randomly
Sort assignments by completion time
Extract best assignments
Extract best of best assignment
Assignment with smallest completion time
Assignment with largest completion time
Repeat the procedure

Figure 4: Process of getting the best achieved assignment.

5 RESULTS

The completion time of a job block provides information about the throughput of the system, which is a key figure to evaluate the performance of the algorithm. Figure 5 shows the average completion time for a job block with respect to the number of vehicles deployed. The average completion time decreases steadily when increasing the number of vehicles. By deploying two vehicles, the completion time can be reduced by 48 % compared to the deployment of one vehicle. However, when deploying more than two vehicles, completion times decrease to a smaller extent. Since multiple vehicles can execute several jobs in parallel, a block of jobs can be finished in a shorter amount of time. This effect is reduced when the number of vehicles deployed is further increased, since the number of evasion maneuvers increases, too. It can be stated, though, that the throughput capacity can be increased by deploying more than one vehicle and by applying the developed algorithm.

Figure 6 shows the average completion time for a job block depending on the configuration of the lifting system. It turns out that adding more lifting systems can decrease the completion time, especially when positioning the lifting system in the center of the tier. In this sense, by positioning lifting systems in the center (configuration 2), the average completion time can be decreased by 32 % compared to lifting systems in the front (configuration 1). However, by deploying lifting systems in the front as well as in the center (configuration 3), the average completion time can be decreased by 18 % compared to lifting systems only in the center (configuration 2). When positioning lifting systems in the front and in the back (configuration 4), the completion time slightly increased compared to lifting systems in the front and in the center (configuration 3). Finally, the smallest average completion time of 25 s for a job block can be achieved by positioning lifting systems in the front, center as well as back (configuration 5).

In order to investigate the quality of the results in terms of the number of transport jobs per experiment, Figure 7 shows the cumulative mean of completion time when processing different numbers of job blocks. The number of tests corresponds to the number of processed job blocks. It turns out that the solutions quickly converge to the average completion time of 56.7 s (see Figure 6).

Table 1: Parameter specifications and configurations.

Tier	
Length	20 m
Number of vehicles	1 - 5
Distance between storage locations	1 m
Vehicle	
acceleration/deceleration	2 m/s²
Velocity	2 m/s
Loading time	1 s
Safety distance	0.1 m
Length	1 m
Lifting system	
Front position	1
Centered position	2
Front and centered position	3
Front and back position	4
Front, centered and back position	5
Algorithm	
Cut number	10
Iteration number	20
Step number	10
Branch number	10
Job block number	10

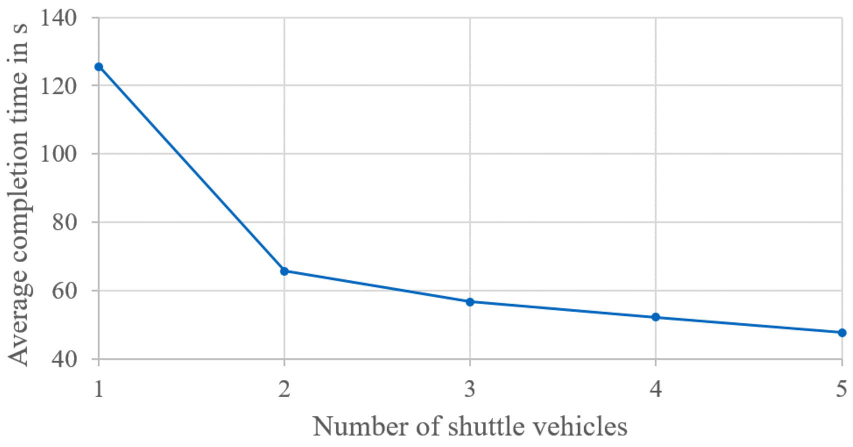

Figure 5: Average completion time depending on the number of shuttle vehicles by applying lifting system configuration 1.

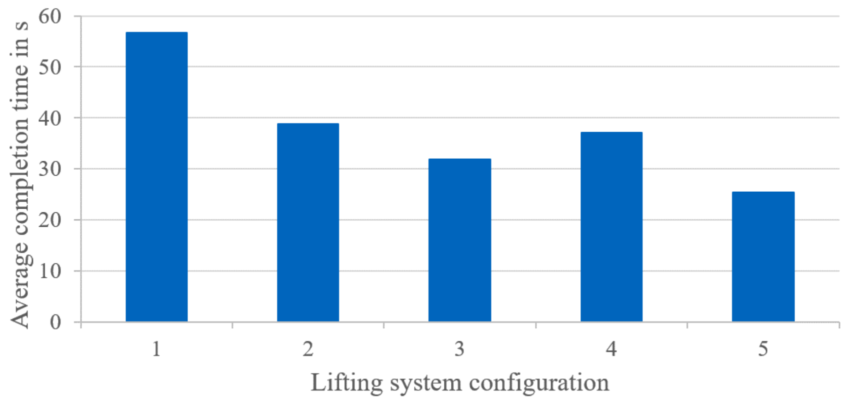

Figure 6: Average completion time depending on the configuration of the lifting system by deploying three shuttle vehicles.

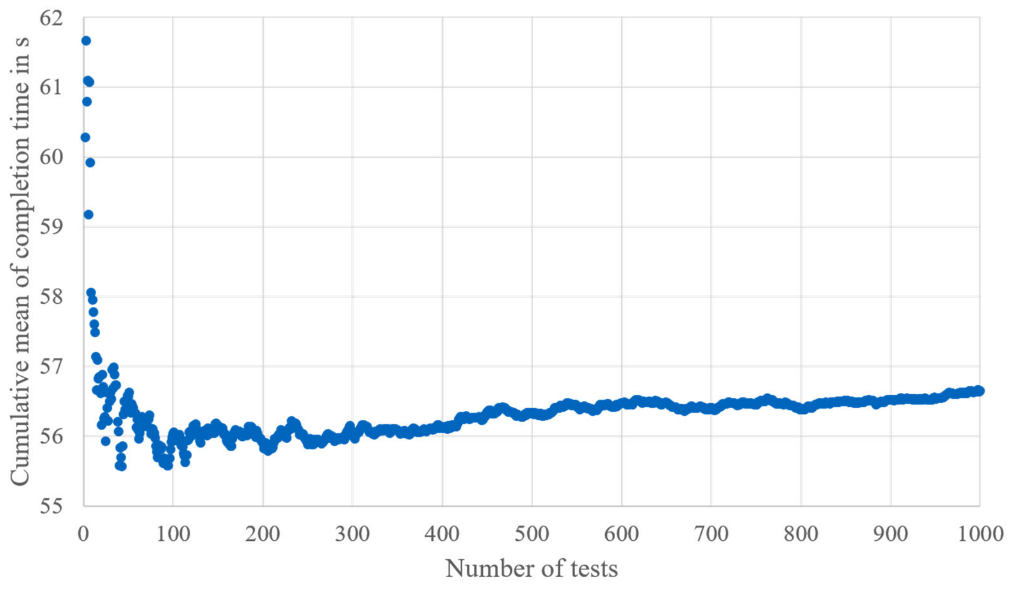

Figure 7: Cumulative mean of completion time by deploying three shuttle vehicles and by applying lifting system configuration 1.

6 CONCLUSION AND OUTLOOK

High-powered AVSRSs represent a new variant of AVSRS that can provide increased flexibility and dynamics. In this paper, an algorithm to schedule and control multiple vehicles on a common track by means of a tier in a high-powered AVSRS is presented. The developed algorithm consists of assignment of tasks, determination of target coordinates, planning of trajectories, as well as search of the solution space. By running computational experiments, the following main results were identified:

- By deploying more than one vehicle on a common track, the completion time can be reduced significantly and, thus, the throughput capacity of the system can be increased.
- The number and position of lifting systems affects the completion time of the tasks. By adding more lifting systems, the completion time of the tasks can be decreased. Centering the lifting system further decreases the completion time and increases the throughput capacity of the tier.

In future work, the algorithm needs to be adapted to lifting systems, in order to schedule and control multiple vehicles in a common shaft. Subsequently, the developed Python application can be integrated in a discrete event simulation model of a high-powered AVSRS to evaluate the robustness and efficiency of the algorithm. In order to optimize the algorithm in terms of task assignment, the applied parameters can be further varied and tested.

REFERENCES

Carlo, H. J., and I. F.A. Vis. 2012. "Sequencing Dynamic Storage Systems with Multiple Lifts and Shuttles". *International Journal of Production Economics* 140:844–853.

Erdoğan, G., M. Battarra, and G. Laporte. 2014. "Scheduling Twin Robots on a Line". *Naval Research Logistics (NRL)* 61:119–130.

FEM European Materials Handling Federation. 2017. FEM 9.860:2017 Guideline Cycle Time Calculation for Automated Vehicle Storage and Retrieval Systems. https://www.fem-eur.com/publication/guideline-cycle-time-calculation-for-automated-vehicle-storage-and-retrieval-system-2017/.

Ishihara, H., and S. Kato. 2013. "Multi-Car Elevator Control Using Dynamic Zoning". In *2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE),* July 4th–7th, Barcelona, Spain, 546–549.

Habl, A., V. Balducci, and J. Fottner. 2019. "Scheduling Multiple Lift Vehicles in a Common Shaft in Automated Vehicle Storage and Retrieval Systems". *The 12th International Workshop on Applied Modeling & Simulation (WAMS2019),* October 30th – November 1st, Singapore, 39-44.

Habl, A., T. Lienert, G. Pradines, and J. Fottner. 2020. "Vehicle Coordination and Configuration in High-powered Automated Vehicle Storage and Retrieval Systems". *SNE Simulation Notes Europe* 30:139–144.

Habl, A., V. Plapp, and J. Fottner. 2019. "Operating High-Powered Automated Vehicle Storage and Retrieval Systems in Multi-Deep Storage". *9th International Conference on Logistics, Informatics and Service Sciences (LISS2019)*, July 26th–29th, Maryland, USA, 715-727.

Habl, A., A. Rautenberg, and J. Fottner. 2020. "Dynamic Control of Multiple Vehicles Moving Along the Same Rail in Automated Vehicle Storage and Retrieval Systems". In *Proceedings of the 19th International Conference on Modeling & Applied Simulation (MAS 2020),* September 16th–18th, Athens, Greece, 12–18: CAL-TEK.

Kress, D., J. Dornseifer, and F. Jaehn. 2019. "An Exact Solution Approach for Scheduling Cooperative Gantry Cranes". *European Journal of Operational Research* 273:82–101.

Kung, Y., Y. Kobayashi, T. Higashi, M. Sugi, and J. Ota. 2014. "Order Scheduling of Multiple Stacker Cranes on Common Rails in an Automated Storage/Retrieval System". *International Journal of Production Research* 52:1171–1187.

Peterson, B., I. Harjunkoski, S. Hoda, and J. N. Hooker. 2014. "Scheduling Multiple Factory Cranes on a Common Track". *Computers & Operations Research* 48:102–112.

Takahashi, S., H. Kita, H. Suzuki, T. Sudo, and S. Markon. 2003. "Simulation-based Optimization of a Controller for Multi-Car Elevators Using a Genetic Algorithm for Noisy Fitness Function". In *The 2003 Congress on Evolutionary Computation (CEC '03),* December 8th–12th, Canberra, Australia, 1582–1587. IEEE.

Valdivielso, A., and T. Miyamoto. 2011. "Multicar-Elevator Group Control Algorithm for Interference Prevention and Optimal Call Allocation". *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans* 41:311–322.

Zhao, N., L. Luo, and G. Lodewijks. 2018. "Scheduling Two Lifts on a Common Rail Considering Acceleration and Deceleration in a Shuttle Based Storage and Retrieval System". *Computers & Industrial Engineering* 124:48–57.

AUTHOR BIOGRAPHIES

ANDREAS HABL is a research assistant at the Chair of Materials Handling, Material Flow and Logistics at Technical University of Munich. His research deals with the control and simulation of high-powered automated vehicle storage and retrieval systems. He holds a Master of Science in Electrical Engineering and Information Technology, which he obtained at Technical University of Munich. His e-mail address is andreas.habl@tum.de.

ANDREI EVSEEV is a master student working at the Chair of Materials Handling, Material Flow and Logistics at Technical University of Munich. He holds a bachelor’s degree in Physics from the Technical University of Munich. His research interests include scheduling problems and optimization of logistics systems. His e-mail address is andrei.evseev@tum.de.

JOHANNES FOTTNER is a professor and head of the Chair of Materials Handling, Material Flow, Logistics at the Technical University of Munich. His e-mail address is j.fottner@tum.de.