Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

THE EFFECT OF INPUT/OUTPUT LOCATION IN AN AUTOMATED STORAGE/RETRIEVAL SYSTEM WITH TWO CRANES

Henri Tokola
Esko Niemi

Department of Engineering Design and Production
Aalto University School of Engineering
Puumiehenkuja 3
Espoo, 02150, FINLAND

ABSTRACT

This paper studies the scheduling of two cranes in automated storage and retrieval systems that have a single output/input location. The cranes are located on a common rail, which restricts their movement, and which also makes the scheduling interesting as the cranes have to dodge each other while operating. The purpose of the paper is to study the scheduling of the retrieval of cartons from the storage to the output location. In order to do that, the paper introduces different scheduling restrictions and constructs a local search heuristic for scheduling the cranes. The heuristic relies on simulation to calculate the length of a given schedule, i.e., the makespan. In the numerical experiments different scheduling restrictions are compared in three different types of automated storage and retrieval systems. The results show how the length of the schedule changes when the input/output location changes in the storage.

1 INTRODUCTION

In modern production, automated storage and retrieval systems (AS/RSs) are used to store items, e.g., in workshops. In AS/RS systems, storage and retrieval functions are automated using robots or cranes. AR/RS allows floor space to be saved and it also makes the storing and retrieval of items efficient. For a review of issues related to AS/RSs see, e.g., the papers written by Roodbergen and Vis (2009) and Muller (1989).

This paper studies an AS/RS system that has two cranes that are located on a common rail. This means that the cranes cannot pass each other. Systems of this kind have been studied in the literature. Kung et al. (2011) introduce a collision avoidance algorithm and study how many collisions there are in an example storage. Kung et al. (2014) have a cluster-based approach where, in each cluster, the cranes pick up and deliver the cartons simultaneously and wait until all the cranes have processed their tasks within a cluster. Ng and Mak (2006) study quay cranes that work on a common linear rail and they propose a heuristic that partitions the area into non-overlapping zones. Aron et al. (2010) and subsequently Peterson et al. (2014) propose an optimization algorithm that finds optimal solutions for the multiple crane scheduling problem. Although there are papers about AS/RS systems with multiple cranes, few papers are similar to the present paper, which studies how different scheduling restrictions and different types of warehouses affect the scheduling of AS/RS. This paper focuses on a situation where there is a single input/output location to which the cartons have to be delivered from the storage.

The rest of the paper is organized as follows. Section 2 describes the AS/RS under study in detail and introduces the scheduling restrictions under study and a local search algorithm. Section 3 presents the results of numerical experiments with different scheduling restrictions in different types of AS/RSs. Section 4 discusses the results and Section 5 concludes the paper.

2 SCHEDULING OF TWO CRANES IN AN AS/RS SYSTEM

2.1 AS/RS Under Study

This paper studies the scheduling of an AS/RS concept that has a single one-sided aisle in which two overhead cranes operate. How the two cranes should operate is not a trivial matter as sometimes it is optimal to fetch the cartons in such a way that they dodge each other. The AS/RS has a single input/output location to which the cartons are moved from the storage. Figure 1 shows the simulation environment which is used by the authors. The environment was built using 3DCreate. Figure 2 shows the operations of a crane in the storage. In the basic delivery of a carton, the operations are the following. First, the empty crane moves to above the place where the carton is. Second, the crane descends to the place where the carton is. Third, the crane grabs the carton. Fourth, the crane ascends to the top of the shelf. Fifth, the crane moves to above the input/output location. Sixth, the crane descends to the place of the input/output location and, seventh, it releases the carton to the input/output location.

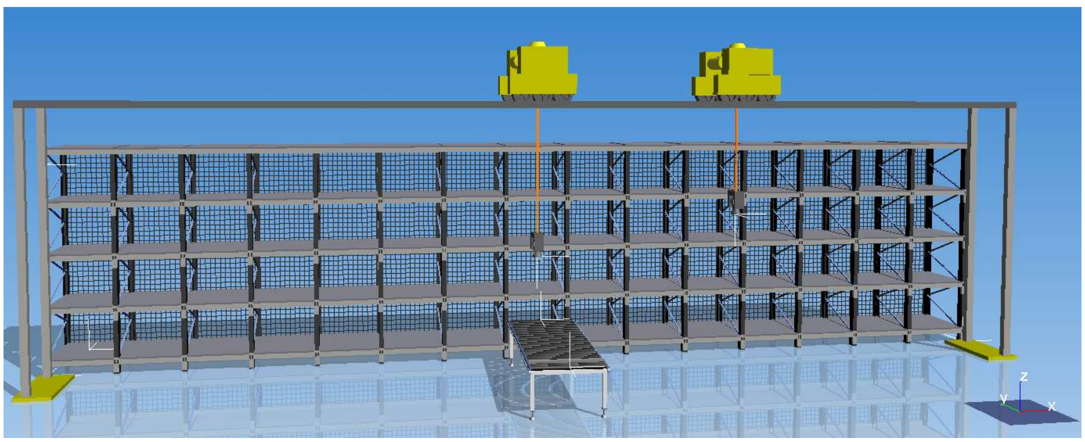


Figure 1: Simulation model of ASRS.

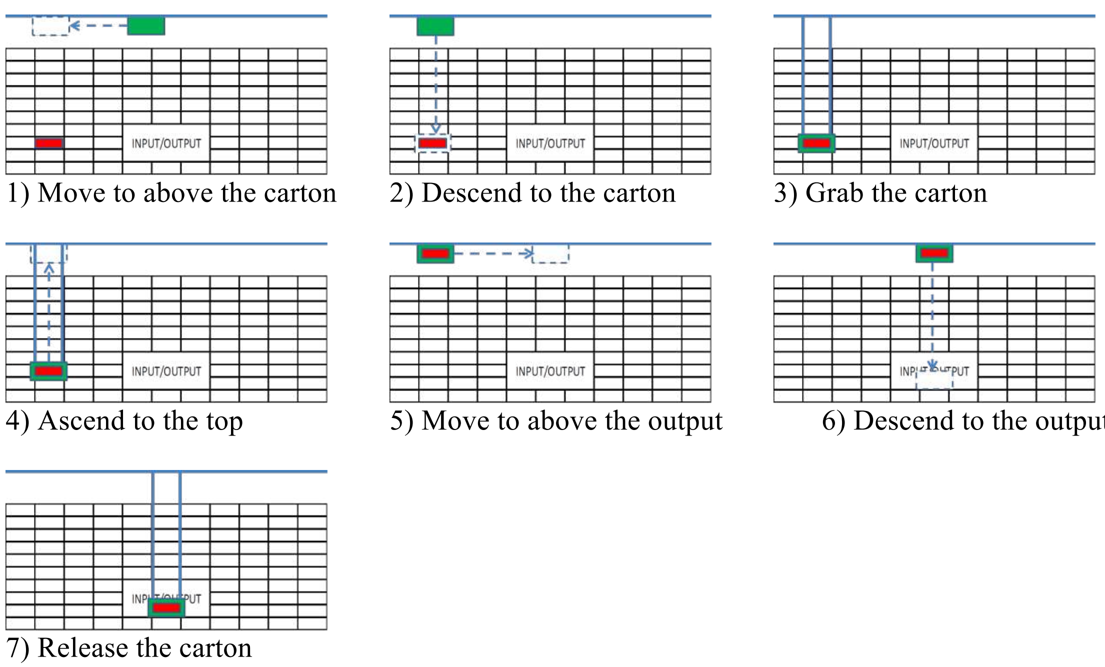


Figure 2: Operations of a crane.

2.2 Scheduling Restrictions Considered

The paper considers six different scheduling restrictions. First, schedules can be constructed using fixed or free sequence constraints, as follows.

- **Fixed sequence:** cartons have to be delivered to the input/output location in a given order. Each crane has its own sequence for the cartons and they can pick the next carton at any time, but they have to wait and dodge the other crane if their carton is not sequenced to be delivered next.
- **Free sequence:** cartons can be delivered in an arbitrary order.

Second, the scheduling might impose restrictions on how the cranes interact with each other. This has three options.

- **No interaction:** the schedule is created in such a way that the cranes can pass each other, pick up cartons and deliver cartons at the same time. This option is not realistic for the AS/RS studied here, but it is used as a comparison option. Here, both fixed sequence and free sequence give the same result.
- **Own area:** each crane can pick up cartons from its own side of the input/output location. The cranes have to dodge each other at the input/output location.
- **Free scheduling**: the cranes have to dodge each other and they can pick up cartons from the same side of the storage.

2.3 Local Search Algorithm for Schedule Creation

The scheduling of two cranes on a single rail is a complex, multi-dimensional problem and it is not obvious how it can be solved using mathematical optimization (see e.g. Aron et al. 2010). Thus, a local search algorithm for schedule creation was implemented. The idea for scheduling is similar to that in the paper by Kung et al. (2011). Given certain cartons to be retrieved, first, a local search method is used to find the best order sequence and best crane for retrieving the cartons. The algorithm is the following:

1. find the initial solution that includes the sequence and initial crane assignments for the retrieved cartons;
2. repeat for a given number of iterations
   a. Make local changes to the sequence and crane assignments (the initial solution or the best solution from step c). In all cases, this includes the assignment of the cranes to deliver a specific carton. In the case of a free sequence, the delivery order of the packets is changed as well.
   b. Generate a schedule using the simulation procedure described later in this section.
   c. If the new solution is better than the best solution, replace the best solution with the new sequence and assignments. The objective is the minimization of the makespan of the schedule.

The above basic local search algorithm was fine-tuned so that it also sometimes sticks to the worse solutions at the beginning of the search. This avoids getting stuck into local optimums. The above local search uses the following simulation procedure to generate the schedule.

1. Generate a sequence of movements without interaction between the cranes. These operations include moving to above the place where the carton is, grabbing the carton, moving to above the output location and releasing the carton into the output/input location, as shown in Figure 2. In the case of no interaction, this schedule is used. Otherwise, the scheduling uses steps 2 and 3.

2. Initialize time1 = 0 for crane 1 and time2 = 0 for crane 2.
3. Repeat the following steps until all operations are gone through:
    a. select the crane which has lower time (time1 or time2). In the case of equal times, select a crane arbitrarily;
    b. if the crane will move to the place where the other crane is, dodge by moving away from the other crane's next operation path and wait until the other crane has performed its next operation;
    c. otherwise, if the next operation or the next after that has the wrong sequence number for outputting a carton then dodge the other crane and wait until it has released its carton to the output/input location;
    d. otherwise, proceed to the next operation of the crane and increase the time respectively.

3 NUMERICAL EXPERIMENTS

In this section, using the local search algorithm described in Section 2, different restrictions on scheduling are studied in three different types of storages. The first type of restriction is a fixed sequence order. In it a sequence is given and has to be followed strictly. The second type includes restrictions on the interactions of the cranes. There are three different interactions. The first is that the cranes do not have any interactions and they can be moved freely without interacting with each other. In the second type, the cranes interact with each other but they have their own areas from which they collect the cartons. The third is free scheduling with interaction.

The default storage details are the following (units are not needed):

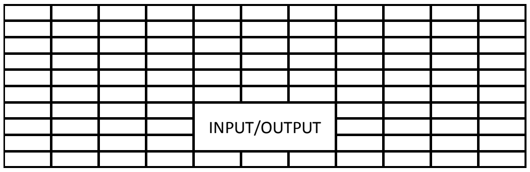
- size of main storage: width = 1, height = 1;
- dodging clearance (cranes have to be at least this far from each other on the horizontal axis)= 0.1;
- release time (time it takes to release a carton at the input/output location) = 1;
- retrieval time (time it takes to grab a carton from a shelf) = 1;
- speed of crane (distance moved in a time unit, assumed to be constant): horizontal = 0.1, vertical = 0.1.
- vertical position of input/output = 0.2. The horizontal position depends on the storage.

The cartons are retrieved from random positions in the storage. The positions of the cartons on the x- and y-axes are uniformly distributed between 0 and 1.

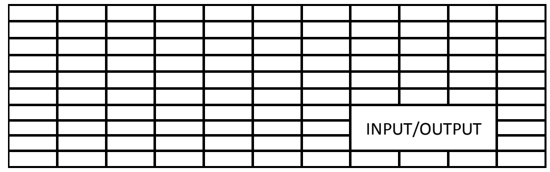
Three different types of storages are studied. They are shown in Figure 3. The first storage is a storage that has an input/output location in the middle. The second storage is a storage that has an input/output location at the side of the storage. The third storage is a storage that has an input/output location far away from the main storage.

Figure 4 shows example schedules for different types of scheduling interactions in the case where the input/output location is at the side of the storage (Figure 3b). A free sequence is used and thus the order of retrievals is different in different schedules.

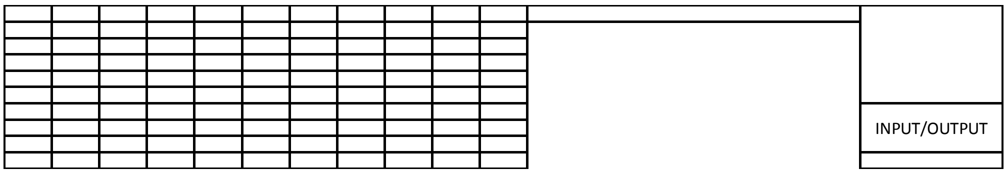
Figure 5 shows large-scale numerical experiments where all the different types of restrictions are compared in all three storages. In each point, 100 iterations are generated and solved, and the average makespans are considered.



a) Single input/output in the middle of the storage (Horizontal position of input/output = 0.5)



b) Input/output at the side of the storage (Horizontal position of input/output = 0.8)



c) Input/output location far from the storage area (Horizontal position of input/output = 2.0)

Figure 3: Storages that are considered in the experiments.

4 DISCUSSION

This section discusses the results shown in the previous section. The purpose is to find out how the different input/output locations and scheduling restrictions affect the makespan of the schedule.

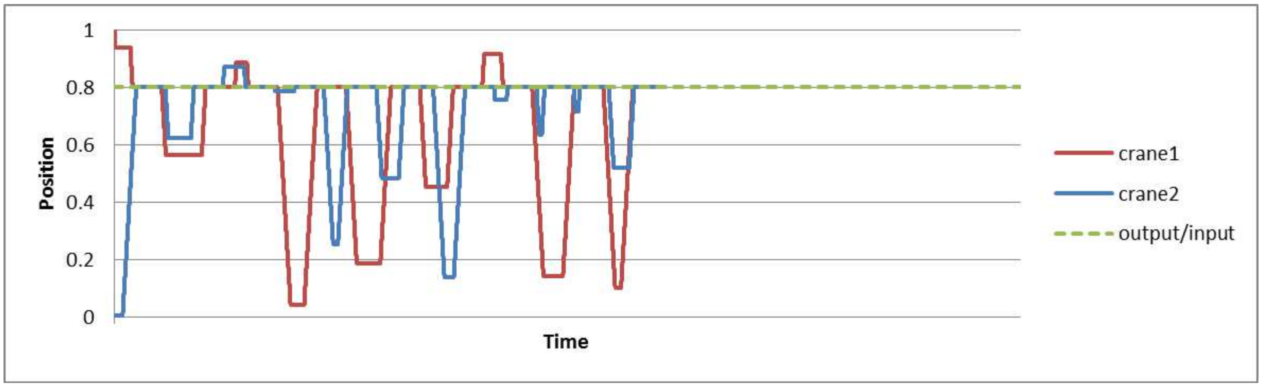
The differences between no-interaction scheduling and the other scheduling restrictions are shown in Figure 5. As expected, no-interaction scheduling generally outperforms free scheduling and free scheduling outperforms own-area scheduling. This can be seen from the examples in Figure 4 and from the results in Figure 5. This is clear because no-interaction scheduling does not have restrictions, free scheduling has interference restrictions, and own-area scheduling has own-area restrictions and interference restriction at the input/output location. As the number of restrictions increases, the results become worse.

The differences between free scheduling and own-area scheduling are interesting. If the input/output is in the middle of the storage (Figure 5a), both methods give similar results if the sequence is free. If the sequence is fixed, free scheduling significantly outperforms own-area. If the input/output is at the side of the storage (Figure 5b), free scheduling outperforms own-area. In the case where the input/output is far from the storage area (Figure 5c), own-area has to use only a single crane, which makes it even worse than in the previous cases.

When it is compared to the no-interaction case, free scheduling gives similar results regardless of the position of the input/output location. However, the differences are smallest in the case where the input/output is far from the storage area. Next comes the case where the input/output is located in the middle of the storage and the worst is the case where the input/output location is at the side of the storage. However, in general, the differences are surprisingly small.

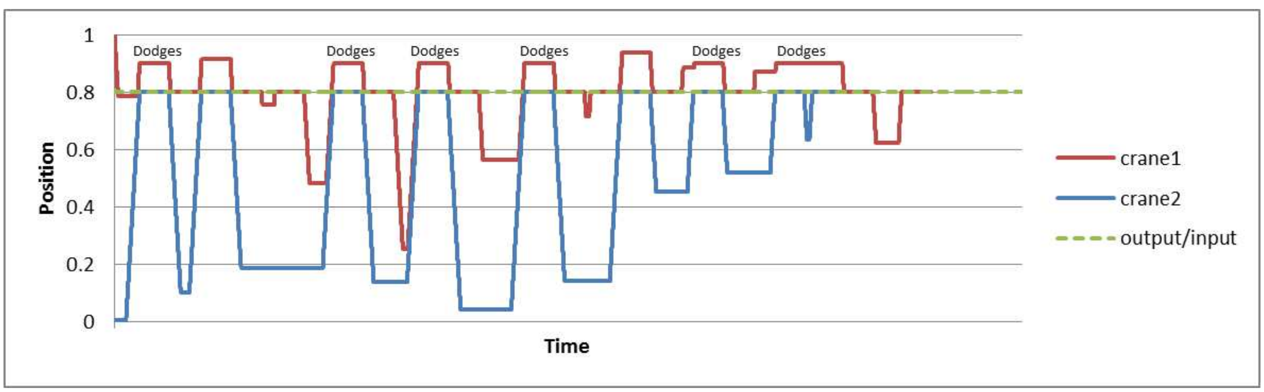
5 CONCLUSION

This paper studies the scheduling of two cranes in automated storage/retrieval systems. The cranes are located on the same rail and thus they cannot pass each other, which makes the scheduling interesting. The paper introduces a local search algorithm that uses simulation to create the schedule. The numerical experiments use the algorithm and they show the differences between the different scheduling restrictions in three different storages. The makespan of the schedule is used as a comparison objective. The most interesting insights found from the experiments are the following.



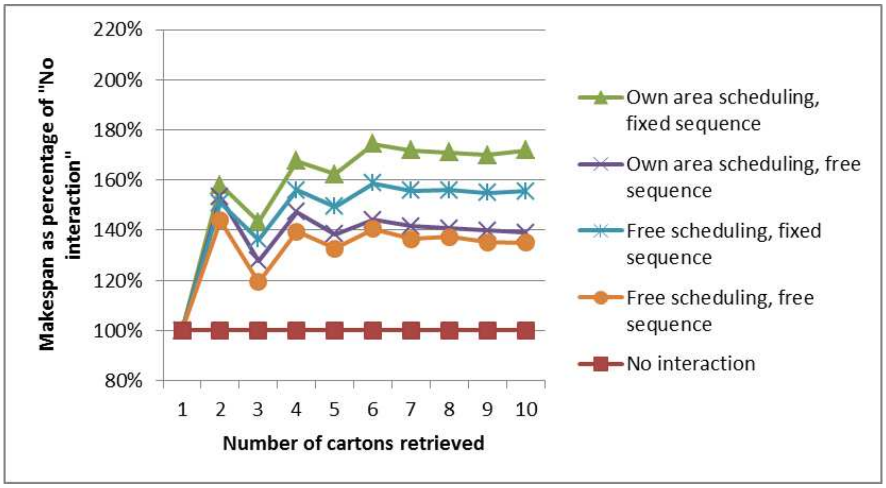
a) No interaction, free order, makespan is 320.

b) Own-area (crane 1 deals with positions from 0.8 to 1 and crane 2 positions from 0 to 0.8), free order, makespan is 528.

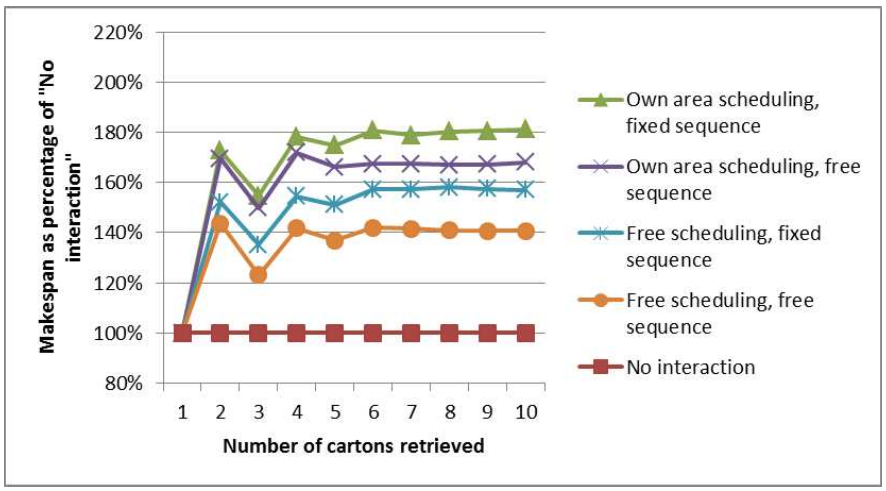


c) Free scheduling, free order, makespan is 475.

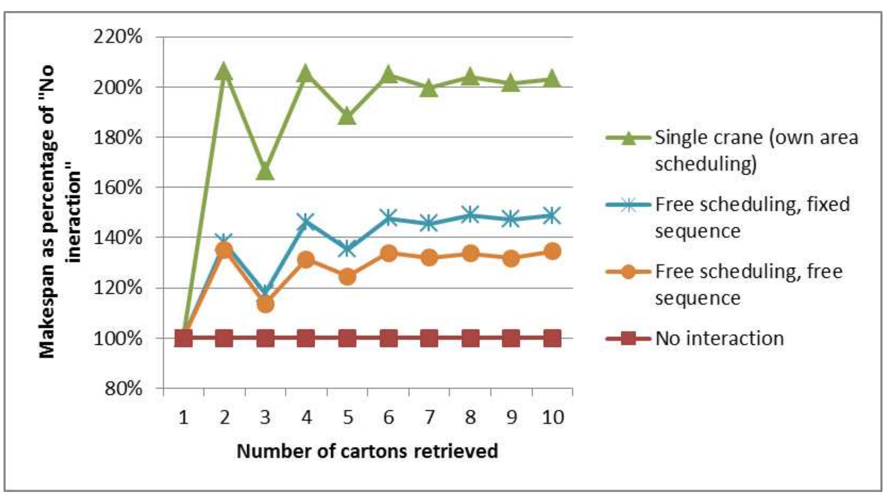
Figure 4: Different schedules for a storage with a single input/output location at the side of the storage. 20 packets are grabbed from random locations and released to the output/input location. The makespan is minimized. The situations where the crane dodges are marked. Other horizontal lines include either grabs from the storage or releases of the carton to the output/input location.



a) Input/output in the middle of the storage.



b) Input/output at the side of the storage.



c) Input/output far from the storage area.

Figure 5: The effect of scheduling restrictions on the makespan of the schedule.

- If the location of the input/output is in the middle of the storage, the own-area algorithm gives a similar makespan to free scheduling.
- The location of the input/output does not have a significant effect on the makespan if free scheduling is compared to no-interaction scheduling.
- The location of the input/output has significant effect if the fixed sequence or own area is used and thus, in these cases, location of input/output should be considered.

During the writing of the paper, several new research ideas arose. First, the effect of the location of the input/output on the makespan could be studied further with different types of speeds and output times of cranes. Second, it would be interesting to implement the scheduling algorithm used here in a three-crane system. This is not, however, trivial. Third, it would also be interesting to compare the proposed algorithms to the algorithms that are introduced in the literature and also to those that are used in practice.

ACKNOWLEDGMENTS

This research was carried out as part of the Finnish Metals and Engineering Competence Cluster (FIMECC)'s MANU program in the LeanMES project.

REFERENCES

Aron, I. D., L. Genç-Kaya, I. Harjunkoski, S. Hoda, and J. N. Hooker. 2010. "Factory Crane Scheduling by Dynamic Programming." In *Operations Research, Computing and Homeland Defense (ICS 2011 Proceedings)*. INFORMS: Monterey, CA, 93-107.

Kung, Y., Y. Kobayashi, T. Higashi, and J. Ota. 2011. "Motion Planning of Two Stacker Cranes in a Large-Scale Automated Storage/Retrieval System." In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 168-173.

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(4), 1171-1187.

Ng, W. C., and K. L. Mak. 2006. "Quay Crane Scheduling in Container Terminals." *Engineering Optimization*, 38(6), 723-737.

Muller, D. J. 1989. "AS/RS and Warehouse Modeling." In *Proceedings of the 21st conference on Winter simulation*, 802-810.

Roodbergen, K. J., and I. F. Vis. 2009. "A Survey of Literature on Automated Storage and Retrieval Systems." *European Journal of Operational Research*, 194(2), 343-362.

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.

AUTHOR BIOGRAPHIES

HENRI TOKOLA is a researcher at the Department of Engineering Design and Production of Aalto University. He holds an M.Sc. degree in Computer Science. His experience includes the modeling and optimization of various production activities from the production control perspective. His current research focuses on manufacturing execution systems (MESs); his e-mail address is henri.tokola@aalto.fi.

ESKO NIEMI is a professor of Production Engineering at Aalto University School of Engineering. His current research interests are related to the modeling and, especially, optimization of manufacturing systems and processes; his e-mail address is esko.niemi@aalto.fi.