International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

Optimising the storage assignment and order-picking for the compact drive-in storage system

David Revillot-Narváez, Francisco Pérez-Galarce & Eduardo Álvarez-Miranda

To cite this article: David Revillot-Narváez, Francisco Pérez-Galarce & Eduardo Álvarez-Miranda (2019): Optimising the storage assignment and order-picking for the compact drive-in storage system, International Journal of Production Research, DOI: 10.1080/00207543.2019.1687951

To link to this article: https://doi.org/10.1080/00207543.2019.1687951

Published online: 14 Nov 2019.

International Journal of Production Research, 2019
https://doi.org/10.1080/00207543.2019.1687951

Optimising the storage assignment and order-picking for the compact drive-in storage system

David Revillot-Narváez[a], Francisco Pérez-Galarce[b] and Eduardo Álvarez-Miranda[a,c*]

[a]*Department of Industrial Engineering, Faculty of Engineering, Universidad de Talca, Curicó, Chile;* [b]*Department of Computer Sciences, Pontificia Universidad Católica de Chile, Santiago, Chile;* [c]*Instituto Sistemas Complejos de Ingeniería, Santiago, Chile*

(*Received 9 March 2018; accepted 24 October 2019*)

One of the most common systems in non-automated warehouses, is drive-in pallet racking with a shared storage policy (which is usually based on the duration-of-stay). Such scheme targets towards an efficient use of storage space, since its operation costs are directly related to the size and layout of the warehouse. In this paper, two mathematical programming models and two greedy-randomised based heuristics for finding (nearly) optimal storage and retrieval operation sequences for this type of storage system are proposed. The computational effectiveness of the proposed approaches is measured by considering two sets of synthetic instances. The obtained results show that the proposed heuristics are not only able to compute high-quality solutions (as observed when being compared with the optimal solutions attained by the mathematical programming models), but it is also capable of providing solutions in very short running times even for large instances for which the mathematical programming model failed to find feasible solutions. At the light of these results, the best heuristic is also tested using a rolling-horizon planning strategy in a real-world case study, obtained from a Chilean company. It turns out that the attained results are more effective than the company's current storage policy.

Keywords: storage systems; warehousing systems; combinatorial optimisation; storage and retrieval heuristic; shared storage policy

1. Introduction

Warehouses play a central role throughout many, if not all, of the supply chains (see, e.g. Roodbergen, Vis, and Taylor 2015; de Koster, Johnson, and Roy 2017). In this setting, storage processes have a direct impact on direct variable costs, calling for efficient management systems in order to gain advantage in a highly competitive global environment (see Gu, Goetschalckx, and McGinnis 2007). Moreover, an effective storage management also improves delivery times and reliability, which has been recognised, over the last two decades, as a key performance driver (see, e.g. Gray, Karmarkar, and Seidmann 1992). From an economic point of view, storage and warehousing services account for approximately 15% of total logistics costs in developed countries such as Germany (Handfield et al. 2013).

In this regard, an efficient assignment and retrieval of orders, is fundamental for reducing logistic expenses (van den Berg and Zijm 1999; Staudt et al. 2015). Assignment and retrieval processes can be improved by using different decision-making tools (see, e.g. Cormier and Gunn 1992; de Koster, Le-Duc, and Roodbergen 2007; Gu, Goetschalckx, and McGinnis 2007; Manzini, Bozer, and Heragu 2015, and the references therein). A key aspect that must be considered, prior to applying an optimisation tool or methodology, is the definition of the storage system. Currently, storage systems fall into two main groups, conventional and compact systems.

In a conventional (known as 2-dimensional or 2D) automated system, unit loads are stored in single-deep racks by an automatic storage/retrieval (S/R) machine operating in the aisles between the racks (see Roodbergen and Vis 2009, for further details on S/R systems). The advantages of this system include labour-cost savings, increased reliability and reduced error rates. Its main disadvantages are the high initial investment, the difficulty of changing system design, and the large amount of space occupied by the aisles. Recent research regarding conventional S/R automated systems (AS/RS), has considered design decisions (Bortolini et al. 2015), storage location assignment and sequencing for multi-aisle layout (Gagliardi, Renaud, and Ruiz 2015; Yang et al. 2015, 2017; Zou et al. 2016), travel-time optimisation models (Lerher et al. 2010; Xu et al. 2015; Lerher 2016), and Dwell-point location (Hale et al. 2015).

One of the technological alternatives to conventional storage systems is the so-called *compact* storage system, also known as 3-dimensional or 3D. These have become increasingly popular for storing products because they provide a solution to the space and storage density problems (Hu et al. 2005). Such benefit is possible due to the compaction of multi-deep

*Corresponding author. Email: ealvarez@utalca.cl

racks, which results in rack layouts with less aisles. There are two ways to operate in this type of systems; either manually with forklifts, or using an automated alternative. The most common form of automation is an S/R machine, which handles the rack's horizontal and vertical movement, in conjunction with an orthogonal transport mechanism (satellite machine and/or pallet shuttle) that handles depth movements. In this system, the satellite connected to the S/R machine moves within a lane to store or retrieve the unit loads, while the S/R machine waits at the front of the lane until the satellite machine returns (the reader is referred to Zaerpour, Yu, and de Koster 2015, for a detailed description of the system). Please note that in the remainder of this paper, unit loads will be also referred to as *pallets*, since these are the most common unit loads in the application context considered in this paper. In Section 2 a comprehensive literature review regarding compact storage systems is presented.

In relation to the advantages and disadvantages of each method, and whether or when it is preferable to use automated systems rather than manual systems, it is possible to state the following: in large manufacturing companies, it is preferable to use conventional automated systems or compact automated systems because the daily picked volume is larger. Which of these two is chosen, hinges directly on the importance given to storage space. In smaller companies, where the daily picked volume is lower, and there is less investment capital available, the use of manual systems is common. The latter case is true, mainly, in logistics distribution centres where large warehouses are being replaced by more and smaller warehouses. This strategy aims at improving the geographical distribution of the logistical network and achieving economies of scale (de Koster, Le-Duc, and Roodbergen 2007). Some current estimates suggest that up to 80% of warehouses perform their order-picking operations manually (Grosse, Glock, and Neumann 2017).

Compact drive-in storage systems This is a system comprising a set of shelves that form interior loading lanes, with support rails for the pallets. To perform the storage and retrieval processes, the forklift must enter the lanes at the front of the rack; this method is known as drive-in pallet racking and it is described in Figure 1. Given this situation, pallet recovery is not a straightforward process as reshuffles are typically required. Reshuffling is defined as the process of removing the unit loads *blocking* the exit of a unit load that needs to be retrieved (Pazour and Carlo 2015). This system is widely used in both, refrigeration and freezing cold stores, where there is a need to make the best possible use of the temperature-controlled storage space.

In the pallet storage process, it is necessary to ensure that there are no empty spaces between stored pallets in order to maximise space utilisation. In the retrieval process, the operator shall seek for the minimum number of reshuffles. Note that a pallet is reshuffled when it is (i) at the same row, (ii) lower or equal tier, and (iii) greater depth, with respect to the pallet to be retrieved. According to this situation, the access to a pallet is hindered not only by the pallets that are in front of the unit load that needs to be retrieved, but also by the pallets that are stored at lower tiers of the same row. This happens because the forklift must always enter at the first tier and raise the fork to gain access to higher tiers. This procedure is clearly different when compared to automated compact systems, where the lower tiers do not influence the retrieval process, due to the incorporation of a transport mechanism for *in-depth* movements. The problem of finding the storage and retrieval operations with the minimum number of reshuffles will be denoted as optimal Drive-in pallet racking problem (DIPR).

Evidently, bottlenecks are the main problem caused by inefficient storage assignments, as the truck loading process is slowed by the retrievals. The most widely used key performance indicators (KPIs) to measure and control these inefficiencies are order-picking accuracy, order fill-rate, rate of return, and total recovery time. It is important to note that the latter KPI, stands as an indicator that relates the other measures; hence, achieving total recovery times ensures an overall good performance of the system. When a truck arrives at the company for picking up products, fast retrieval of customer orders reduces the truck's waiting time. This means that the truck is more likely to be loaded within its expected arrival and departure timeframe. Furthermore, a possible reduction in each truck's waiting time makes possible to serve more trucks during a shift, thereby increasing the overall performance and utilisation of the system.

Another relevant aspect in frozen food supply chain is the energy efficiency of the storage process, and how it impacts when analyzing the sustainability performance of the whole chain (see Yakovleva, Sarkis, and Sloan 2012). In this context, and complementing the arguments presented above, rapid order retrieval minimises open door times, allowing temperature control and minimising energy costs. Moreover, it also reduces the risk of loss of quality as the foodstuff, as the cold chain is better preserved, enhancing the efficiency of the whole production and supply chain. This situation shows that an efficient operation in the storage and warehousing processes is crucial for a cost effective supply chain operation and also for an adequate procurement of quality standards. Therefore, developing tools for optimising storage and the retrieval operations appears as a fundamental task for improving the supply chain competitiveness. Such tools are particularly relevant for the case of DIRP systems, where the human factor plays an important role in their performance.

In order to understand the context in which this work took place developed, and the performance gaps, please take into account that according to the data gathered in field studies at the considered company,the current operation is characterised by the following indicators: (i) reshuffling a pallet takes, in average, one minute; (ii) over 50% of the trucks to be loaded require pallet reshuffling; (iii) in some cases, pallet retrieval can take up to two hours due inefficient storage assignment;

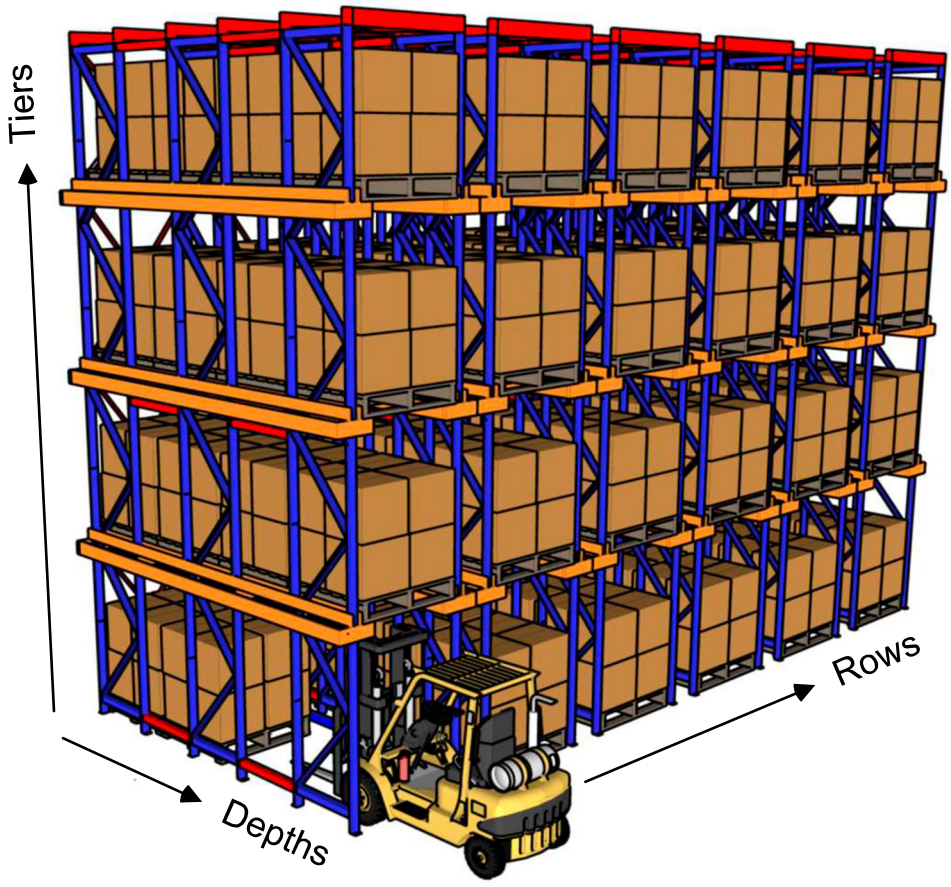

Figure 1. Drive-in pallet racking.

(iv) of the total time recorded, 29% was spent on pallet reshufflings (although in one observation this proportion rose up to 63.7%). Bear in mind that the number of loadings and unloadings depends on the size of the company and the business model (storage or production-storage). Nonetheless, the operations carried out by the company under study are standard for companies in the industry, so the observed data is a good approximation for the sector.

Contribution and paper outline The main goal of this work is to develop a decision aid tool for improving the performance of the storage and retrieval operations in a temperature controlled warehouse for food storage. The operational objective corresponds to the reduction of movements in the warehouses because when the movements increase, the probability of damage a pallet is bigger. Moreover, each movement generates a loss of energy (cold chain) since in each movement, the pallets are taken out of the warehouse, increasing the temperature of both the warehouse and the pallet. Therefore, the contribution of this paper is the development of a novel optimisation tool, based on mathematical models and ad-hoc algorithms, for optimising sequencing operations in drive-in pallet racking systems, characterised by a shared storage policy based on duration-of-stay. The optimisation tool relies on a mathematical representation of the storage and retrieval operation, that captures the dynamic interaction between the storage and retrieval/reshuffle processes. To the best of our knowledge, this is the first attempt for developing a computational system for aiding the operation of this type of storage systems. As a matter of fact, the closest examples in the literature correspond to optimisation tools for compact storage systems with some level of automation for vertical, horizontal and orthogonal movements (see, e.g. Mirzaei, de Koster, and Zaerpour 2017; Yu et al. 2017; Zaerpour, Yu, and de Koster 2015).

The proposed optimisation tool is comprised by two novel integer programming (IP) formulations and a greedy-randomised heuristic (GR), which is improved in a Greedy randomised adaptive search procedure (aka GRASP). By means of comparing the solutions attained by the heuristics with those obtained by the IP models (for which a certificate of quality is available, i.e. the primal/dual gap), the effectiveness of the heuristics is measured. Computational results show that the GR algorithm is capable of providing excellent solutions within short running times in the case of random instances. Moreover, the GR and GRASP are able to deliver solutions, using a rolling-horizon scheme, for real-life instances, even in cases where

Table 1. Overview of the literature on multi-deep compact systems for operations, planning and control ('A': Automated operation, 'M': Manual operation, ILP: Integer Linear Programming, MILP: Mixed Integer Linear Programming, MINLP: Mixed Integer Nonlinear Programming).

Reference	System	Opt.	Research issue	Methodology
Stadtler (1996)	Satellite-based AS/RS	A	Reservation of resources at order time, managing busy period workload	MILP, heuristic
Yu and de Koster (2009)	Conveyor-based (paired) AS/RS	A	Optimal zone boundaries for two product classes	MINLP
Alfieri et al. (2012)	GridFlow	A	Gridflow with limited number of vehicles, optimally dispatch AGVs	Heuristic
Bessenouci, Sari, and Ghomri (2012)	Flow-Rack AS/RS	A	Sequencing storage and retrieval	Heuristic
Cardin et al. (2012)	Flow-Rack AS/RS	A	Evaluation of In-Deep Class Storage	Simulation
Yu and de Koster (2012)	Conveyor-based (paired) AS/RS	A	Sequencing storage and retrieval	Heuristic
Zaerpour, de Koster, and Yu (2013)	Conveyor-based (single) AS/RS	A	Choice of storage strategies	MINLP, Heuristic
Gue et al. (2014)	GridStore	A	Deadlock free decentralised control scheme, effect of WIP and escorts on the throughput rate	Discrete time simulation
Zaerpour, Yu, and de Koster (2015)	AS/RS	A	Lane-sharing storage policy	ILP, heuristic
Chen, Li, and Gupta (2016)	Flow-Rack AS/RS	A	Sequencing storage and retrieval	Heuristic
Zou, de Koster, and Xu (2016)	RCSRS	A	Evaluating dedicated versus, shared storage policies	Semi-open queuing network
Yu et al. (2017)	Grid-Based	A	Effect of simultaneous and block movement of items and escorts	ILP
Mirzaei, de Koster, and Zaerpour (2017)	Grid-Based	A	Simultaneous multi-load retrieval	Monte Carlo simulation, heuristic
Boysen, Briskorn, and Emde (2017)	Mobile rack	A	Sequencing of picking orders	Heuristic
Yu and Yu (2019)	AS/RS with Dwell Point	A	AS/RSs with I/O stations at opposite ends of the aisle	Analytical Models
Our work	Drive-in	M	Sequencing storage and retrieval	ILP, heuristic

the IP-based approaches fail to compute feasible solutions within one hour. From a managerial point of view, the obtained results show that the proposed GRASP provides solutions that outperform the company's current storage policy.

The remainder of the paper is organised as follows. In Section 2 a literature review is presented. The proposed IP models, as well as the GR-based algorithms for the DIPR, are presented in Section 3. In Section 4, computational results on random instances are reported. In Section 5, the case study is presented and the corresponding results are discussed. Lastly, conclusions and paths for future work are drawn in Section 6.

2. Literature review

Order retrieval in storage processes is currently a growing scientific field, due to the interest of warehouses and distribution centres in minimising costs and improving productivity. This process of product retrieval from storage areas is one of their most labour-intensive and cost-intensive operations (Tompkins et al. 2010). Within this context, the conventional 2D and 3D storage system is a valuable alternative for reducing retrieval costs because access to the unit loads is direct. This is possible because the system consists of a set of single-deep or double-deep shelves separated by lanes that occupy a significant percentage of the surface.

The increase of population, as well as the rise of real estate costs near urban areas, have made compact storage systems increasingly popular. In this type of system there are two levels of decision-making: warehouse design (tactical) and planning (operational). The tactical decisions are closely associated with the selection and optimisation of the storage system. The primary objective is to maximise performance and storage capacity. The operational decisions focus on operational planning and control. Some of the objectives are to minimise waiting times, response times and resource idle time. Table 1 shows the most important studies relating to multi-deep compact storage systems (for further details regarding the classifications and features of the storage systems, see Azadeh, de Koster, and Roy 2017).

As Table 1 shows, the previous papers only studied automated compact systems, and none considered manually operated systems, which are highly employed in practice (see, e.g. Altarazi and Ammouri 2018 for a recent example of a manual-operated warehouse design approach, where the importance of manual-operated storage systems is highlighted).

For comparison purposes, our study, which addresses the compact drive-in storage system with an exact and heuristic approach, appears at the end of the table.

From an operative point of view, drive-in pallet racking is quite similar to container handling models. In the case of container handling, when a cargo ship arrives at the port, the containers must be first unloaded and stacked in a marine terminal with the aim of minimising movements during their retrieval and subsequent loading onto the ship. Based on this problem, Yang and Kim (2006) developed a genetic algorithm to minimise container reshuffling. Wan, Liu, and Tsai (2009) proposed the first ILP formulation that minimises the number of export containers reshuffles for the static problem of emptying a stack in which new arrivals are not considered. Park et al. (2011) proposed a simple heuristic with weighted decision criteria for the stacking of containers at an automated terminal, while Caserta, Schwarze, and Voß (2012) studied the relocation problem using an ILP model for a 2-dimensional stacking area in a multi-period environment. The authors in Gharehgozli et al. (2014) developed a decision-tree heuristic for a shared stacking policy with the aim of minimising the number of reshuffles.

A crucial issue in S/R systems, both autonomous and manual, corresponds to time models that analytically allow to approximate time performance of storage systems (time travel and S/R operations). The authors in Bozer and White (1984), develop one of the first travel-time models for a basic AS/RS systems. Recent examples on modern storage systems can be found in Lerher (2018), where analytical travel-time models for aisle changing shuttle carriers in autonomous vehicle S/R system are proposed, and in Xu et al. (2019), where the authors propose a travel-time model for a 3D AS/RS system combining conveyors with a crane for depth and horizontal/vertical transportation, respectively. In our setting, S/R operation times are surrogated by reshuffling operations and, therefore, by minimising reshuffles, we minimise the total S/R operation time.

Similarly, storage volume assignment is another relevant issue in the operation of warehouses. One of the earliest references on this topic corresponds to Hausman, Schwarz, and Graves (1976), where the storage area assigned to a product depends on its average inventory level, although back-up spaces are reserved for storing the product's maximum inventory. More recently, in Yu and de Koster (2013), the authors present a volume-based dedicated storage policy, similar to the one presented in Hausman, Schwarz, and Graves (1976), and compare it with the ABC-class-based storage policy, showing that the former is not only more practical than the later, but also more efficient. Furthermore, current managerial challenges, as e-commerce, have also been addressed; for instance, new storage volume management strategies, specially devised for an e-commerce warehouse operation, characterised by very large number of small sized manually operated orders and returns, are proposed in Bahrami, Aghezzaf, and Limère (2019) and Calzavara et al. (2019). As it will be shown in detail in the next section, our model does not explicitly include storage volume assignment; nonetheless, it could be included by an-hoc adaptation of the model.

3. Optimisation models and algorithms for DIPR

This section presents a modelling and algorithmic approach for solving the practical problem described above. Subsection 3.1 presents common notation and definitions of the proposed formulations. Subsections 3.2 and 3.3 respectively present the two ILP models proposed in the this paper, which are referred to as ILP1 and ILP2, respectively. Lastly, Subsection 3.4 presents a randomised heuristic for the solving the problem under study.

3.1. Problem statement, notation and basic definitions

In order to present the problem statement, consider the small rack shown in Figure 2, comprised by a tri-tier three-deep row and a drive-in operation scheme. There are six pallets, each represented by a square. For modelling purposes, let a *stage* be an event in the planning horizon, there are retrieval stages and storage stages. Therefore, a stage is not a time unit nor it is measured by a time unit, but rather it is a retrieval or storage action. In the problem we are considering, pallets were stored using the FIFO management technique, therefore their *retrieval stage* is in the ascending order of their ranking numbers (which is shown in the squared tags); e.g. the pallet with the tag corresponding to 5 means that this is the pallet that this is the 5th pallet to be retrieved. Figure 2 shows the rack emptying sequence. In the first stage (rack on the left), retrieval of pallet 1 does not generate any reshuffles. However, for retrieving pallet 2 (rack in the middle) it is necessary to reshuffle pallets 3, 5 and 6. After the retrieval of pallet 2, one can observe that no reshuffling is needed for retrieving the remaining pallets in the later stages. Hence, as the forklift must always enter at the first tier and raise the fork in order to access to a higher tier, retrieving a pallet is constrained by the pallets that are in front of the unit load that needs to be retrieved, and also by those that are stored at lower tiers of the same row, i.e. reshuffles are required. However, reshuffles are undesired as they typically induce load damages, problems in the preservation of the cold chain, time inefficiencies and other operational issues; therefore, the drive-in pallet racking problem (DIPR) can be stated as the *problem of finding the storage and retrieval*

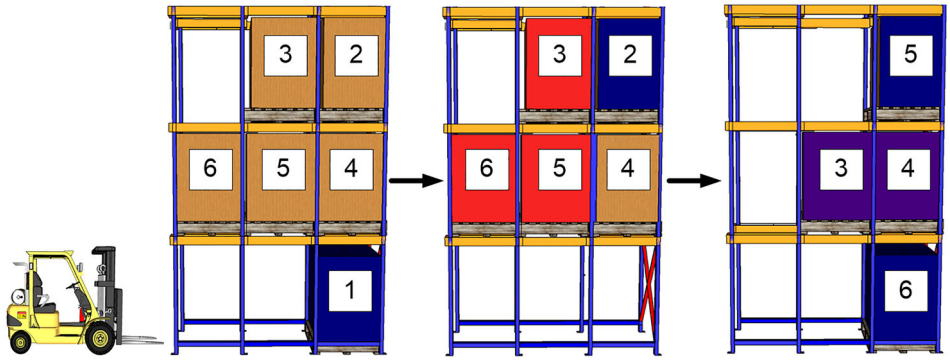

Figure 2. Feasible solution example.

operations of a single forklift with the minimum number of reshuffles given an initial rack configuration, and a set of pallet arrivals (i.e. pallets must be stored) and pallet departures (i.e. pallets that must be retrieved).

The optimisation goal embodied by the DIPR can be achieved if during the retrieval process, the pallets that are blocking the exit of others are reshuffled to the *best* locations. However, resulting optimisation problem is quite complex because of the tremendous number of configuration and the temporal dependency among them. The reader should know that, until now, there has not been any explicit optimisation study capturing the dynamic interaction of pallets in a compact drive-in storage system.

The sets and parameters that are used in the problem definition are the following:

S	Set of storage and retrieval stages.
I	Set of pallets.
B	Set of rows.
T	Set of tiers.
P	Set of depths.
$A_{is} \in \{0,1\}$	1, if pallet $i \in I$ arrives at stage $s \in S$, 0 otherwise.
$D_{is} \in \{0,1\}$	1, if pallet $i \in I$ departs at stage $s \in S$, 0 otherwise.
$X_{ibtp} \in \{0,1\}$	location (b, t, p) of pallet $i \in I$ in the initial configuration, $b \in B$, $t \in T$, $p \in P$.

For ease of understanding of the mathematical models, the definition of some auxiliar sets Φ_1, Φ_2 and Ω are required for modelling the combinatorial nature of the storage and retrieval operations in the three-dimensional configuration. In order to improve the readability of the model, let function $range(\cdot,\cdot)$ be such that $range(A_{is}, D_{is})$ returns a subset $t \subseteq S$ which considers the stages between arrival and departure of a pallet i. The definition of Φ_1, Φ_2 and Ω is given by

$$\Phi_1 \quad \{\langle s,j,i\rangle \mid s \in range(A_{is}, D_{is}), j \in I, i \in I : D_{js} = 1 \wedge i \neq j\}.$$
$$\Phi_2 \quad \{\langle s,i,k\rangle \mid s \in range(A_{is}, D_{is}) \cap range(A_{ks}, D_{ks}), i \in I, k \in I : i \neq k\}.$$
$$\Omega \quad \{\langle s,i,b,t,p\rangle \mid s \in range(A_{is}, D_{is}), i \in I, b \in B, t \in T, p \in P\}.$$

Therefore, for a given triplet $\langle s,j,i\rangle$, the subset Φ_1 corresponds to the stages where the pallets $i \in I$ stay in the warehouse and the pallet $j \in I$ is departed. From a structural point of view, the following decision variables are used:

$x^s_{ibtp} \in \{0,1\}$	it takes value 1, if pallet $i \in I$ is at location (b, t, p) at stage $s \in S$, and 0 otherwise, with $b \in B$, $t \in T$, $p \in P$.
$y^s_{ji} \in \{0,1\}$	it takes value 1, if pallet i is reshuffled in the retrieval of j at stage s, and 0 otherwise, with $\langle s,j,i\rangle \in \Phi_1$.

Along with the elements presented above, and the overview of the literature presented in Section 2, there are additional assumptions and consideration that shall be taken into account in the following. For instance, no uncertainty of the input parameters (e.g. pallet location, pallet arrival or pallet departures) is assumed. Furthermore, the decision context considered in this paper assumes that the tasks are performed by a single forklift; nonetheless, as it will be explained below, a situation with multiple forklifts can be modelled by the proposed formulations through an ad-hoc adaptation.

3.2. An optimisation model for DIPR

Following the notation presented before, the ILP1 formulation is presented next. Note that ILP1 is a generalisation of the formulation presented in Wan, Liu, and Tsai (2009). In this formulation, reshuffles are restricted only to those unit loads that

are blocking the exit of the unit load that needs to be retrieved; this is why this model is also referred to as the *restricted* model.

In addition to the decision variables presented in previous subsection, the following variables are also required.

- $u^s_{ji} \in \{0,1\}$ it takes value 1, if the row index of pallet i is no less than that of pallet j at stage s, and 0 otherwise, $\langle s,j,i\rangle \in \Phi_1$.
- $v^s_{ji} \in \{0,1\}$ it takes value 1, if the row index of pallet i is no greater than that of pallet j at stage s, and 0 otherwise, $\langle s,j,i\rangle \in \Phi_1$.
- $z^s_{ji} \in \{0,1\}$ it takes value 1, if pallets i and j are at the same row at stage s, and 0 otherwise, $\langle s,j,i\rangle \in \Phi_1$.
- $w^s_{ji} \in \{0,1\}$ it takes value 1, if the tier index of pallet i is no greater than that of pallet j at stage s, and 0 otherwise, $\langle s,j,i\rangle \in \Phi_1$.
- $l^s_{ji} \in \{0,1\}$ it takes value 1, if pallet i is at the same row and the tier index is no greater than that of pallet j at stage s, and 0 otherwise, $\langle s,j,i\rangle \in \Phi_1$.

The objective function of ILP1 seeks for the minimisation of the number of reshuffles over the planning horizon and it is encoded by

$$\min\ Z_1 = \sum_{\langle s,j,i\rangle \in \Phi_1} y^s_{ji} \tag{1}$$

The constraint set presented below, (2)–(7), characterises the storage conditions. In this sense, constraints (2)–(3) ensure the duration-of-stay. Constraint (4) ensures that at most one pallet is stored in one location, while (5) ensures that there are no empty locations between stored pallets. Because the forklift enters the rack on the first tier, constraint (6) ensures that storage at a location is permitted by the lower tiers of a row.

$$\sum_{b\in B}\sum_{t\in T}\sum_{p\in P} x^s_{ibtp} = 1, \quad \forall\, i \in I,\ A_{is} \le s \le D_{is} \tag{2}$$

$$\sum_{b\in B}\sum_{t\in T}\sum_{p\in P} x^s_{ibtp} = 0, \quad \forall\, i \in I,\ s \in [1, A_{is}) \cup s \in (D_{is}, S] \tag{3}$$

$$\sum_{i\in I} x^s_{ibtp} \le 1, \quad \forall\, s \in S,\ b \in B,\ t \in T,\ p \in P \tag{4}$$

$$\sum_{i\in I} x^s_{ibtp} \le \sum_{i\in I} x^s_{ibt,p-1}, \quad \forall\, s \in S,\ b \in B,\ t \in T,\ 2 \le p \le P \tag{5}$$

$$\sum_{i\in I}\sum_{p\in P} x^s_{ibtp} \le \sum_{i\in I}\sum_{p\in P} x^s_{ibt'p} + 1, \quad \forall\, s \in S,\ b \in B,\ (t,t') \in T,\ t < t' \tag{6}$$

Similarly, the set of constraints presented below encodes the requirements of the retrieval and reshuffle operations. More precisely, the constraint set (7)–(11) determines whether the pallet i and the pallet to be retrieved j are in the same row in stage s, while the pair (12)–(13) determine whether pallet i is at a tier not higher than j in stage s. Likewise, constraints (14)–(16) account for the linearisation of quadratic factors. Moreover, these constraints also ensure that if i and j are in the same row, but i is in a tier not higher than the tier of j, in stage s, then it holds that $l^s_{ji} = 1$. The reshuffles resulting from the retrieval of j are modelled by (17)–(19). If the depth of i is greater than the one of j, then (18) forces $y^s_{ji} = 1$, and 19 is redundant. On the contrary, if the depth of i is less than the one of j, then (19) forces $y^s_{ji} = 0$, and (18) is redundant.

$$|B|\, u^s_{ji} \ge \sum_{b\in B}\sum_{t\in T}\sum_{p\in P} bx^s_{ibtp} - \sum_{b\in B}\sum_{t\in T}\sum_{p\in P} bx^s_{jbtp} + 1, \quad \forall\ \langle s,j,i\rangle \in \Phi_1 \tag{7}$$

$$|B|\, u^s_{ji} - |B| \le \sum_{b\in B}\sum_{t\in T}\sum_{p\in P} bx^s_{ibtp} - \sum_{b\in B}\sum_{t\in T}\sum_{p\in P} bx^s_{jbtp}, \quad \forall\ \langle s,j,i\rangle \in \Phi_1 \tag{8}$$

$$|B|\, v^s_{ji} \ge \sum_{b\in B}\sum_{t\in T}\sum_{p\in P} bx^s_{jbtp} - \sum_{b\in B}\sum_{t\in T}\sum_{p\in P} bx^s_{ibtp} + 1, \quad \forall\ \langle s,j,i\rangle \in \Phi_1 \tag{9}$$

$$|B|\, v^s_{ji} - |B| \le \sum_{b\in B}\sum_{t\in T}\sum_{p\in P} bx^s_{jbtp} - \sum_{b\in B}\sum_{t\in T}\sum_{p\in P} bx^s_{ibtp}, \quad \forall\ \langle s,j,i\rangle \in \Phi_1 \tag{10}$$

$$z^s_{ji} = u^s_{ji} + v^s_{ji} - 1, \quad \forall\ \langle s,j,i\rangle \in \Phi_1 \tag{11}$$

$$|T|\, w^s_{ji} \geq \sum_{b\in B}\sum_{t\in T}\sum_{p\in P} tx^s_{jbtp} - \sum_{b\in B}\sum_{t\in T}\sum_{p\in P} tx^s_{ibtp} + 1, \quad \forall\ \langle s,j,i\rangle \in \Phi_1 \tag{12}$$

$$|T|\, w^s_{ji} - |T| \leq \sum_{b\in B}\sum_{t\in T}\sum_{p\in P} tx^s_{jbtp} - \sum_{b\in B}\sum_{t\in T}\sum_{p\in P} tx^s_{ibtp}, \quad \forall\ \langle s,j,i\rangle \in \Phi_1 \tag{13}$$

$$l^s_{ji} - z^s_{ji} \leq 0, \quad \forall\ \langle s,j,i\rangle \in \Phi_1 \tag{14}$$

$$l^s_{ji} - w^s_{ji} \leq 0, \quad \forall\ \langle s,j,i\rangle \in \Phi_1 \tag{15}$$

$$l^s_{ji} - z^s_{ji} - w^s_{ji} \geq -1, \quad \forall\ \langle s,j,i\rangle \in \Phi_1 \tag{16}$$

$$y^s_{ji} \leq l^s_{ji}, \quad \forall\ \langle s,j,i\rangle \in \Phi_1 \tag{17}$$

$$y^s_{ji} \geq l^s_{ji} - 1 + \left(\sum_{b\in B}\sum_{t\in T}\sum_{p\in P} px^s_{ibtp} - \sum_{b\in B}\sum_{t\in T}\sum_{p\in P} px^s_{jbtp}\right) \Big/ |P|, \quad \forall\ \langle s,j,i\rangle \in \Phi_1 \tag{18}$$

$$\left(\sum_{b\in B}\sum_{t\in T}\sum_{p\in P} px^s_{jbtp} - \sum_{b\in B}\sum_{t\in T}\sum_{p\in P} px^s_{ibtp} + 1\right) \Big/ |P| \leq 1 - y^s_{ji}, \quad \forall\ \langle s,j,i\rangle \in \Phi_1 \tag{19}$$

The following constraints, (20)–(21) are the so-called precedence constraints, and they ensure that the pallets that were not reshuffled retain their positions in the next stage (otherwise, these restrictions are redundant).

$$x^{s+1}_{ibtp} - x^s_{ibtp} \geq \sum_{j\in J: i\neq j} -y^s_{ji}, \quad \forall\ \langle s,i,b,t,p\rangle \in \Omega \tag{20}$$

$$x^s_{ibtp} - x^{s+1}_{ibtp} \geq \sum_{j\in J: i\neq j} -y^s_{ji}, \quad \forall\ \langle s,i,b,t,p\rangle \in \Omega. \tag{21}$$

Additionally, constraints

$$x^0_{ibtp} = X_{ibtp}, \quad \forall\, i \in I,\ b \in B,\ t \in T,\ p \in P, \tag{22}$$

are the inventory constraints and associate the initial configuration of the warehouse (which is an input parameter) with the corresponding decision variables. Finally, the nature of the decision variables is imposed by

$$x^s_{ibtp} \in \{0,1\}, \quad \forall\, s \in S,\ i \in I,\ b \in B,\ t \in T,\ p \in P \tag{23}$$

$$u^s_{ji}, v^s_{ji}, z^s_{ji}, w^s_{ji}, l^s_{ji}, y^s_{ji} \in \{0,1\}, \quad \forall\ \langle s,j,i\rangle \in \Phi_1. \tag{24}$$

The complexity nature of the DIPR is characterised by the following result.

PROPOSITION 1 *The DIPR is NP-Hard*

The correctness of the result follows fron the fact that the Container Stacking Problem is equivalent to the DIPR when $T = 1$; since the CSP is NP-hard, as shown in Avriel, Penn, and Shpirer (2000), then the DIPR is NP-Hard as well.

3.3. A generalised optimisation model for multiple storage systems

In this section an additional ILP model is presented which will be referred to as ILP2. Two clear differences are established with respect to ILP1. First, consider $\Re^{tp}_{t'p'} \in \{0,1\}$, a parameter that takes value 1 when the access to location (t,p) is blocked by location (t',p'). This new parameter provides flexibility so that the model can be adapted to multiple storage systems. Second, the problem is formulated using a pre-marshalling strategy. The pre-marshalling problem (which is decribed in Lee and Hsu 2007; Lee and Chao 2009; Bortfeldt and Forster 2012; Wang, Jin, and Lim 2015) is the problem associated to the repositioning of unit loads so that none or few reshuffles are needed when unit loads are loaded. In this way, the proposed

new formulation assumes that during retrieval stages unit loads stored in any rack location can be reshuffled (unrestricted model).

For the formulation ILP2, the following decision variables are required;

$\varphi^s_{ik} \in \{0,1\}$ it takes value 1, if pallet k is reshuffled in the reshuffle of i at stage s, and 0 otherwise, $\langle s,i,k\rangle \in \Phi_2$.

Considering the variables presented above, the constraints required for modelling the conditions of retrieval and reshuffle operations are given by

$$y^s_{ji} - \Re^{tp}_{t'p'}\left(x^s_{jbtp} + x^s_{ibt'p'}\right) \geq -1, \quad \forall\, b \in B, \langle s,j,i\rangle \in \Phi_1,\ (t,t') \in T,\ (p,p') \in P \tag{25}$$

$$\varphi^s_{ik} - \Re^{tp}_{t'p'}\left(x^s_{ibtp} + x^s_{kbt'p'}\right) \geq -2 + \sum_{\langle s,j,i\rangle \in \Phi_1} y^s_{ji}, \quad \forall\, b \in B,\ \langle s,i,k\rangle \in \Phi_2,\ (t,t') \in T,\ (p,p') \in P \tag{26}$$

$$\varphi^s_{ik} \leq y^s_{jk}, \quad \forall\ \langle s,j,i,k\rangle \in (\Phi_1 \cap \Phi_2)\,. \tag{27}$$

Likewise, constraints

$$\varphi^s_{ik} \in \{0,1\}\,, \quad \forall\ \langle s,i,k\rangle \in \Phi_2 \tag{28}$$

are required for modelling the nature of the new variable.

Consequently, the ILP2 formulation is given by

$$Z_2 = \min\{(1) \mid (2)\text{–}(6), (20)\text{–}(23), (25)\text{–}(28)\}\,.$$

Note that $\Phi_1 \cup A_{is} : A_{is} = 1$, also allows reshuffling operations in storage stages. The formulation of this such variant is given by

$$Z'_2 = \min\{(2)\text{–}(5), (20)\text{–}(23), (25)\text{–}(28)\}\,.$$

The reader should be aware that solving this variant is out of the scope of this paper, but it should be part of a further research agenda.

Instances of both, ILP1 and ILP2, are tackled by means of an off-the-shelf commercial solver; the resulting computational results are reported and discussed in Section 4.

3.4. A greedy randomised algorithm for DIPR

As it will be shown in the following section, the use of an off-the-shelf commercial solver is practical only for instances of limited size (instances with B$\geq$ 3 are quite challenging from a computational point of view). Therefore, the use of a heuristic approach is mandatory for tackling medium and large instances. In this paper, a greedy randomised (GR) algorithm, devised for finding approximate solutions to the DIPR variant modelled by ILP1, is presented in this section. GR algorithms are among the most effective strategies for solving optimisation problems composed by binary variables (such as the DIPR) and have been applied to a wide range of problems (Resende and Ribeiro 2014).

Initially, the routine described in Algorithm 1 begins by cycling through all of the stages of the planning horizon. In a storage stage, Algorithm 2 ensures that the pallet is placed in a feasible location, while in a retrieval stage the Algorithm 3 identifies and reshuffles the blocking pallets. In this way, Algorithms 2 and 3 construct a step-by-step solution. Once a solution is constructed (encoded by variables $x^{s^*}_{ibtp}$), the positions are tracked in order to determine the total number of reshuffles (represented by Z_3). The details of both algorithms will be presented in the remainder of this section.

Algorithm 1 Greedy Randomised Algorithm

$M = MaxIterations$
for $m = 1, ..., M$ **do**
  for $s \in S, i \in I$ **do**
    if $A_{is} = 1$ **then**
      $x^s_{ibtp} \leftarrow StorageAlgorithm(i)$
    end if
    if $D_{is} = 1$ **then**
      $x^s_{ibtp} \leftarrow RetrievalAlgorithm(i)$
    end if
  end for
  $Z_3 \leftarrow ReshuffleCount\left(x^s_{ibtp}\right)$
  if $Z_3 < Z^*_3$ **then**
    $Z^*_3 \leftarrow Z_3$
    $x^{s^*}_{ibtp} \leftarrow x^s_{ibtp}$
  end if
end for
return $Z^*_3, x^{s^*}_{ibtp}$

Storage algorithm The first step of the routine described in Algorithm 2 is to identify the set of candidate locations, denoted by C. Set C is comprised by all the elements (storage locations) e that can be incorporated into the partial solution being constructed, without destroying its feasibility. A location can be a storage candidate, if it meets conditions presented in Section 1. Once the candidate list has been finalised, the partial cost $c(e)$, of each element, is calculated. As explained before, in this paper, the partial costs are represented by the number of potential reshuffles that would be involved in the storage candidate e.

The next step is the creation of a restricted candidate list (RCL). To achieve this, an element e is included in the RCL if its partial cost $c(e)$ falls within the interval $[S_{\min}; S_{\min} + \alpha(S_{\max} - S_{\min})]$, where $S_{\min}$ and $S_{\max}$ are, respectively, the lowest and highest partial cost obtained, while α is a scalar that varies between 0 and 1. If the value of α is 0 there is no randomness in the choice, whereas a value of α equal to 1 means the choice is totally random. Next, the following solution element is randomly chosen from the RCL set.

Algorithm 2 Storage Algorithm

Require: i
$C \leftarrow \{\oslash\}$ (Set of candidates)
$S \leftarrow \{\oslash\}$ (Partial cost of candidates)
$RCL \leftarrow \{\oslash\}$ (Restricted candidate list)
Initialise set of candidates : $C \leftarrow E$
for $e \in C$ **do**
  $S \leftarrow S \cup \{c\,(e)\}$
end for
for $e \in C$ **do**
  if $S_{\min} \leq c\,(e) \leq S_{\min} + \alpha\,(S_{\max} - S_{\min})$ **then**
    $RCL \leftarrow RCL \cup \{e\}$
  end if
end for
$x^s_{ibtp} \leftarrow Random\,(RCL)$
return x^s_{ibtp}

Retrieval algorithm In retrieving a pallet, Algorithm 3 identifies the set of blocking unit loads R. When this set is not empty, the pallets are sorted from highest to the lowest retrieval stage, and then entered one by one as arguments into Algorithm 2. The last step is the retrieval of the incumbent pallet j.

Algorithm 3 Retrieval Algorithm

Require: j
$R \leftarrow \{\oslash\}$ (Set of pallets for reshuffling)
Initialise : $R \leftarrow I$
if $R \neq \oslash$ **then**
  $R \leftarrow Sort\,(R)$
  for $i \in R$ **do**
    $x^s_{ibtp} \leftarrow StorageAlgorithm\,(i)$
  end for
end if
$x^s_{ibtp} \leftarrow Retrieval\,(j)$
return x^s_{ibtp}

3.5. A reactive GR adaptive search procedure (GRASP)

A natural alternative for boosting Greedy Randomised algorithms is using GRASP. It is a multi-start metaheuristic that allows us to manage the *exploration* and the *explotation* in the space of feasible solutions. This class of algorithms considers two stages. Firstly, in each iteration, it is generated a solution through a Greedy Randomised procedure; and secondly, the solution is polished by applying a local search procedure.

To avoid the manual regulation of α (the greedy rate) which is associated with the *exploration* phase of GRASP (GR algorithm), recently was proposed a variant for solving the Container Stacking Problem that controls its values reactively (da Silva, de Abreu, and Times 2019). In this approach, the greedy rate is estimated as follow. First of all, it is assigned to α a value within the interval $\{0, 1\}$, then, are defined an initial state (E: increment or decrement) and a modification value $0 \leq \Delta \leq 1$. In each iteration of the algorithm, the current solution s is evaluated in the construction phase. If s is better than the previous one, α is modified by the current state E by Δ. Otherwise, E is updated by the opposite state. In Algorithm 4 is presented the reactive GRASP framework for DIRP.

The local search is based on undoing movements from a stage of a solution by different movements. The substitution of a movement in a visited stage s', requires reconstructing all the subsequent stages. To generate a neighbour solution first is selected a stage in the constructive solution and then, it is used the GR algorithm to rebuild the solution.

Algorithm 4 Reactive GRASP for solving the DIPR

Require: $M = Population$, $N = Iterations$, $\alpha = GreedyRate$, E= increment or decrement, $\Delta = Change$.
Ensure: best found solution.
$Z^* \leftarrow \infty$
for $m = 1, ..., M$ **do**
  $\alpha \leftarrow$ GetGreedyRate (α, E, Δ)
  $x^s_{ibtp} \leftarrow$ ConstructionPhase (α)
  for $n = 1, ..., N$ **do**
    $x^s_{ibtp} \leftarrow$ LocalSearchPhase $\left(x^s_{ibtp}, \alpha\right)$
    $Z \leftarrow ReshuffleCount\left(x^s_{ibtp}\right)$
    if $Z < Z^*_3$ **then**
      $Z^* \leftarrow Z$
      $x^{s^*}_{ibtp} \leftarrow x^s_{ibtp}$
    end if
  end for
end for
return bestsolution $\leftarrow Z^*, x^{s^*}_{ibtp}$

4. Computational results: synthetic instances

In this section, computational results obtained when solving an instance benchmark comprised by synthetic instances of the DIRP are reported.

Implementation details The ILP models ILP1 and ILP2 were solved using CPLEX 12.6.3 (default setting), while the GR-based algorithms were implemented in JAVA. Both approaches were run on an Intel Core i7-5820K PC with a 3.30 GHz CPU and 32 GB RAM. A running time limit of 3600 seconds was set in all cases.

In the first part of the computational analysis, and mainly for comparison purposes, two types of synthetic instances were used, RANDOM and FIFO, which are described below.

Results on RANDOM instances In these type of instances, the storage stages A_{is} (pallet arrivals) are generated from a uniform distribution $U(1, S-1)$, while the retrieval stages D_{is} (pallet departures) are generated from a uniform distribution $U(A_{is}+1, S)$ and each arrival considers one pallet. Consider, $B \in \{1,2,3,4,5\}$, $T \in \{2,3,4\}$ and $P \in \{2,4,6\}$. Therefore, there is a total of 45 rack sizes, and for each of them 10 instances are generated for a total of $I = BTP$ pallets. This generation process follows the procedure presented in Chen, Li, and Gupta (2016). Note that the model supports any probability distribution for the generation of arrivals and departures. Note that decision-makers also can apply the model for several simultaneous pallet arrivals; in order to do so it is necessary to modify the corresponding model by replicating variables for each additional pallet.

Table 2 reports statistics on the number of reshuffles associated to the solutions attained by each of the considered approaches. In this table, each row is associated with a rack size. Hence, for each strategy the minimum, average and maximum objective function value, attained when solving the corresponding 10 random instances are reported. For instance, with a rack size given by 2 rows ($B = 2$), 2 tiers ($T = 2$) and a 6-pallet depth ($P = 6$), one has that $ILP1^{\min} = 4$, meaning that when comparing the best found solution of each of the 10 instances, the one requiring the least number of reshuffles requires 4 of them. Likewise, the one requiring the most number of reshuffles requires 18 (i.e. $ILP1^{\max} = 18$), and the average number of required reshuffles is 10 (i.e. $ILP1^{av} = 10$). A similar analysis can be performed for ILP2, GR and GRASP. Table 3, similar to Table 2, reports statistics (minimum, average and maximum values) of the quality of the solutions computed by the different approaches. For the case of the ILP1 and ILP2 approaches, the table shows statistics of the primal/dual gaps (measured as %) attained, when possible, by CPLEX within the time limit. The primal/dual gap corresponds to the relative difference between the best primal bound (i.e. induced by an integer and feasible solution) and best dual bound (i.e. induced by a linear relaxation solution) computed by the solver within the time limit. For the case of the GR approach and GRASP, the table reports the average value of the relative difference (measured as %) of the number of reshuffles attained with respect to the best known solution considering the four strategies. Finally, Table 4 reports, for different rack sizes, statistics (minimum, average and maximum values) of the running times (in seconds), associated to each of the proposed algorithmic approaches.

From the results reported in Tables 2, 3 and 4, it is possible to conclude that the ILP-based approaches (ILP1 and ILP2), fail to be computationally effective even for rather small instances. For instances, for $B = 2$, $T = 3$ and $P = 6$ both approaches are unable to compute solutions, with one hour, for any of the 10 problems associated to that rack size. On the contrary, the GR-based approaches are capable of finding solutions for *all* instances within very short running times (as can be seen from Table 4). More importantly, they are not only computationally efficient, but it is also capable of finding *very* good solutions as reported in Tables 2 and 3. From these tables, one can conclude that, when a comparison is possible, the solutions computed by the GR-based algorithms are not only equal than those found by the ILP approaches, but they can be much better (e.g. when $B = 2$, $T = 3$ and $P = 4$), and they are found in much shorter computing times. When the two heuristic approaches are compared, GRASP outperforms GR, obtaining smaller gaps in less running time.

One of the shortcomings of the analysis presented for RANDOM testbed, is that for many of these instances the optimal solution is trivial, i.e. no reshuffle is needed. Such shortcoming is tackled by introducing another set of synthetic instances, and the corresponding results are reported next.

Results on FIFO instances In these type of instance, the storage demand is equal to the Random instances blackand retrieval follows a FIFO policy, i.e. the i'th pallet stored is the ith pallet retrieved. The instances were generated considering $B \in \{1,2,3,4,5\}$, $T \in \{2,3,4\}$ and $P \in \{2,4,6\}$. Therefore, there is a total of 45 rack sizes, and for each for them, $I = BTP$ pallets are generated. In this way, 100% of the rack capacity is used and then emptied completely, which represents the worst case and, therefore, it is the hardest instance to solve.

Table 5 reports summarised results for FIFO instances. The columns associated to model ILP1 (resp. ILP2), report the attained primal/dual gap (column GAP (%)), and the CPU time (column CPU(s)). The columns corresponding to the heuristics, report the gap (min, av and max) respect to the best known solution, and the computing time (column CPU(s)).

Looking at the table, one can conclude that, as well as for RANDOM instances, the ILP2 approach allows to find solutions for a larger set of instances when compared to ILP1. However, in both cases, most of instances cannot be solved within the time limit. Likewise, and following the pattern presented for the RANDOM instance set, the GR algorithm is able to provide quite good solutions in rather short computing times, with the exception of 5 our of 45 cases in which the GR approach requires more than 200 seconds for finding the corresponding solution. Lastly, in this class of instances GRASP is also able to improve the performance of the GR algorithm. To assess, from a statistical point of view, the difference between

Table 2. Objective value (reshuffles) for ILP1, ILP2, GR and GRASP in RANDOM instances. (* means that CPLEX did not reach a primal bound or it presented a memory problem).

Rack Size			ILP1			ILP2			Greedy Randomised			Reactive GRASP		
B	T	P	min	av	max	min	av	max	min	av	max	min	av	max
1	2	2	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00	0
		4	0	1.90	4	0	1.90	4	0	1.90	4	0	1.90	4
		6	1	3.30	5	1	3.10	5	1	3.30	5	1	3.30	5
	3	2	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00	0
		4	0	0.60	5	0	0.60	5	0	0.60	5	0	0.60	5
		6	0	3.80	8	0	6.20	22	0	3.80	8	0	3.80	8
	4	2	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00	0
		4	0	1.10	3	0	1.10	3	0	1.10	3	0	1.10	3
		6	0	5.50	16	0	35.30	106	0	4.00	11	0	3.85	9
2	2	2	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00	0
		4	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00	0
		6	4	10.00	18	4	34.70	84	1	3.10	5	1	3.10	5
	3	2	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00	0
		4	0	0.60	4	0	4.10	36	0	0.20	1	0	0.20	1
		6	*	*	*	*	*	*	0	2.20	6	0	2.20	6
	4	2	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00	0
		4	*	*	*	7	60.00	122	0	0.50	4	0	0.50	4
		6	*	*	*	*	*	*	2	5.80	11	2	5.70	10
3	2	2	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00	0
		4	0	0.80	8	0	0.60	6	0	0.10	1	0	0.10	1
		6	*	*	*	*	*	*	0	2.20	6	0	2.20	6
	3	2	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00	0
		4	*	*	*	*	*	*	0	0.20	2	0	0.20	2
		6	*	*	*	*	*	*	0	3.10	11	0	3.10	11
	4	2	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00	0
		4	*	*	*	*	*	*	0	0.10	1	0	0.10	1
		6	*	*	*	*	*	*	0	6.00	13	0	5.85	12
4	2	2	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00	0
		4	*	*	*	0	3.50	26	0	0.00	0	0	0.00	0
		6	*	*	*	*	*	*	0	2.40	6	0	2.40	6
	3	2	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00	0
		4	*	*	*	*	*	*	0	0.00	0	0	0.00	0
		6	*	*	*	*	*	*	0	5.20	15	0	5.20	15
	4	2	*	*	*	0	0.00	0	0	0.00	0	0	0.00	0
		4	*	*	*	*	*	*	0	0.30	1	0	0.30	1
		6	*	*	*	*	*	*	4	8.60	18	4	8.30	16
5	2	2	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00	0
		4	*	*	*	*	*	*	0	0.00	0	0	0.00	0
		6	*	*	*	*	*	*	0	0.80	2	0	0.80	2
	3	2	0	0.00	0	0	0.00	0	0	0.00	0	0	0.00	0
		4	*	*	*	*	*	*	0	0.00	0	0	0.00	0
		6	*	*	*	*	*	*	0	5.60	14	0	5.50	13
	4	2	*	*	*	*	*	*	0	0.00	0	0	0.00	0
		4	*	*	*	*	*	*	0	0.70	2	0	0.70	2
		6	*	*	*	*	*	*	4	11.00	24	4	10.80	22

the mean performance of GRASP and GR, we use the non-parametric Kruskal-Wallis test to analyse the attained gaps. For Random instances, the test indicates strong evidence (p-value $= 0.0309$) against the null hypothesis (i.e. the two populations have equal mean). Likewise, for FIFO instances, the null hypothesis is rejected with weak evidence (p-value $= 0.0609$). We do not apply statistical tests to the gaps attained by the exact approaches as there are many missing entries.

Further discussion From the results attained when solving both, RANDOM and FIFO instances, it seems clear that longer dwell times result in much harder instances. This phenomenon can be explained by the fact that, the longer a unit load must remain in the rack, the more likely it is that it will require additional reshuffles. On the contrary, the instances with lower dwell times can be solved at the root node, when solved by the ILP1 and/or the ILP2 approaches.

In Figure 3, a summary of the solutions attained by the different algorithmic approaches on the considered instances is shown. Each plot shows the value of the objective function ('Reshuffles', in the y-axis) attained by the corresponding

Table 3. Gaps (%) attained when solving ILP1, ILP2, GR and GRASP in RANDOM instances (* means that CPLEX did not find a primal bound or presented a memory problems.

Rack Size			ILP1			ILP2			Greedy Randomised			Reactive GRASP		
B	T	P	min	av	max	min	av	max	min	av	max	min	av	max
1	2	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	3	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		6	0.00	0.00	0.00	0.00	49.26	100	0.00	0.00	0.00	0.00	0.00	0.00
	4	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		6	0.00	62.71	100	0.00	80.00	100	0.00	3.90	22.22	0.00	0.00	0.00
2	2	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		6	100	100	100	100	100	100	0.00	0.00	0.00	0.00	0.00	0.00
	3	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		4	0.00	20.00	100	0.00	20.00	100.0	0.00	0.00	0.00	0.00	0.00	0.00
		6	*	*	*	*	*	*	0.00	0.00	0.00	0.00	0.00	0.00
	4	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		4	*	*	*	*	*	100	0.00	0.00	0.00	0.00	0.00	0.00
		6	*	*	*	*	*	*	0.00	1.75	10.00	0.00	0.00	0.00
3	2	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		4	0.00	10.00	100	0.00	10.00	100	0.00	0.00	0.00	0.00	0.00	0.00
		6	*	*	*	*	*	*	0.00	0.00	0.00	0.00	0.00	0.00
	3	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		4	*	*	*	*	*	*	0.00	0.00	0.00	0.00	0.00	0.00
		6	*	*	*	*	*	*	0.00	0.00	0.00	0.00	0.00	0.00
	4	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		4	*	*	*	*	*	*	0.00	0.00	0.00	0.00	0.00	0.00
		6	*	*	*	*	*	*	0.00	2.56	8.33	0.00	0.00	0.00
4	2	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		4	*	*	*	0.00	30.00	100	0.00	0.00	0.00	0.00	0.00	0.00
		6	*	*	*	–	–	*	0.00	0.00	0.00	0.00	0.00	0.00
	3	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		4	*	*	*	*	*	*	0.00	0.00	0.00	0.00	0.00	0.00
		6	*	*	*	*	*	*	0.00	0.00	0.00	0.00	0.00	0.00
	4	2	*	*	*	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		4	*	*	*	*	*	*	0.00	0.00	0.00	0.00	0.00	0.00
		6	*	*	*	*	*	*	0.00	3.61	12.50	0.00	0.00	0.00
5	2	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		4	*	*	*	*	*	*	0.00	0.00	0.00	0.00	0.00	0.00
		6	*	*	*	*	*	*	0.00	0.00	0.00	0.00	0.00	0.00
	3	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		4	*	*	*	*	*	*	0.00	0.00	0.00	0.00	0.00	0.00
		6	*	*	*	*	*	*	0.00	1.82	7.69	0.00	0.00	0.00
	4	2	*	*	*	*	*	*	0.00	0.00	0.00	0.00	0.00	0.00
		4	*	*	*	*	*	*	0.00	0.00	0.00	0.00	0.00	0.00
		6	*	*	*	*	*	*	0.00	1.85	9.09	0.00	0.00	0.00

strategy for different settings of the generated instances. Instances (both FIFO and RANDOM) are characterised by the 'number of rows' (x-axis), the number of tiers (indicated by 'T'), and the rack depth (indicated by 'P').

From Figure 3 one can recognise that for the case of FIFO instances, the number of reshuffles increases when increasing the number of rows, the number of tiers and the depth of the rack. This turns out to be particularly clear when looking at the different curves associated to the GR approach or GRASP. On the contrary, for the RANDOM instances, since dwell times do not follow a clear pattern, there is no straightforward relation among the number the reshuffles and the dimensional characteristics of the racks; this is also clear when looking at the resulting curves associated to the GR-based methods.

Complementary, Figure 3 also shows how ILP1 and ILP2 approaches fail in providing feasible solutions, within the time limit, for several cases in both classes of instances. However, this situation is much more critical for the case of RANDOM

Table 4. Running times for ILP1, ILP2, GR and GRASP in RANDOM instances.

Rack Size			ILP1 (s)			ILP2 (s)			GR (s)			GRASP (s)		
B	T	P	min	av	max	min	av	max	min	av	max	min	av	max
1	2	2	0.08	0.28	0.55	0.08	0.27	0.56	0.00	0.01	0.01	0.00	0.00	0.00
		4	0.34	0.97	0.77	0.38	0.98	2.02	0.00	0.01	0.02	0.00	0.00	0.00
		6	0.41	1.40	4.53	0.72	6.12	23.59	0.00	0.05	0.06	0.00	0.04	0.05
	3	2	0.19	0.38	0.63	0.16	0.36	0.53	0.00	0.01	0.02	0.00	0.01	0.01
		4	0.53	5.03	6.91	1.08	4.77	12.00	0.00	0.07	0.09	0.00	0.05	0.07
		6	1.55	566	1063	9.31	2329	TL	0.00	3.51	4.51	0.00	1.71	2.12
	4	2	0.13	0.47	0.63	0.24	0.45	1.08	0.00	0.02	0.10	0.00	0.01	0.02
		4	1.16	58.05	93.74	2.64	378	1444	0.01	0.56	1.76	0.01	0.42	1.35
		6	81.00	2591	TL	107	3248	TL	0.13	8.05	16.71	0.10	6.85	9.02
2	2	2	0.13	0.41	0.59	0.11	0.41	0.66	0.00	0.01	0.01	0.00	0.01	0.01
		4	0.70	1.24	1.84	0.66	2.23	4.92	0.01	0.08	0.22	0.01	0.07	0.16
		6	TL	TL	TL	TL	TL	TL	0.10	3.75	6.00	0.08	2.05	3.21
	3	2	0.38	0.74	0.97	0.52	1.20	1.97	0.00	0.05	0.10	0.00	0.04	0.15
		4	3.95	790	TL	8.33	845	TL	0.00	1.84	4.71	0.00	1.12	3.56
		6	TL	TL	TL	TL	TL	TL	0.63	12.24	16.84	0.49	9.64	13.30
	4	2	1.16	1.91	3.06	1.61	4.01	6.41	0.01	0.05	0.17	0.01	0.04	0.05
		4	TL	TL	TL	TL	TL	TL	0.01	4.80	12.12	0.01	3.90	10.73
		6	TL	TL	TL	TL	TL	TL	7.35	48.20	64.27	6.95	35.20	50.71
3	2	2	0.53	0.72	0.83	0.53	0.72	1.44	0.01	0.02	0.04	0.01	0.01	0.02
		4	5.70	457	TL	8.55	394	TL	0.01	0.70	3.20	0.01	0.82	2.41
		6	TL	TL	TL	TL	TL	TL	0.10	16.36	26.91	0.12	11.06	18.91
	3	2	1.72	2.88	4.45	1.44	2.02	2.84	0.01	0.04	0.08	0.01	0.04	0.07
		4	TL	TL	TL	TL	TL	TL	0.00	2.32	8.28	0.00	2.23	7.42
		6	TL	TL	TL	TL	TL	TL	2.01	29.19	42.58	1.50	22.59	34.19
	4	2	10.16	181	656	2.44	8.40	33.83	0.01	0.06	0.27	0.01	0.07	0.29
		4	TL	TL	TL	TL	TL	TL	0.10	10.83	19.33	0.12	9.39	18.95
		6	TL	TL	TL	TL	TL	TL	6.18	76.86	132	5.98	42.25	70.25
4	2	2	0.78	1.36	1.88	0.61	1.26	1.88	0.01	0.03	0.05	0.00	0.02	0.04
		4	TL	TL	TL	89.11	2185	TL	0.01	0.76	1.70	0.01	0.53	2.34
		6	TL	TL	TL	TL	TL	TL	0.17	30.77	50.55	0.15	21.32	41.30
	3	2	2.23	62.55	273	1.25	5.05	10.80	0.01	0.03	0.09	0.01	0.04	0.07
		4	TL	TL	TL	TL	TL	TL	0.03	6.17	11.15	0.04	7.29	12.57
		6	TL	TL	TL	TL	TL	TL	6.52	58.66	102	6.33	43.12	89.61
	4	2	TL	TL	TL	4.53	33.31	125	0.01	0.07	0.38	0.01	0.04	0.16
		4	TL	TL	TL	TL	TL	TL	1.14	33.37	62.58	1.64	24.87	49.45
		6	TL	TL	TL	TL	TL	TL	8.78	96.75	190	5.78	73.92	137
5	2	2	1.55	3.57	6.89	1.38	2.16	3.41	0.01	0.03	0.07	0.00	0.04	0.05
		4	TL	TL	TL	TL	TL	TL	0.01	1.59	5.89	0.01	1.22	5.21
		6	TL	TL	TL	TL	TL	TL	0.22	47.90	92.61	0.20	35.62	67.48
	3	2	33.11	563	1474	3.47	8.40	16.81	0.00	0.05	0.22	0.00	0.03	0.17
		4	TL	TL	TL	TL	TL	TL	0.03	5.18	13.73	0.02	4.68	9.50
		6	TL	TL	TL	TL	TL	TL	0.63	55.86	115	0.72	42.56	61.87
	4	2	TL	TL	TL	TL	TL	TL	0.01	0.10	0.34	0.02	0.15	0.23
		4	TL	TL	TL	TL	TL	TL	13.12	51.78	82.08	8.82	48.28	58.25
		6	TL	TL	TL	TL	TL	TL	7.24	146	233	7.98	156	182

instances. As can be seen, ILP1 is capable of finding solutions for instances with more than 1 row only for a couple of instances with two tiers. Likewise, ILP2 also performs poorly when increasing the size of the racks.

When looking at the results reported un Tables 4 and 5, one can observe that two instances (i.e. two racks), with the same size (i.e. same number of pallets that can be stored), can lead to different algorithmic performances (specially for the proposed randomised algorithm). For instance, a rack of 24 pallets might be comprised by a structure of 1 row ($|B| = 1$), 4 tiers ($|T| = 4$) and a depth of 6 pallets ($|P| = 6$); however, the same number of pallets might be stored in a rack given by $|B| = 3$, $|T| = 2$, and $|P| = 4$; as can be seen in Table 5, while the first instance can be solved in 20.87 seconds, the second instances can be solved in 12.55 seconds. Therefore, although two configurations can be equivalent from the size point of view, if their structure is different, then the storage operations will be different as well. As consequence, different rack configurations will necessary yield different sets of feasible solutions, whose exploration will, therefore, require different computational efforts.

Table 5. Results of computational experiments in FIFO instances (* means that CPLEX did not find a primal bound or presented a memory problem.

Rack Size			ILP1		ILP2		Greedy Randomised				GRASP			
B	T	P	GAP(%)	CPU(s)	GAP(%)	CPU(s)	$\min_{\%}$	$\text{av}_{\%}$	$\max_{\%}$	av(s)	$\min_{\%}$	$\text{av}_{\%}$	$\max_{\%}$	av(s)
1	2	2	0.00	0.02	0.00	0.52	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
		4	0.00	0.42	0.00	1.81	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.05
		6	0.00	8.24	0.00	169	0.00	0.60	10.00	0.07	0.00	0.33	10.00	0.05
	3	2	0.00	0.19	0.00	0.31	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.01
		4	0.00	205	0.00	2113	10.00	10.00	10.00	0.09	10.00	10.00	10.00	0.50
		6	56.90	TL	98.31	TL	4.76	8.33	9.52	10.21	0.00	1.59	4.76	6.29
	4	2	0.00	2.00	0.00	2.48	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.03
		4	59.20	TL	92.61	TL	30.77	33.46	38.46	3.46	30.77	31.28	38.46	2.58
		6	*	TL	99.74	TL	0.00	4.13	13.33	20.87	0.00	0.44	6.67	9.56
2	2	2	0.00	3.44	0.00	2.30	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01
		4	82.20	TL	88.46	TL	0.00	0.00	0.00	2.82	0.00	0.00	0.00	2.87
		6	*	TL	99.75	TL	0.00	1.50	10.00	7.23	0.00	0.67	10.00	4.15
	3	2	56.07	TL	64.57	TL	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.29
		4	*	TL	99.83	TL	0.00	0.91	4.55	12.55	0.00	0.00	0.00	8.08
		6	*	TL	*	TL	9.76	15.07	17.07	19.45	0.00	4.72	9.76	14.92
	4	2	87.79	TL	93.30	TL	0.00	0.00	0.00	0.49	0.00	0.00	0.00	0.41
		4	*	TL	*	TL	0.00	3.59	8.82	51.06	0.00	0.39	5.88	33.29
		6	*	TL	*	TL	5.00	8.92	13.33	78.21	0.00	2.56	5.00	55.55
3	2	2	58.87	TL	66.96	TL	0.00	0.00	0.00	0.06	0.00	0.00	0.00	0.03
		4	*	TL	99.82	TL	0.00	0.00	0.00	15.29	0.00	0.00	0.00	10.86
		6	*	TL	*	TL	3.33	6.17	13.33	30.10	0.00	1.56	6.67	19.81
	3	2	88.86	TL	97.71	TL	0.00	0.00	0.00	0.90	0.00	0.00	0.00	0.80
		4	*	TL	*	TL	3.03	6.52	9.09	46.20	0.00	2.63	6.06	57.78
		6	*	TL	*	TL	9.38	13.83	18.75	47.88	0.00	5.68	12.50	37.42
	4	2	*	TL	99.80	TL	0.00	0.00	0.00	8.25	0.00	0.00	0.00	4.48
		4	*	TL	*	TL	1.92	7.37	11.54	72.24	0.00	0.58	5.77	53.26
		6	*	TL	*	TL	3.26	8.26	11.96	150	0.00	1.45	4.35	74.34
4	2	2	86.37	TL	87.49	TL	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.04
		4	*	TL	*	TL	0.00	0.00	0.00	52.67	0.00	0.00	0.00	64.54
		6	*	TL	*	TL	5.00	9.30	12.50	134	0.00	2.00	10.00	50.85
	3	2	*	TL	99.28	TL	0.00	0.00	0.00	1.43	0.00	0.00	0.00	1.64
		4	*	TL	*	TL	2.27	7.39	13.64	174	0.00	1.59	6.82	126
		6	*	TL	*	TL	4.44	10.17	13.33	170	0.00	2.93	6.67	124
	4	2	*	TL	99.97	TL	0.00	0.00	0.00	15.17	0.00	0.00	0.00	9.18
		4	*	TL	*	TL	2.90	8.91	13.04	167	0.00	1.79	7.25	112
		6	*	TL	*	TL	4.88	9.96	12.20	479	0.00	4.01	7.32	246
5	2	2	92.50	TL	90.91	TL	0.00	0.00	0.00	0.09	0.00	0.00	0.00	0.04
		4	*	TL	*	TL	0.00	0.00	0.00	93.35	0.00	0.00	0.00	97.49
		6	*	TL	*	TL	10.00	14.30	20.00	209	0.00	4.73	8.00	70.44
	3	2	*	TL	99.97	TL	0.00	0.00	0.00	8.20	0.00	0.00	0.00	2.20
		4	*	TL	*	TL	5.45	11.58	14.55	290	0.00	4.00	7.27	234
		6	*	TL	*	TL	4.50	12.52	15.32	332	0.00	4.23	9.91	173
	4	2	*	TL	*	TL	0.00	0.00	0.00	31.40	0.00	0.00	0.00	12.95
		4	*	TL	*	TL	2.27	9.26	13.64	313	0.00	1.70	6.82	239
		6	*	TL	*	TL	6.41	10.04	13.46	765	0.00	2.61	7.69	556

5. Case study

In this section, an application of the modelling and algorithmic framework proposed in the previous sections is reported. The application contexts correspond to Frunar Ltda., a storage and distribution company, located in Chile's central region. The storage capacity of the company is comprised by 14 freezer chambers, which are able to store, in total, 6000 pallets at -18°C. The company currently employs a so-called 'ABC' classification policy. This policy classifies products into three groups; A, B and C. It is preferable to devote the front layers of the rack to products of Category A, because of their rapid turnover; while, in the other extreme, the turnover of products of Category C is slower, so they are stored *deeper* into the rack.

The cold store chamber selected for the case study is 10 metres high, 18 metres long, 12 metres wide, and equipped with drive-in type compact storage shelves (see Figure 4). On one side of the aisle, the storage capacity is given by 12 rows, with

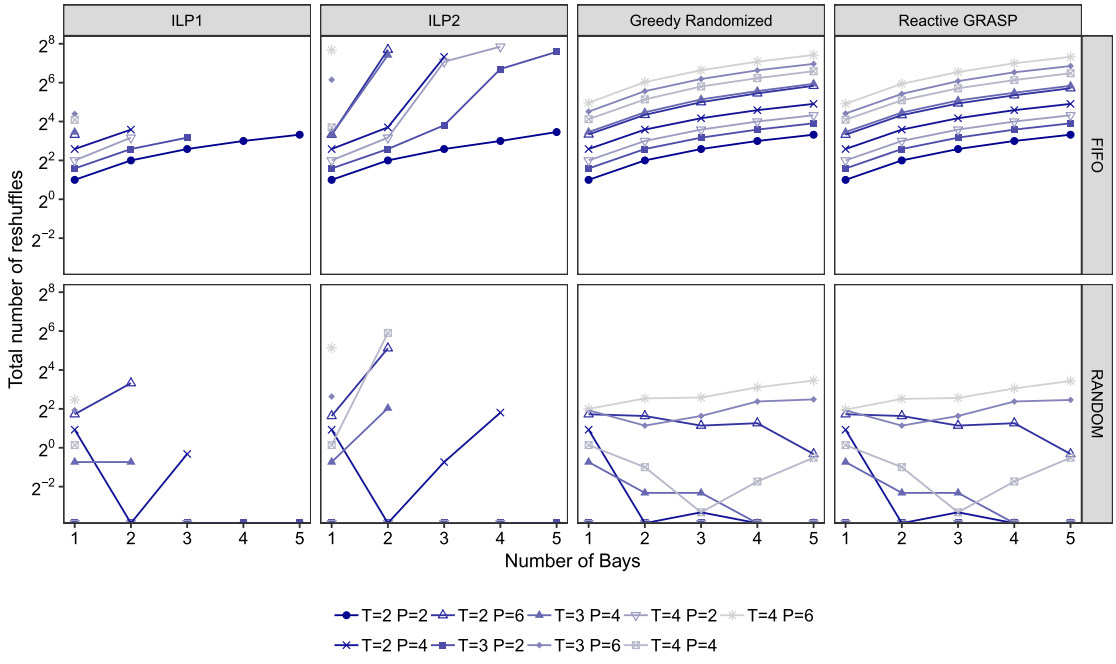

Figure 3. Comparison of the solutions attained by the different approaches on FIFO and RANDOM instances ('*T*' indicates the number of tiers, and '*P*' indicates the rack depth).

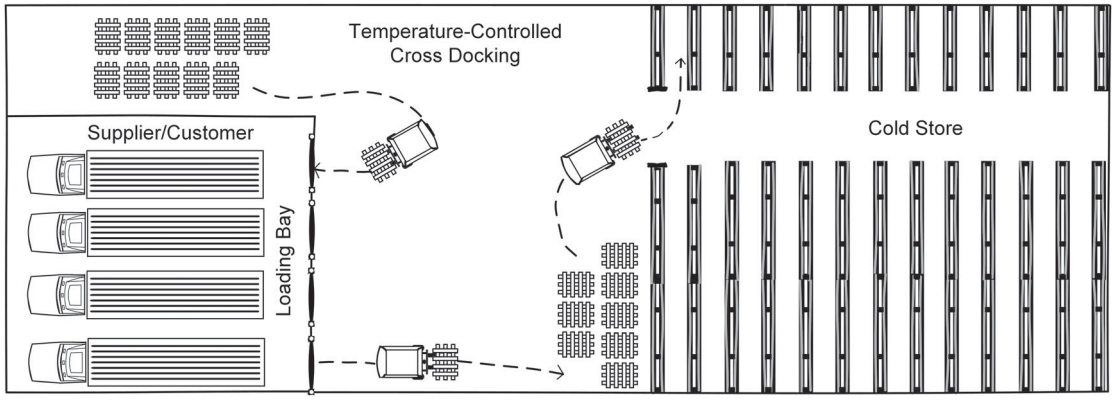

Figure 4. Layout of the case study cold storage chamber at Frunar Ltda.

four load tiers and six depth locations, while on the other side of the aisle has 12 rows, with four load tiers and two depth locations. Thus the store chamber has storage capacity for 384 pallets.

In real-life settings, such as the one of the considered company, decision-makers typically do not have total certainty, even in short periods such as weeks, regarding the future *workflow* towards and from the storage chambers, i.e. the volume of trucks' loading/unloading. In order to deal with such dynamics, decision-makers typically incorporate so-called rolling-horizon strategies (Bredström, Flisberg, and Rönnqvist 2013; Sahin, Narayanan, and Robinson 2013), which allow to dynamically update decisions as true data reveals over time. A rolling-horizon scheme is usually applied as depicted in Figure 5, and functions according to an iterative scheme as the one presented in Algorithm 5, which can be characterised as follows. A decision-making tool (e.g. one based on the proposed GR algorithm) is used to design the decisions that are to be taken for a horizon of, say, p days (see line 1). Nonetheless, it is assumed that such decisions must be updated after, say, f days ($f < p$), since it is likely the already designed decision does not perform correctly with the new available data (see line 2). Such period of f days is regarded as *frozen interval*, while the remaining $p - f$ days are regarded as *anticipation interval*. At the end of the fth day (denoted as f'), the input data is updated (see lines 5 and 6), and the decision tool is ran again for a horizon of p days (cycle within lines 4–12), assuming that after f days the process shall be repeated (see line 11), and the decisions must be rescheduled. Such scheme is iteratively applied along the whole planning horizon.

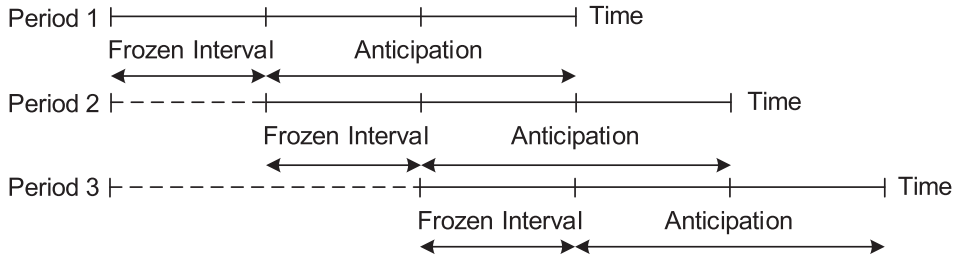

Figure 5. Rolling-horizon scheme.

For the case of the problem studied in this paper, a rolling-horizon scheme will depend on iteratively solving instances of the DIPR problem. Thereafter, and at the light of the results reported in the previous section, it is clear that a real-size instance of the DIPR, as the one of Frunar Ltda., can be approached effectively only by the GRASP algorithm. All these elements are combined into the aforementioned Algorithm 5, which, as explained above, outlines the functioning of the rolling-horizon scheme for solving DIPR problem, with periodical updates of the input data, using the GRASP algorithm.

Algorithm 5 Rolling-horizon based-algorithm

```
p ← Initialise length of the planning horizon
f ← Initialise length of the frozen interval
f′ ← Correspond to the f-th frozen interval (day) of planning
for f′ : 1 . . . p do
    Update new available data
    Update anticipation/forecasting for the next p − f′ days
    for Storage assignment and order-picking do
        Run GRASP Algorithm
        Update inventory
    end for
    p=p + f
end for
```

The data gathered from this real-world instance considers initial inventory, and the receipt and dispatch of pallets available during the period of 16 January 2017 to 21 January 2017. The chosen planning horizon p was established to be a working week because the information becomes increasingly imprecise as larger horizons are planned. The frozen interval f was set to one day. During this period, 153 pallets were stored, 110 pallets were dispatched, and the initial inventory on January 16 was 216 pallets.

The rolling-horizon algorithm returns a total of 15 reshuffles, which in comparison to the 36 reshuffles undertaken by the company is a 58% decrease. However, there is a large number of pallets that are not retrieved during this planning horizon, so the number of reshuffles should naturally increase over longer trial periods.

The significant improvement is explained because the data comes from a cold storage chamber, which is used to store a large variety of fruits belonging to different customers. In these situations, the company uses dedicated storage, storing only one type of product in a row, while the algorithm assigns locations based on the shared storage policy. Please note that the company's decision to use dedicated storage does not restrict the problem as it is entirely permissible to store different varieties of fruit in the same row. However, different foodstuffs, such as fruits and meats, should not be stored in the same cold store due to the risk of cross-contamination. The results obtained contribute not only to minimising total retrieval time but also to improving space utilisation.

6. Conclusions and future work

Compact storage systems have become increasingly popular in distribution centres. They require little space, provide flexibility in managing demand, and contribute to minimising direct variable costs.

In spite of their ever increasing use, manually operated compact systems have barely been studied. One important problem that has been identified in this type of systems is the high number of reshuffles stemming from the order retrieval

process due to the inefficient allocation of storage locations, which increases the complexity of the process and creates bottlenecks that increase total retrieval time. In practice, dedicated storage policies are used to avoid these reshuffles. This results in poor space utilisation due to partially filled storage rows and unprofitable warehouses.

In this study, a framework for optimising compact drive-in storage systems, that use a duration-of-stay based shared storage policy, has been proposed. The objective is to minimise the total number of reshuffles across the planning horizon, which implies improved space utilisation and a reduction of direct variable costs. Such goal is attained by proposing optimisation models and test them in synthetic instances using different arrival and departure unit load distributions (including worst cases). However, the results show that it is very difficult to achieve an optimal solution to the problem posed using a traditional optimisation approach due to the high computational complexity. Therefore, its application is only recommended for small instances and to assess the quality of heuristic solutions.

In order to deal with such shortcoming, a GRASP metaheuristic to find good solutions in reasonable computation time is proposed. The algorithm was tested using a rolling-horizon planning strategy in a frozen food distribution centre located in the Maule Region of Chile. Results obtained in a limited case study show that the number of reshuffles decreases by more than 50% over a one-week planning horizon. The improvement can be explained by the fact that the algorithm assigns locations based on a shared storage policy whereas the company uses a dedicated storage policy.

Future research could focus on three important issues. The first issue concerns studying how alternative algorithmic strategies could complement and, eventually, contrast with respect to the presented results. In this sense, exact approaches such as decomposition and cutting plane algorithms, or metaheuristic approaches such as genetic algorithms or tabu search (both relying on more advanced randomised schemes), could help to overcome the computational difficulties induced by larger instances. The second is to investigate how uncertain unit load demand patterns could be incorporated as part of the modelling and algorithmic strategies (see, e.g. Ang, Lim, and Sim 2012), to enhance the practical use of the proposed tool. Finally, a third issue that it is worth studying is how to adapt our approach in order to recognise heterogeneous loads and heterogeneous racks as those of e-commerce which are characterised by a very large number of small sized orders and returns (see, e.g. the recently published article by Bahrami, Aghezzaf, and Limère (2019), where a new storage assignment strategy is proposed for this type of warehouse setting).

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

D. Revillot and F. Pérez-Galarce acknowledge the support of the Centro de Estudios en Alimentos Procesados CEAP, Chile; Programa Regional de CONICYT. E. Álvarez-Miranda acknowledges the support of the National Commission for Scientific and Technological Research CONICYT, Chile, through the grant FONDECYT N.1180670, and through the Complex Engineering Systems Institute PIA/BASAL AFB180003; Programa Investigación Asociativa CONICYT.

References

Alfieri, A., M. Cantamessa, A. Monchiero, and F. Montagna. 2012. "Heuristics for Puzzle-based Storage Systems Driven by a Limited Set of Automated Guided Vehicles." *Journal of Intelligent Manufacturing* 23 (5): 1695–1705.

Altarazi, S., and M. Ammouri. 2018. "Concurrent Manual-order-picking Warehouse Design: A Simulation-based Design of Experiments Approach." *International Journal of Production Research* 56 (23): 7103–7121.

Ang, M., Y. Lim, and M. Sim. 2012. "Robust Storage Assignment in Unit-load Warehouses." *Management Science* 58 (11): 2114–2130.

Avriel, M., M. Penn, and N. Shpirer. 2000. "Container Ship Stowage Problem: Complexity and Connection to the Coloring of Circle Graphs." *Discrete Applied Mathematics* 103 (1): 271–279.

Azadeh, K., M. de Koster, and D. Roy. 2017. "Robotized Warehouse Systems: Developments and Research Opportunities." ERIM Report Series Research in Management Erasmus Research Institute of Management, ERS-2017-009-LIS. http://hdl.handle.net/1765/99983.

Bahrami, B., E. Aghezzaf, and V. Limère. 2019. "Enhancing the Order Picking Process Through a New Storage Assignment Strategy in Forward-reserve Area." *International Journal of Production Research* 57 (21): 6593–6614.

Bessenouci, H., Z. Sari, and L. Ghomri. 2012. "Metaheuristic Based Control of a Flow Rack Automated Storage Retrieval System." *Journal of Intelligent Manufacturing* 23 (4): 1157–1166.

Bortfeldt, A., and F. Forster. 2012. "A Tree Search Procedure for the Container Pre-marshalling Problem." *European Journal of Operational Research* 217 (3): 531–540.

Bortolini, M., R. Accorsi, M. Gamberi, R. Manzini, and A. Regattieri. 2015. "Optimal Design of AS/RS Storage Systems with Three-class-based Assignment Strategy Under Single and Dual Command Operations." *The International Journal of Advanced Manufacturing Technology* 79 (9–12): 1747–1759.

Boysen, N., D. Briskorn, and S. Emde. 2017. "Sequencing of Picking Orders in Mobile Rack Warehouses." *European Journal of Operational Research* 259 (1): 293–307.

Bozer, Y., and J. White. 1984. "Travel-time Models for Automated Storage/Retrieval Systems." *IIE Transactions* 16 (4): 329–338.

Bredström, D., P. Flisberg, and M. Rönnqvist. 2013. "A New Method for Robustness in Rolling Horizon Planning." *International Journal of Production Economics* 143 (1): 41–52.

Calzavara, M., C. Glock, E. Grosse, and F. Sgarbossa. 2019. "An Integrated Storage Assignment Method for Manual Order Picking Warehouses Considering Cost, Workload and Posture." *International Journal of Production Research* 57 (8): 2392–2408.

Cardin, O., P. Castagna, Z. Sari, and N. Meghelli. 2012. "Performance Evaluation of in-deep Class Storage for Flow-rack AS/RS." *International Journal of Production Research* 50 (23): 6775–6791.

Caserta, M., S. Schwarze, and S. Voß. 2012. "A Mathematical Formulation and Complexity Considerations for the Blocks Relocation Problem." *European Journal of Operational Research* 219 (1): 96–104.

Chen, Z., X. Li, and J. Gupta. 2016. "Sequencing the Storages and Retrievals for Flow-rack Automated Storage and Retrieval Systems with Duration-of-stay Storage Policy." *International Journal of Production Research* 54 (4): 984–998.

Cormier, G., and E. Gunn. 1992. "A Review of Warehouse Models." *European Journal of Operational Research* 58 (1): 3–13.

da Silva, A., R. de Abreu, and V. Times. 2019. "A Reactive GRASP Metaheuristic for the Container Retrieval Problem to Reduce Crane's Working Time." *Journal of Heuristics* 25 (2): 141–173.

de Koster, R., A. Johnson, and D. Roy. 2017. "Warehouse Design and Management." *International Journal of Production Research* 55 (21): 6327–6330.

de Koster, R., T. Le-Duc, and K. Roodbergen. 2007. "Design and Control of Warehouse Order Picking: A Literature Review." *European Journal of Operational Research* 182 (2): 481–501.

Gagliardi, J., J. Renaud, and A. Ruiz. 2015. "Sequencing Approaches for Multiple-aisle Automated Storage and Retrieval Systems." *International Journal of Production Research* 53 (19): 5873–5883.

Gharehgozli, A., Y. Yu, R. de Koster, and J. Udding. 2014. "A Decision-tree Stacking Heuristic Minimising the Expected Number of Reshuffles At a Container Terminal." *International Journal of Production Research* 52 (9): 2592–2611.

Gray, A., U. Karmarkar, and A. Seidmann. 1992. "Design and Operation of An Order-consolidation Warehouse: Models and Application." *European Journal of Operational Research* 58 (1): 14–36.

Grosse, E., C. Glock, and W. Neumann. 2017. "Human Factors in Order Picking: A Content Analysis of the Literature." *International Journal of Production Research* 55 (5): 1260–1276.

Gu, J., M. Goetschalckx, and L. McGinnis. 2007. "Research on Warehouse Operation: A Comprehensive Review." *European Journal of Operational Research* 177 (1): 1–21.

Gue, K., K. Furmans, Z. Seibold, and O. Uludağ. 2014. "Gridstore: A Puzzle-based Storage System with Decentralized Control." *IEEE Transactions on Automation Science and Engineering* 11 (2): 429–438.

Hale, T., M. Hanna, F. Huq, and A. Gil. 2015. "Closed Form Models for Dwell Point Locations in a Multi-aisle Automated Storage and Retrieval System." *International Journal of Industrial and Systems Engineering* 19 (3): 364–388.

Handfield, R., F. Straube, H. Pfohl, and A. Wieland. 2013. *Trends and Strategies in Logistics and Supply Chain Management*. Hamburg: DVV Media Group GmbH.

Hausman, W., L. Schwarz, and S. Graves. 1976. "Optimal Storage Assignment in Automatic Warehousing Systems." *Management Science* 22 (6): 629–638.

Hu, Y., S. Huang, C. Chen, W. Hsu, A. Toh, C. Loh, and T. Song. 2005. "Travel Time Analysis of a New Automated Storage and Retrieval System." *Computers & Operations Research* 32 (6): 1515–1544.

Lee, Y., and S. Chao. 2009. "A Neighborhood Search Heuristic for Pre-marshalling Export Containers." *European Journal of Operational Research* 196 (2): 468–475.

Lee, Y., and N. Hsu. 2007. "An Optimization Model for the Container Pre-marshalling Problem." *Computers & Operations Research* 34 (11): 3295–3313.

Lerher, T. 2016. "Travel Time Model for Double-deep Shuttle-based Storage and Retrieval Systems." *International Journal of Production Research* 54 (9): 2519–2540.

Lerher, T. 2018. "Aisle Changing Shuttle Carriers in Autonomous Vehicle Storage and Retrieval Systems." *International Journal of Production Research* 56 (11): 3859–3879.

Lerher, T., M. Sraml, I. Potrc, and T. Tollazzi. 2010. "Travel Time Models for Double-deep Automated Storage and Retrieval Systems." *International Journal of Production Research* 48 (11): 3151–3172.

Manzini, R., Y. Bozer, and S. Heragu. 2015. "Decision Models for the Design, Optimization and Management of Warehousing and Material Handling Systems." *International Journal of Production Economics* 170 :711–716.

Mirzaei, M., R. de Koster, and N. Zaerpour. 2017. "Modelling Load Retrievals in Puzzle-based Storage Systems." *International Journal of Production Research* 55 (21): 6423–6435.

Park, T., R. Choe, Y. Kim, and K. Ryu. 2011. "Dynamic Adjustment of Container Stacking Policy in An Automated Container Terminal." *International Journal of Production Economics* 133 (1): 385–392.

Pazour, J., and H. Carlo. 2015. "Warehouse Reshuffling: Insights and Optimization." *Transportation Research Part E: Logistics and Transportation Review* 73: 207–226.

Resende, M., and C. Ribeiro. 2014. "GRASP: Greedy Randomized Adaptive Search Procedures." In *Search Methodologies*, 287–312. New York: Springer.

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

Roodbergen, K., I. Vis, and G. Taylor. 2015. "Simultaneous Determination of Warehouse Layout and Control Policies." *International Journal of Production Research* 53 (11): 3306–3326.

Sahin, F., A. Narayanan, and E. Robinson. 2013. "Rolling Horizon Planning in Supply Chains: Review, Implications and Directions for Future Research." *International Journal of Production Research* 51 (18): 5413–5436.

Stadtler, H. 1996. "An Operational Planning Concept for Deep Lane Storage Systems." *Production and Operations Management* 5 (3): 266–282.

Staudt, F., G. Alpan, M. Di Mascolo, and C. Rodríguez. 2015. "Warehouse Performance Measurement: A Literature Review." *International Journal of Production Research* 53 (18): 5524–5544.

Tompkins, J., J. White, Y. Bozer, and J. Tanchoco. 2010. *Facilities Planning*. John Wiley & Sons.

van den Berg, J., and W. Zijm. 1999. "Models for Warehouse Management: Classification and Examples." *International Journal of Production Economics* 59 (1): 519–528.

Wan, Y., J. Liu, and P. Tsai. 2009. "The Assignment of Storage Locations to Containers for a Container Stack." *Naval Research Logistics* 56 (8): 699–713.

Wang, N., B. Jin, and A. Lim. 2015. "Target-guided Algorithms for the Container Pre-marshalling Problem." *Omega* 53: 67–77.

Xu, X., G. Shen, Y. Yu, and W. Huang. 2015. "Travel Time Analysis for the Double-deep Dual-shuttle AS/RS." *International Journal of Production Research* 53 (3): 757–773.

Xu, X., X. Zhao, B. Zou, Y. Gong, and H. Wang. 2019. "Travel Time Models for a Three-dimensional Compact AS/RS Considering Different I/O Point Policies." *International Journal of Production Research*, in press.

Yakovleva, N., J. Sarkis, and T. Sloan. 2012. "Sustainable Benchmarking of Supply Chains: The Case of the Food Industry." *International Journal of Production Research* 50 (5): 1297–1317.

Yang, J., and K. Kim. 2006. "A Grouped Storage Method for Minimizing Relocations in Block Stacking Systems." *Journal of Intelligent Manufacturing* 17 (4): 453–463.

Yang, P., L. Miao, Z. Xue, and B. Ye. 2015. "Variable Neighborhood Search Heuristic for Storage Location Assignment and Storage/retrieval Scheduling Under Shared Storage in Multi-shuttle Automated Storage/retrieval Systems." *Transportation Research Part E: Logistics and Transportation Review* 79: 164–177.

Yang, P., Y. Peng, B. Ye, and L. Miao. 2017. "Integrated Optimization of Location Assignment and Sequencing in Multi-shuttle Automated Storage and Retrieval Systems Under Modified 2 N-command Cycle Pattern." *Engineering Optimization* 49 (9): 1604–1620.

Yu, Y., and R. de Koster. 2009. "Optimal Zone Boundaries for Two-class-based Compact Three-dimensional Automated Storage and Retrieval Systems." *IIE Transactions* 41 (3): 194–208.

Yu, Y., and R. de Koster. 2012. "Sequencing Heuristics for Storing and Retrieving Unit Loads in 3D Compact Automated Warehousing Systems." *IIE Transactions* 44 (2): 69–87.

Yu, Y., and R. de Koster. 2013. "On the Suboptimality of Full Turnover-based Storage." *International Journal of Production Research* 51 (6): 1635–1647.

Yu, H., and Y. Yu. 2019. "Optimising Two Dwell Point Policies for AS/RSs with Input and Output Point At Opposite Ends of the Aisle." *International Journal of Production Research* 57 (21): 6615–6633.

Yu, Y., H. Yu, R. de Koster, and N. Zaerpour. 2017. "Optimal Algorithm for Minimizing Retrieval Time in Puzzle Based Storage System with Multiple Simultaneously Movable Empty Cells." Working Paper Erasmus University, Rotterdam.

Zaerpour, N., R. de Koster, and Y. Yu. 2013. "Storage Policies and Optimal Shape of a Storage System." *International Journal of Production Research* 51 (23–24): 6891–6899.

Zaerpour, N., Y. Yu, and R. de Koster. 2015. "Storing Fresh Produce for Fast Retrieval in An Automated Compact Cross-dock System." *Production and Operations Management* 24 (8): 1266–1284.

Zou, B., M. de Koster, and X. Xu. 2016. "Evaluating Dedicated and Shared Storage Policies in Robot-based Compact Storage and Retrieval Systems." ERIM Report Series Research in Management Erasmus Research Institute of Management. URL https://ssrn.com/abstract = 2888649.

Zou, B., X. Xu, Y. Gong, and R. de Koster. 2016. "Modeling Parallel Movement of Lifts and Vehicles in Tier-captive Vehicle-based Warehousing Systems." *European Journal of Operational Research* 254 (1): 51–67.