SEQUENCING APPROACHES FOR MULTIPLE-AISLE AUTOMATED STORAGE AND RETRIEVAL SYSTEMS

Jean-Philippe Gagliardi[1,2], Jacques Renaud[1,2,*] & Angel Ruiz[1,2]

[1] *Faculté des Sciences de l'administration, Université Laval, Canada G1K 7P4*
[2] *Interuniversity Research Center on Enterprise Networks, Logistics and Transportation (CIRRELT)*
**Corresponding author, Tel: 418-656-7029; Fax: 418-656-2624; E-mail address: Jacques.Renaud@fsa.ulaval.ca*

Abstract

Automated Storage and Retrieval Systems (AS/RS) are used in high velocity distribution centers to provide accurate and fast order processing. While almost every industrial system is comprised of many aisles, most of the academic research on the operational aspects of AS/RS is devoted to single-aisle systems, probably due to the broadly accepted hypothesis proposing that an *m* aisles system can be modeled as *m* 1-aisle independent systems. In this article we present two multi-aisles sequencing approaches and evaluate their performance when all the aisles are managed independently first, and then in a global manner. Computational experiments conducted on a multi-aisle AS/RS simulation model clearly demonstrate that a multi-aisle system cannot be accurately represented by multiple single-aisle systems. The numerical results demonstrate that, when dealing with random storage, globally sequencing multi-aisle AS/RS leads to makespan reductions ranging from 14% up to 29% for 2 and 3-aisle systems, respectively.

Keywords: Automated storage and retrieval systems; multi-aisle; sequencing; simulation.

Pour citer ce texte:
Gagliardi J.-P., Renaud J. & Ruiz A., Sequencing approaches for multiple-aisle automated storage and retrieval systems. *International Journal of Production Research*, 53 (19), **2015**, 5873-5883

1. Introduction

Distribution centers (DC) play a central role in modern supply chains as they make the connection between supply and demand. Recent trends lead to smaller, but more frequent deliveries, and the "within 24 hours" service is becoming a standard in many industries. As a result, *Automated Storage and Retrieval Systems* (AS/RS) are now widely used in modern DC as they provide fast, accurate and efficient handling of materials on a 24/7-basis. In its most basic form, an AS/RS consists of storage racks where products are stored and retrieved automatically. In this article, we are interested in one of the most popular AS/RS type: the unit-load single-depth multiple-aisle AS/RS. Unit-load AS/RS are systems where a single unit-load, typically a full pallet, is moved by an automated crane between the input/output point of the system (I/O) and the storage or retrieval locations. In single-depth systems, each location can only hold one pallet. Each aisle has storage racks on both sides and multi-aisle AS/RS are equipped with a number of cranes (or storage and retrieval machines). In some contexts, cranes can move from one aisle to another by means of different technologies such as automatic aisle-transferring or curve going, for example (Lerher et al. 2006). However, in the case considered here, each crane is confined to a particular aisle. In other words, the cranes are "captive" within their respective aisle and, needless to say, the number of cranes is equal to the number of aisles. Almost every industrial AS/RS is comprised of several aisles. However, as it will be seen later, scientific research focused mostly on single-aisle systems or viewed multi-aisle AS/RS (MA–AR/RS) as several independent single-aisle AR/RS.

This article contributes to the literature in the following manner. First, we develop sequencing heuristics for managing MA–AS/RS efficiently. Then, we clearly demonstrate the benefits of a global approach which manages the MA–AR/RS as a single system instead of operating multiple single-aisle AS/RS. We show that under different number of aisles and different location policies, the proposed global multi-aisle method reduces both the crane travel time and the makespan required to complete a set of orders, when compared to those where aisles are managed independently. Finally, we demonstrate that a storage assignment using an across-aisle strategy is clearly preferable to a within-aisle strategy when managing a MA–AS/RS. The rest of this paper is structured as follows: Section 2 reviews the relevant literature on multi-aisle systems and Section 3 presents storage and sequencing basics for single-aisle AS/RS. Section 4 develops two new look-ahead algorithms for globally sequencing multi-aisle systems. Section 5 depicts the simulation model used to execute our numerical experiments, whose results are presented in Section 6. Finally, Section 7 concludes the paper.

2. Literature review

AS/RS research focuses in either on strategic-tactical or operational issues. Among the former, research questions are concerned mostly by the system's configuration and the selection of the appropriate storage assignment policy. Research devoted to operational decisions deals with request sequencing and/or batching and dwell-point positioning. However, these researches are almost exclusively conducted on single-aisle systems. We refer the interested reader to Roodbergen and Vis (2009) for a comprehensive survey on AS/RS. In this section, we aim to review and position the few papers dealing explicitly with multi-aisle systems. As most of the works devoted to MA–AR/RS are of an analytical nature, we will first review works proposing analytical models, followed by those including simulation models. Finally, we will review some other specific and interesting multi-aisle contexts.

In one of the pioneering works on analytical models, Ashayeri et al. (1985) proposed a design optimization model minimizing investment and operating costs over the lifetime of an AS/RS system. Hwang and Ko (1988) developed a travel time model to find both the minimum number of non-captive cranes and the number of aisles to be served by each crane in order to minimize the system's total travel time. Malmborg (2001, 2002 and 2003) developed several analytical models to evaluate the impact of the number of aisles on various performance measures. Lee et al. (2005) studied a multi-aisle captive cranes AS/RS having racks with modular cells and used a mathematical model to minimize the total unused space in the system. Koh et al. (2005) studied a multi-aisle end-of-aisle order picking system served by a single order picker. Lerher et al. (2006) presented analytical models for travel time computation in multi-aisle automated warehouses. Although this work considered a single non-captive crane serving several aisles, it assumed that both the storage and the retrieval operations associated with a given cycle were performed in the same aisle. Later on, Lerher et al. (2010a) extended this work to a more complex situation where the storage and the retrieval operations can be performed in different aisles.

When a very detailed representation of reality is pursued, simulation becomes a very appealing tool. Houshyar and Chung (1991) developed a simulation model to evaluate the behavior of a multi-aisle captive-crane AS/RS system under various assumptions. Lee et al. (1996) simulated a complex narrow multi-aisle AS/RS with aisle-captive cranes where a set of rail-guided vehicles brought pallets to the order picking station. Lerher (2006) proposed a simulation model to evaluate the benefits of different storage strategies in a multi-aisle AS/RS with non-captive cranes. Recently, Gagliardi et al. (2014b) proposed a generic object-oriented multi-aisle AS/RS with aisle-captive cranes simulation framework which accounts for a number of features observed in industrial settings.

Multiple aisles systems have also been studied in different related contexts. Lerher et al. (2010b) proposed travel time models for the double-deep AS/RS. Lerher et al. (2011) simulated mini-load multi-shuttle AS/RS and Ekren & Heragu (2011) proposed a simulation model to analyze a complex storage and retrieval system served by an autonomous vehicle. Carlo and Vis (2012) studied a special storage system with multiples lifts and shuttles. In the particular context studied, activities' sequencing became very challenging due to the presence of two non-passing lifts which shared a single mast per aisle. In recent years, energy efficiency has also become a subject of interest for researchers (see for example Lerher et al., 2014).

Despite of the large amount of research devoted to AS/RS and related storage systems, it appears that multi-aisle unit-load captive-crane AS/RS (the specific configuration studied here) were almost exclusively approached from a system's design perspective, leaving the operational issues such as multi aisle product location or activities sequencing, aside. This can be explained by the fact that many articles suggested tackling multi-aisle AS/RS as a set of independent one-aisle systems (see Hausman et al. 1976). For example, both Kulturel et al. (1999) and Van den Berg (2002) postulated that *a single m-cranes system can be represented by m one-crane systems* (Kulturel et al. 1999, p. 741). To mitigate this drawback, this article analyzes sequencing decisions within a multi-aisle system and demonstrates that a greater performance can be achieved when the system is managed as a whole instead of as a set of independent systems. Our results show that an appropriate sequencing of the system's aisles is worth performing and that, by doing so, important savings can be achieved. Before proposing multi-aisle systems' sequencing strategies, let us review the basics of sequencing and storage assignment decisions in a single-aisle case and how they differ in a multi-aisle context.

3. Storage and sequencing basics for AS/RS

AS/RS performance is closely related to two kinds of decisions: *storage assignment* and *requests sequencing*. When dealing with a multi-aisle AS/RS, storage assignment and requests sequencing decisions become more complex.

3.1 Storage assignment decisions

Storage assignment refers to how products, or families of products, are assigned to storage locations in order to optimize the performance of the system. In *random storage*, each product can be assigned to any location within the system. On the other hand, *dedicated storage* implies that each product has its own set of dedicated locations. In dedicated storage, products are generally assigned to locations following the full-turnover-based (FTB) policy, which consists in assigning the best storage locations to products with the highest demand frequency. In a *class-based* storage policy (CB), AS/RS locations are divided into classes according to their distance to the I/O point. High velocity products are assigned to the class nearest to the I/O point, often called class A. Products are then allocated to classes according to their relative velocity and, within a class, products are randomly assigned to locations. In a multi-aisle configuration, a system output point related to the conveyor deserving all the aisles needs to be considered. In this case, distances may be calculated from either the aisle I/O point or to the system output point, thus leading to potentially different configurations if a class-based storage is used.

To the best of our knowledge, storage assignment has never been explicitly studied in multi-aisle AS/RS. Researches often assume that results obtained for single-aisle are also valid for multi-aisle AS/RS. However, multi-aisle systems introduce additional possibilities with respect to space allocation. We may distinguish between *across-aisle* and *within-aisle* storage location assignments, as is the case with standard warehouses (see for example Petersen and Schmenner, 1999). In MA–AS/RS, and considering aisle ordering, an *across-aisle storage* is obtained by assigning the product with the highest demand to the best available location in the first aisle, the second highest demand product to the best location in the second aisle and so on, and then coming back to the first aisle after assigning a product to the last one. This implies that the global workload between the aisles is balanced due to the forced presence of high velocity products in all the aisles.

Considering an order or preference in the system's aisles, a *within-aisle storage* is obtained by assigning the product with the highest demand to the best location available in the first aisle, the second highest demand product to the second best location of the same aisle and so on, in order to assign every location in the first aisle before using the next preferred aisle. This implies that the highest workload will be concentrated in the first preferred aisle. These differences are illustrated in Figure 1.

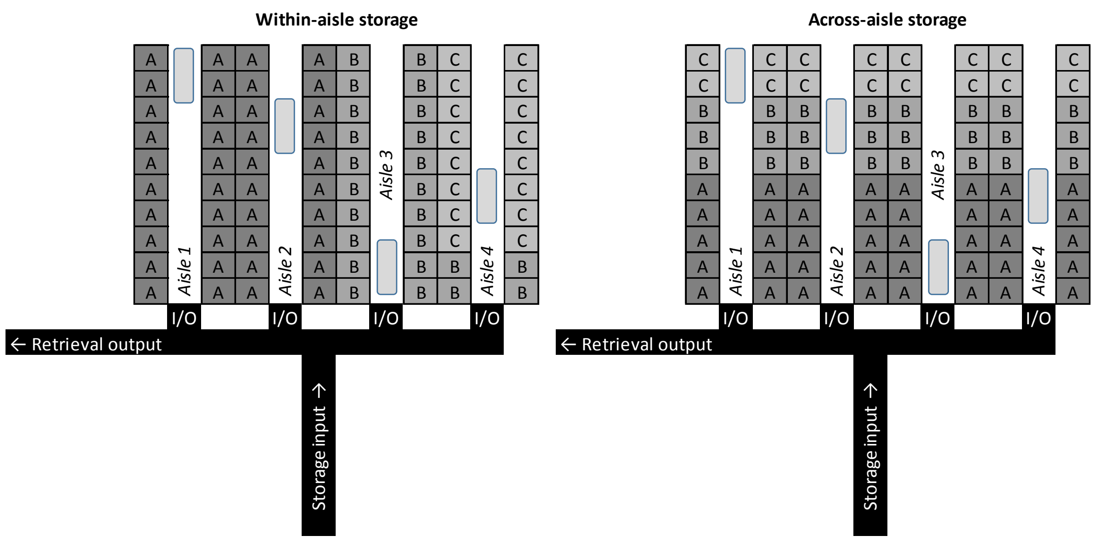

Figure 1. Within-aisle and across-aisle storage in a multi-aisle AS/RS

3.2 Single-aisle classical sequencing approaches

Due to the conveyor system's physical constraints, it is generally assumed that storage requests (pallets arriving from the production system) are served following the first-come first-serve rule (Roodbergen and Vis, 2009) and are represented by the ordered set *S*. However, it is also assumed that the set *R* of retrievals doesn't follow a precise order. For given sets *S* and *R*, the single-aisle AS/RS sequencing problem consists in minimizing the distance traveled by the crane to store all SKU of *S* and retrieving all SKU of *R*. This can be done by using a combination of *single-cycles* (the crane leaves the I/O point and performs only one operation, either storage or retrieval) or of *dual-cycles* (the crane leaves the I/O point, performs a storage operation followed by a retrieval and then returns to the I/O). The sequencing can also be *static* or *dynamic*. In static sequencing, all the SKU in S and R are planned and executed before considering new incoming requests. In the static case, Lee and Schaefer (1996) studied the special case of a one-class system, in which each retrieval request was associated with a specific location and where storage requests could be stored in any open (unused) position. In another study by Lee and Schaefer (1997), the storage requests were considered according to a first-come, first-served rule, and assigned to predetermined storage locations. Retrievals could be sequenced, but their specific locations were known. In this case, the problem corresponds to an assignment problem.

When we allow either S or R to change over time, the sequencing is called *dynamic*. Han et al. (1987) suggested two strategies for dynamic sequencing. In *block sequencing*, storage requests and retrieval requests sets are separated into blocks, or subsets. Then, a single block of storage requests and a single block of retrieval requests are sequenced. Once the requests in these blocks have been scheduled, another pair of blocks is selected. In *dynamic sequencing*, the sets are updated each time a new request is added.

Before reviewing the three basic sequencing heuristics proposed in the literature to solve requests sequencing problems, let us introduce the following notation which divides the crane's travelling into three legs. Let T_s be the travelling time between the input/output point (I/O point) and the storage location s, T_{sr}, the interleaving travelling time between the storage location s and the location of the retrieval r, and T_r, the return time from the retrieval location r to the I/O point. If we define $D_{sr} = T_{sr}$, and sequence the retrievals in order to minimize D_{sr}, we obtain the nearest neighbor (NN) heuristic (Han et al., 1987), which minimizes the interleaving travel time. If we define $D_{sr} = T_s + T_{sr}$, we obtain the shortest-leg (SL) heuristic, which minimizes the total travel time from the I/O to the storage location and from the storage location to the retrieval location (Han et al., 1987). Finally, if we set $D_{sr} = T_s + T_{sr} + T_r$, we obtain the minimum total travel time (TT) (Lee and Schaefer, 1996). Gagliardi et al. (2014a) generalized the algorithmic design of Lee and Schaefer (1996). They also proposed a mathematical model and demonstrated that it consistently outperforms classical heuristics for various configurations of single-aisle AS/RS. Again, to the best of our knowledge, no sequencing approach has been developed for the multi-aisle AS/RS.

4. A look-ahead algorithm for sequencing MA–AS/RS

In this section we describe a look-ahead algorithm for sequencing storage and retrieval requests in an AS/RS with L aisles and where each product is available in one single location. The basic assumptions used in the studied multiple-aisle unit-load AS/RS with captive cranes are the following:

- The system contains a number of parallel aisles with storage racks on both sides. There are double storage racks between two consecutive aisles and single storage racks along the warehouse walls, as depicted in Figure 1.
- The system is a unit-load AS/RS. In other words, each pallet holds only one part number or item type.
- Cranes have independent drives on both axes, allowing them to travel horizontally and vertically simultaneously.
- There is one captive crane per aisle and it operates either on single command or dual command cycles.
- All storage locations can hold any item.
- The turnover rate of each item is known in advance and does not change over time.
- Distance (i.e., travel time) from rack location i to rack location i' is symmetrical and does not change over time.
- Crane acceleration and deceleration are assumed instantaneous and are neglected. See Section 6 for additional discussion on this assumption.
- Pickup and deposit times are assumed constant.

This new look-ahead algorithm dynamically considers the locations released when extraction requests are executed as well as the availability of the products stored in the system during the sequencing. To this end, the new algorithm performs the scheduling over the foreseen or anticipated state of the AS/RS.

As each product is available in one single location, the set R of retrievals can be divided into L subsets R_l, one for each aisle. We also define, for each location in aisle l, its *status* (*open* or *occupied*) and its *profile*. The location profile indicates the kind of product it can hold: if a random storage policy is elected, a location can receive any product (*profile* = *all*); under 2-class storage, each location may either be *A* or *B*. If full-turnover is used, each location is assigned a specific product *p* (*profile* = *p*). Finally, for a single-depth, two-sided system, the state of aisle *l*, π_l, is a matrix storing the status and profile of every location.

The system is managed by *L Aisle managers* and by a *Global scheduler*. The *Global scheduler* receives all the retrieval and storage requests, pairs them up in order to optimize the global system and assign pairs to aisles. For each aisle, the *Aisle manager* performs the storage and retrieval tasks assigned to its aisle by the *Global scheduler*. For a given aisle *l*, tasks assigned by the Global Scheduler are put in an ordered set M_l, which contains a list of information *pairs* describing the forthcoming operations. Four pairs are possible: $\{s,r\}$, $\{\varnothing, r\}$, $\{s, \varnothing\}$ and $\{\varnothing,\varnothing\}$. For aisle *l*, the pair $\{s,r\}$ corresponds to a dual-cycle starting at the I/O point of aisle *l*, storing product *s*, retrieving product *r* and coming back to the I/O point of aisle *l*. The pair $\{\varnothing, r\}$ corresponds to a single-cycle in aisle *l* retrieving product *r* only; $\{s, \varnothing\}$ corresponds to a single-cycle storing product *s* only; finally $\{\varnothing,\varnothing\}$ means that the crane is idle as there is no more operation to be executed in aisle *l*. Finally, we also record σ_l indicating the time at which the last task on aisle *l* is to be completed.

The sequencing decision consists in establishing for each incoming storage request, (1) to which aisle it should be assigned and (2) how it will be sequenced with the available retrievals. This task is performed by the look-ahead algorithm described below.

Look-ahead scheduling algorithm

Step 0 Initialization: At the beginning of the process, all the data structures describing the MA–AS/RS are built: the set of storages *S*, the sets of retrievals for each aisle R_l, the state of each aisle π_l and its pending tasks M_l. This step is performed only once at the beginning of the process.

Step 1 Sequencing: Let *s* be the first SKU in *S*. This step considers each aisle *l* and each retrieval candidate $r \in R_l$, and calculates the dual-cycle leading to the lowest total travel time. This step can be formalized as follows:

1: *BestTime*: = Inf

2: *BestChoice*: = {}

3: For each aisle l: = 1 to L do

4: Considering the foreseen state π_l of aisle l

5: For each retrieval $r \in R_l$ Do

6: For each empty (storage) location δ in aisle l where s can be stored

7: $Time := T_\delta + T_{\delta r} + T_r$

8: If $Time < BestTime$

9: Then $BestTime := Time$, $BestChoice := \{l, r, \delta\}$

Note that if a particular R_l is empty, the travel time is simply computed as a single-cycle and is $2T_\delta$.

Step 2 Implementing: It has been decided that storage s will be assigned to location δ in aisle l, and paired up with retrieval r (or with no retrieval if a single-cycle is selected). This task is put at the end of the list of operations M_l to be performed by the crane in aisle l. Then, the foreseen state π_l and time σ_l are updated as if the task had been executed: status of location δ changes from *open* to *occupied* and the status of the location where r was retrieved is changed to *empty*. Sets S and R_l are updated and the system returns to Step 1.

The algorithm described above will be called *Travel Time Look-ahead Scheduling Algorithm* (TT–LSA) as it searches over all aisles and over all retrieval for the best single or dual-cycle to minimize the cranes' travel time. As we will see in the computational section, it may be appealing to perform the scheduling in order to balance the work between the aisles. In this case, we may want to assign the next operation according to σ_l. The *Early Finish Look-ahead Scheduling Algorithm* (EF–LSA) can be obtained by modifying line 7 of Step 1 as follows:

7: $Time := \sigma_l + T_\delta + T_{\delta r} + T_r$

Scheduling independent aisles

When a multiple-aisle AS/RS is managed as multiple single-aisle AS/RS, it means that there is no interaction between the aisles. Thus, when a product is retrieved, the next unit of the same product arriving to the AS/RS is stored back in the same aisle. The proposed look-ahead algorithms can easily be adapted to reproduce the behavior of independent aisles by removing the loop on each aisle (line 3 in *step 1*) and returning each SKU to its initial aisle. In the following, the tested independent sequencing algorithm, Ind–Seq, is obtained by removing the global view of the TT–LSA.

5. Simulation model

In order to evaluate the performance of the proposed sequencing approaches for MA–AR/SR, we used a discrete-event simulation engine based on the model proposed by Gagliardi et al. (2014b). Discrete-event simulation (DES) focuses on the asynchronous creation and execution of instantaneous events coordinated by a discrete-event engine or clock. The simulation engine used in this paper is inspired by the one proposed by Pidd (2004), which consists of a three-phase algorithm that allows the clock to be advanced asynchronously from one event to the next. We refer the interested reader to Gagliardi et al. (2014b) for a thorough description of the DES implementation.

The MA–AR/SR model is basically comprised of a demand generator (*Global retrieval requests list*), decisional components (the *Global Scheduler* and *Aisle managers*), and physical components (cranes and storage locations). Figure 2 summarizes the logic of the MA–AS/RS simulation model. As it can be observed, a *Global retrieval requests list* (the retrievals' demand) is built by sampling from a probability distribution selected by the user. This list contains every retrieval request required for a particular run and feeds the *Global Scheduler* according to the required length of the planning horizon. The *Global Scheduler* mimics the behavior of the AS/RS controller and this is where the sequencing approaches (i.e. TT–LSA and EF–LSA) are programmed. Depending on the length of the planning horizon, the *Global scheduler* may have 0, 1 or multiple requests queued in front of each aisle when a decision is required. Once a best cycle is determined, it is sent to a specific aisle and queued for execution by the *Aisle manager*. Retrievals are generated using an approach similar to Gagliardi et al. (2012) and Hausman et al. (1976).

Finally, physical components represent the actual storage racks and cranes. Storage locations in an aisle are modeled by their unique coordinates in the rack and an array containing different attributes (length, width, depth, etc…). Also, for each aisle, the travelling times between each pair of storage locations are computed and recorded in a distance matrix. By doing so, any model can be used to model travelling times in the system. Cranes are modeled as servers, whose function is to process the storage and retrieval requests received from the *Aisle manager*. Each crane has one inbound storage queue and one outbound retrieval queue. The storage queue lists the products to be stored in a given aisle, while the retrieval queue lists the information concerning the loads to retrieve. Thus, when a double cycle is performed, the server is busy during a lap equal to the time required to travel from the I/O point to the storage location, plus the travel time from the storage location to the retrieval location, plus the time to travel between the retrieval locations to the I/O point.

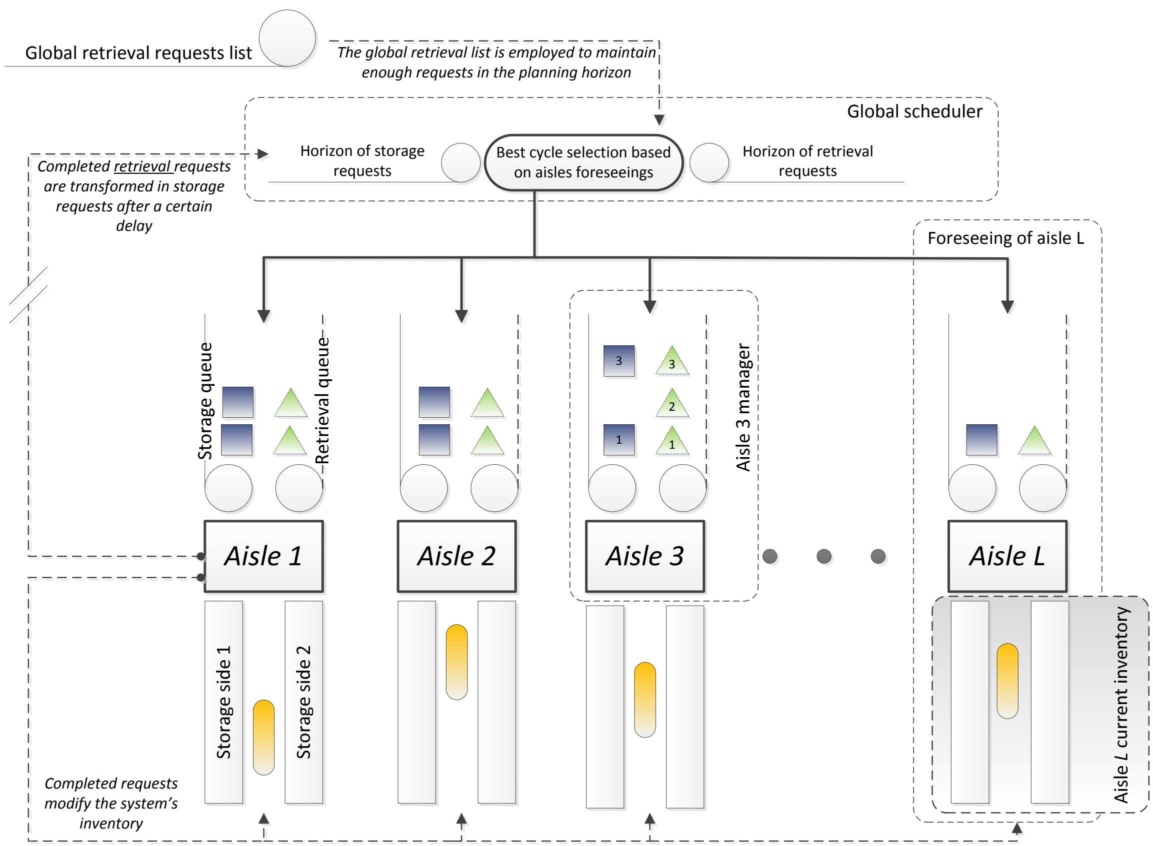

Figure 2. Request management logic of the MA-AS/RS simulation model

6. Computational results

Extensive numerical experiments were conducted to outline the major findings of this article. Based on these experiments, we will first show that important improvements can be obtained by globally managing a multiple-aisle AS/RS rather than as a set of independent single-aisle systems and, secondly, that an across-aisle strategy is clearly preferable to a within-aisle strategy when managing a MA–AS/RS.

Table 1 presents the system's configurations that were tested, which includes 1, 2 or 3 aisles each containing 600 locations. Each location is assumed to be 1 meter, both in height and in length. Since the locations to product ratio (LTPR) is equal to one, these designs include 600, 1200 or 1800 SKUs that are stored using an across-aisle policy or a within-aisle policy. Following each policy, we are also interested in the performance of different class divisions. In a 2-class policy, it is assumed that the first 20% of the fast-moving items are assigned to the 20% of locations closest to the I/O following their respective storage assignment policy. The retrieval requests list is generated by sampling a "*s*" shape probability distribution where *s* was set to 0.5, meaning that 20% of the SKUs generate 45% of the total demand. See Gagliardi et al. (2012) and Hausman et al. (1976) for more information on demand generation for AS/RS systems. As previously stated, the crane's speed is assumed to be constant, which does not correspond to the reality. However, as mentioned in Section 5, acceleration and deceleration can easily be incorporated into the travel time calculations. Also, as there is one crane per aisle, there is no need to consider curve-going or automatic aisle-transferring systems (Lerher et al. 2006).

Table 1. MA-AS/RS configurations under study

Factor	Levels
Number of aisles	1, 2, 3
Rack length (in number of locations)	25
Rack height (in number of locations)	12
Locations to product ratio (LTPR)	1
Number of storage classes	1, 2 (20/80)
Sequencing policies	Independent, TT-LSA, EF-LSA
Storage assignment policies	Across, Within aisle storage
Crane horizontal travel velocity	1 meter per second
Crane vertical travel velocity	0.4 meters per second
Handling time (storage or retrieval)	10 seconds

The following simulation parameters were employed throughout the experimentation. Each configuration was replicated 10 times. A replication consisted in the execution of 10,000 dual-cycles preceded by a certain amount of retrieval requests executed in single-cycle mode. The amount of single-cycles per run acts as a warm-up period in order to dissociate the system from its initialization condition, implying that the system is full at time $t = 0$. The length of the warm-up period is set according to the system's design and consists of $0.20L$ where L is the total number of locations within the system. At the end of the warm-up period, storage requests begin to arrive at the global scheduler, and the dual-cycles start.

The next two sections report and analyze the numerical results when using within and across aisle policies respectively. For each policy, we tested two class-based storage assignment configurations (1 and 2 classes). We also ran experiments using the full-turnover based policy. These results will not be presented since, as expected, all the methods (Ind–Seq, TT–LSA and ET–LSA) produced the same results. Indeed, since each SKU has its own dedicated location the sequencing decision clearly becomes irrelevant. Nevertheless, these results contributed in verifying our simulation model.

Tables 2 and 3 report, for each configuration, the average *makespan* (lines *Mk*) and the average total travelling time (lines *TT*) of the cranes (or the single crane in the 1-aisle configuration), given in minutes over 10 instances. These results were produced when the system was managed as a set of independent aisles (column *Ind–Seq*) or globally managed by the *Travel Time Look-ahead Scheduling Algorithm* (column *TT–LSA*) and the *Early Finish Look-ahead Scheduling Algorithm* (column *EF–LSA*). Line *Q* gives the system productivity as the average number of pallets handled per hour. Lines Δ_{Mk} and Δ_{TT} give the improvement, in percentage, achieved by TT–LSA and EF–LSA methods over the independent sequencing in terms of makespan (Δ_{Mk}) and total travel time (Δ_{TT}) respectively. Finally, for each configuration and management method, Tables 2 and 3 also report the crane's utilization (ρ_1 to ρ_3) and its average ($\bar{\rho}$). The crane's utilization, in percentage, is computed as the crane's total travel time plus the crane's total handling time divided by the makespan.

Results for the within aisle storage policy

The upper part of Table 2 reports results for the 1–aisle case for the within aisle storage policy. As expected, the results for the 1 and 2-class configurations, Ind–Seq and TT–LSA were the same and, with respect to EF–LSA, the differences were not significant. Since the crane's utilization is of 100% and there's no idle time, *Mk* equals the *TT* value plus the crane handling time. This computation is not as straightforward as in the multi-aisle case. In fact, *Mk* depends on how requests are assigned to aisles, which can eventually create idle times, but also different handling times between aisles.

Table 2. Performance comparison between independent and global scheduling (within aisle storage strategy)

# of aisles		1-class			2-class		
		Ind–Seq	TT–LSA	EF–LSA	Ind–Seq	TT–LSA	EF–LSA
1	*Mk*	13 153	13 153	**13 149**	12 480	12 480	12 486
	Δ_{Mk}		0%	0%		0%	-0.0%
	Q	91.3	91.3	91.3	96.2	96.2	96.1
	TT	6,486	6,486	6,482	5,814	5,814	5,819
	Δ_{TT}		0%	0%		0%	-0.0%
	$\bar{\rho}$	100%	100%	100%	100%	100%	100%
2	*Mk*	9 712	7 503	**6 968**	9 368	9 037	**9 031**
	Δ_{Mk}		23%	28%		3.5%	3.6%
	Q	123.6	159.9	172.2	128.1	132.8	132.9
	TT	7,944	6,386	6,440	6,628	6,802	6,722
	Δ_{TT}		20%	19%		-2.6%	-1.4%
	ρ_1	100%	85%	90%	100%	100%	100%
	$\rho 2$	51%	93%	100%	43%	50%	49%
	$\bar{\rho}$	75.6%	88.7%	95.0%	71.4%	74.5%	74.0%
3	*Mk*	6 745	5 106	**4 763**	8 246	7 920	**7 915**
	Δ_{Mk}		24%	29%		3.9%	4.0%
	Q	177.9	235.0	251.9	145.5	151.5	151.6
	TT	8,071	6,276	6,338	7,058	7,198	7,210
	Δ_{TT}		22%	21%		-1.9%	-2.1%
	$\rho 1$	100%	96%	98%	100%	100%	100%
	$\rho 2$	52%	100%	100%	37%	39%	39%
	$\rho 3$	68%	57%	78%	30%	36%	37%
	$\bar{\rho}$	73.4%	84.5%	92.0%	55.9%	58.5%	59.0%

The results produced for the two and three aisles configurations will now be commented, starting by the 1-class system (random storage). Global scheduling using TT–LSA reduces the makespan with respect to the results produced by Ind–Seq by 23% and 24% for the configurations having two and three aisles, respectively. EF–LSA obtained even better results, reducing the makespan by 28% and 29% for the cases with two and three aisles, respectively. These results are extremely important because not only do they show that a multi-aisle system cannot be approximated by many single-aisle systems, as is generally accepted in the AS/RS literature, but they also quantify the error incurred when accepting such a hypothesis. Still for the 1-class system, we can observe in Table 2 that both TT–LSA and EF–LSA produced similar reductions with respect to Ind–Seq in terms of the cranes' total travel time (between 19% and 22%). These improvements impact the cranes' average utilization, which increased from 75.6% and 73.4%, for Ind–Seq, up to 95.0% and 92.0% for EF–LSA for two and three aisles respectively. All in one, the system productivity, given by Q, increases with the number of aisles as expected, but the choice of the sequencing algorithm is also of paramount importance.

For the 2-class storage policy, global scheduling produced reductions in the makespan of 3.5% up to 4.0% with respect to independent sequencing. However, in the 2-class case, these makespan reductions were achieved at a cost. Indeed, the crane's total travel times produced by both TT–LSA and EF–LSA increased slightly with respect to the ones produced by Ind–Seq. Furthermore, if we compare makespan values produced by both the one and two class strategies, one may observe that for two and three aisles, it generally increases. This behaviour is unexpected because the classic literature in single-aisle AS/RS demonstrated that the makespan reduces when the number of classes increases. This behavior, only observed in multi-aisle configurations under *within* space allocation policy, is simply explained by the location of many high velocity SKUs in the same aisle. This leads to an important unbalance in the crane's utilization between aisles. For example, in the 3-aisles case, average crane utilizations for the one-class strategy are 73.4, 84.5 and 92.0 for Ind–Seq, TT–LSA, and EF–LSA respectively while these corresponding average values are 55.9, 58.5 and 59.0% for the 2-class case. These results confirm that most of the requests are sent to the crane where the highest demand products are located, creating an important queue, while the others cranes are clearly underused.

Results for the across aisle storage policy

Table 3 presents numerical results when products are allocated to locations according to an across aisle strategy. Note that the results for the single-aisle case are not reported in Table 3 because they are the same as those in Table 2 (within and across policies only makes sense when more than one aisle are considered). Again, for one aisle both Ind–Seq and TT–LSA gave the same results, with EF–LSA being almost equal.

Table 3. Performance comparison between independent and global scheduling (across aisle storage strategy)

# of aisles		1-class			2-class		
		Ind–Seq	TT–LSA	EF–LSA	Ind–Seq	TT–LSA	EF–LSA
2	Mk	8 005	6 874	**6 605**	6 694	**6 262**	6 328
	Δ_{Mk}		14%	17%		6.4%	5.4%
	TT	8,051	6,418	6,448	6,301	5,849	5,880
	Q	149.9	174.6	181.7	179.3	191.6	189.6
	Δ_{TT}		20%	20%		7.1%	6.6%
	ρ_1	100%	100%	100%	95%	100%	100%
	$\rho 2$	85%	93%	98%	100%	100%	100%
	$\bar{\rho}$	92.5%	96.6%	99.2%	97.5%	100%	100%
3	Mk	5 387	4 410	**4 405**	4 573	**4 296**	4 318
	Δ_{Mk}		18%	18%		6.0%	5.5%
	Q	222.8	272.1	272.4	262.4	279.3	277.9
	TT	8,258	6,407	6,415	6,393	5,864	5,893
	Δ_{TT}		22%	22%		8.2%	7.8%
	$\rho 1$	94%	100%	100%	96%	100%	100%
	$\rho 2$	86%	100%	97%	93%	95%	97%
	$\rho 3$	99%	97%	100%	99%	96%	93%
	$\bar{\rho}$	93.1%	98.8%	98.8%	96,1%	97.3%	96.8%

Unlike the within aisle case, in an across configuration, global scheduling methods dominate Ind–Seq for all the tested configurations as well as for the makespan, productivity, and the total travel time. When a 1-class across aisle storage strategy is used, managing aisles in a global manner with TT–LSA leads to makespan reductions of 14% and 18%, for two and three aisles respectively. These improvements are of about 17% and 18% for EF–LSA. Again, in the 1-class case and when managing the system globally with TT–LSA or EF–LSA, important reductions ranging from 20% to 22% of the total travel times are obtained.

As in the within strategy, results produced for the 2-class cases are less impressive but still quite appealing. Makespan reductions achieved by TT–LSA and EF–LSA ranged between 5.4% and 6.4%. However, with the across -aisle storage, both TT–LSA and EF–LSA succeeded in reducing the total travel time for the 2-class system (between 6.6% and 8.2%), which was not the case for the within strategy.

As per the crane's average utilization, the *across* strategy significantly improved the results produced by the *within* strategy for all tested configurations. This better overall cranes utilization's percentage is due to a better request balancing among the aisles as shown by the individual percentages of crane's utilization.

Comparison between within and across policies

Finally, it is worth explicitly comparing the system's performance under a *within* and an *across* policy. To this end we have built Table 4, which displays in columns under header *Mk* the values previously reported by Tables 2 (*within*) and 3 (*across*) for each configuration and sequencing approach. Column *Δ%* reports, in percentage, the makespan's reduction produced by a given method using an *across* policy when compared to the makespan produced by the same method using a *within* policy. Table 4 confirms that, despite the number of aisles and the number of classes considered, *across* storage leads to a much lower makespan than *within*, provided that the same sequencing algorithm is used.

Table 4. Impact of within and across storage policies on makespan

# of Aisles	# of classes	Sequencing approach	*Mk Within*	*Mk Across*	*Δ%*
2	1	Ind–Seq	9 712	8 005	17.5%
		TT–LSA	7 503	6 874	8.4%
		EF–LSA	6 968	6 605	5.2%
	2	Ind–Seq	9 368	6 694	28.5%
		TT–LSA	9 037	6 262	30.7%
		EF–LSA	9 031	6 328	29.9%
3	1	Ind–Seq	6 745	5 387	20.1%
		TT–LSA	5 106	4 410	13.6%
		EF–LSA	4 763	4 405	7.5%
	2	Ind–Seq	8 246	4 573	44.5%
		TT–LSA	7 920	4 296	45.8%
		EF–LSA	7 915	4 318	45.4%

7. Conclusions

In this paper we demonstrated that globally sequencing an *m*-aisle system instead of independently sequencing *m* single-aisle systems leads to important reductions in the makespan required to complete a set of orders. In fact, we observed that globally sequencing leads to reductions ranging from 14% up to 29% when 1-class storage is used, and from 3.5% up to 6.4% in the 2-class case. We also compared *within-aisle* to *across-aisle* storage strategies, for system configurations including two and three aisles and using class-based storage location policies with one and two zones. Our experiments showed that, for the same configuration and sequencing algorithm, *across-aisle* storage outperforms *within-aisle* storage, achieving makespan reductions ranging from 5.2% up to 45.8%. To the best of our knowledge, this is the first time that such results are presented and, due to the significant potential improvements revealed by our numerical simulations, we believe this justifies further researches devoted to multi-aisle AS/RS. We believe that the impact of the system's configuration (rack length and height, number of aisles and of cranes) on its performance as well as its impacts on the sequencing methods needs to be studied. The storage assignment of a multi-aisle AS/RS also presents interesting challenges, especially when one considers cranes' breakdowns and maintenance activities.

Acknowledgements

This research was partially funded by grants [OPG 0293307 and OPG 0172633] from the *Canadian Natural Sciences and Engineering Research Council* (NSERC). This support is gratefully acknowledged. The authors are grateful to two anonymous referees for their useful and constructive comments and suggestions.

References

Ashayeri J., Gelders L. & Van Wassenhove L., A microcomputer-based optimization model for the design of automated warehouses. *International Journal of Production Research*, 23, **1985**, 825-839.

Carlo H. J. & Vis I. F. A., Sequencing dynamic storage systems with multiple lifts and shuttles. *International Journal of Production Economics*, 140, **2012**, 844-853.

Ekren B. Y. & Heragu S. S., Simulation based performance analysis of an autonomous vehicle storage and retrieval system. *Simulation Modelling Practice and Theory*, 19, **2011**, 1640-1650.

Gagliardi, J-P., Renaud, J., Ruiz, A. On storage assignment policies for unit-load automated storage and retrieval systems. *International Journal of Production Research*, 50, **2012**, 879-892.

Gagliardi, J-P., Renaud, J., Ruiz, A. On sequencing policies for unit load automated storage and retrieval systems. *International Journal of Production Research,* 52, **2014a**, 1090-1099.

Gagliardi, J-P., Renaud, J., Ruiz, A. A simulation modelling framework for multiple-aisle automated storage and retrieval systems. *Journal of Intelligent Manufacturing*, 25, **2014b**, 193-207.

Han, M.-H.; McGinnis, L. F.; Shieh, J. S. and White, J., On sequencing retrievals in an automated storage / retrieval system. *IIE Transactions*, 19, **1987**, 56-66.

Hausman W. H., Schwarz L. B. & Graves S. C., Optimal storage assignment in automatic warehousing systems. *Management Science*, 22, **1976**, 629-638.

Houshyar A. & Chung I., Using simulation to compare different automated storage/retrieval system designs. *Computers & Industrial Engineering*, 21, **1991**, 629-633.

Hwang H. & Ko C. S., A study on multi-aisle system served by a single storage/retrieval machine. *International Journal of Production Research*, 26, **1988**, 1727-1737.

Koh S.-G., Kwon H.-M. & Kim Y.-J., An analysis of the end-of-aisle order picking system: Multi-aisle served by a single order picker. *International Journal of Production Economics*, 98, **2005**, 162-171

Kulturel S., Ozdemirel N. E., Sepil C. & Bozkurt Z., Experimental investigation of shared storage assignment policies in automated storage/retrieval systems. *IIE Transactions*, **1999**, 31, 739-749.

Lee S. G., De Souza R., Ong E. K., Simulation modelling of a narrow aisle automated storage and retrieval system (AS/RS) services by rail-guided vehicles. *Computers in Industry*, **1996**, 30, 241-253.

Lee Y. H., Lee M. H., & Hur S., Optimal design of rack structure with modular cell in AS/RS. *International Journal of Production Economics*, 98, **2005**, 172-178.

Lee, H.F. & Schaefer, S.K., Retrieval sequencing for unit-load automated storage and retrieval systems with multiple openings. *International Journal of Production Research*, 34, **1996**, 2943-2962.

Lee H. F. & Schaefer S. K., Sequencing methods for automated storage and retrieval systems with dedicated storage. *Computers and Industrial Engineering*, 32, **1997**, 351-362.

Lerher T., Design and evaluation of the class-based multi-aisle AS/RS. *International Journal of Simulation Modelling*, **2006**, 5, 25-36.

Lerher T., Edl M. & Rosi B., Energy efficiency model for the mini-load automated storage and retrieval systems. *International Journal of Advanced Manufacturing Technology*, 70, **2014**, 97-115.

Lerher T., Potrc I., Sraml M. & Tollazzi T., Travel time models for automated warehouses with aisle transferring storage and retrieval machine. *European Journal of Operational Research*, 205(3), **2010a**, 571-583.

Lerher T., Sraml M., Kramberger J., Potrc I., Borovinsek M. & Zmazek B., Analytical travel time models for multi aisle automated storage and retrieval systems. *International Journal of Advanced Manufacturing Technology*, 30, **2006**, 340-356.

Lerher T., Sraml M. & Potrc I., Simulation analysis of mini-load multi-shuttle automated storage and retrieval systems. *International Journal of Advanced Manufacturing Technology*, 54, **2011**, 337-348.

Lerher T., Sraml M., Potrc I., & Tollazzi T., Travel time models for double-deep automated storage and retrieval systems. *International Journal of Production Research,* 48, **2010b**, 3151-3172.

Malmborg C. J., Rule of thumb heuristics for configuring storage racks in automated storage and retrieval systems design. *International Journal of Production Research*, **2001**, 39, 511-527

Malmborg C. J., Conceptualizing tools for autonomous vehicle storage and retrieval systems. *International Journal of Production Research*, **2002**, 40, 1807-1822

Malmborg C. J., Design optimization models for storage and retrieval systems using rail guided vehicles. *Applied Mathematical Modelling*, 27, **2003**, 929-941

Petersen II C. G. & Schmenner R. W., An evaluation of routing and volume-based storage policies in a order picking operation. *Decision Science*, 30, **1999**, 481-501.

Roodbergen K. J. & Vis I. F.A., A survey of literature on automated storage and retrieval systems. *European Journal of Operational Research*, 194, **2009**, 343-362

Van den Berg J. P., Analytic expressions for the optimal dwell point in an automated storage/retrieval system. *International Journal of Production Economics*, 76, **2002**, 13-25.