A support-design tool for block-stacking storage system layout

Accorsi R.*, Baruffaldi G., Galizia F.G.**, Manzini R.*, Maranesi F.***

* *Department of Industrial Engineering, University of Bologna – Alma Mater Studiorum, Viale Risorgimento 2, 40136, Bologna, Italy (riccardo.accorsi2@unibo.it)*

** *Department of Management and Engineering, University of Padua, Stradella San Nicola, 3 36100, Vicenza, Italy*

Abstract: In warehousing system, block stacking is one of the most common storage mode to guarantee high storage density in case of large quantities and limited product variety. Most common applications for block stacking storage are end-of-line warehouses at manufacturing facilities in processing industry (e.g., beverage, bakery, tissue industries). These storage systems put-away the incoming pallets into deep lanes of homogeneous products and take advantage of vertical space through high stacks. Although the design of the optimal lane depth is widely debated in the extant literature through analytical modelling, the lack of support-design methodologies able to tackle the proper design of high-turnover warehouses that experience high variability on inbound and outbound flows is even stated. Therefore, the aim of this paper is to illustrate a decision-support tool that implements and incorporates some existing and original models and uses a simulation approach to aid the design of a block-stacking storage system layout.

Design/Methodology/Approach

The proposed tool identifies the proper layout configuration of a block-stacking storage system in accordance with the inventory turnover and the inbound and outbound flows. The input dataset includes the inbound/outbound flows and the historical inventory profile and fuels a set of analytical models and an iterative design-procedure which have been incorporated into a user friendly software application. This tool support the design of the storage layout and quantifies a set of associated KPIs accounting for the space utilization.

Findings

A case study illustrating the novelty and the effectiveness of this tools is illustrated and the results discussed in a what-if analysis. The obtained results and the quantified metrics in terms of space efficiency demonstrate the effectiveness of the resulting storage layout in comparison with the layout benchmark.

Keywords: block-stacking storage, layout design, product flow manufacturing, storage lane

1.Introduction

Block storage systems are widely diffuse in the flow/line manufacturing and processing systems (Van den Berg and Zijm, 1999) that typically deal with commodities and other products characterized by large inventory and high turnover. Many firms, belonging to different sectors as beverage, food, tissue, require such a storage mode, because it is flexible and easy-to-manage. Because of its scalability, it is suitable for being modelled through mathematics for both layout design and operations management (Gu et al., 2010).

The storage lane is an homogeneous queue of unit-load, holding a generic Stock-Keeping Unit (SKU), facing the aisle by one or both sides. The aisle provides accessibility to each lane, but this empty space is not profit-generating for the warehouse. In such a storage mode, the pallets locations are shared among the inventory mix and in case of high turnover, the infrastructural can be amortized. The available number of lanes of a generic depth is indeed the main driver for the setting of a block storage layout. As a consequence, the lane depth is anciently debated by operations researchers (Marsh, 1979, Matson and White, 1981, Goetschalckx and Ratliff, 1991).

Deep lanes fit with large production lots of a generic SKU, but the empty locations resulting from the pallet retrieving remain empty but not available until the last pallet of that lane is retrieved. This space loss is called honey combing. During the inventory cycle, the space losses decrease proportionally with the increasing demand rate. Short lanes are released earlier and the unoccupied space is less than with deep lanes. Conversely, large lots of a SKU requires more lanes, and this results in longer aisles which waste space for accessibility (Bartholdi and Hackman, 2013, Goetschalckx, 2003).

The definition of the storage layout entails a wide set of issues, recognized under the goals of increasing space and time efficiency (Kind, 1975). Matson et al. (2014) explore how random or dedicated storage patterns affect the determination of the optimal lane depth for a given SKU. Sonnentag et al. (2014) calculate the optimal lane depth that minimises the costs associated to put-away and retrieving operations. In the last decade, the development of this field of research moved toward the determination of the proper lane-depth that addresses the so-called block relocation problem. Kim and Hong (2006), Meneghetti (2009), White et al. (2013) tackled these aspects in the design of storage systems, while Yang and Kim (2006) and Jang et al. (2013) studied the layout of port container terminals through formulations of product-yards allocation problem.

This paper deals with the design of storage layout configuration that maximizes the space utilization in response to the incoming lots from the manufacturing lines, in presence of high inventory turnover and seasonal demand.

The design of block storage systems involves many different storage modes as floor storage, drive-in, drive-through rack, pallet flow-rack and automated storage and retrieving vehicle systems (ASRVS). The latter consists on an automated drive-in rack, adopting shuttles that travel within the rack instead of forklifts. These systems, when properly shaped (Manzini et al., 2016), optimize the space efficiency and makes the storage locations more selective.

We propose an original iterative top-down procedure for the design and management of block storage systems. This procedure builds upon the analytical models developed by the extant literature and aids the design of the storage layout configured as a set and number of storage lanes per each depth.

The outline of the paper is organized as follows. Section 3 deeply illustrates a support-decision tool aiding the design and management of block-stacking warehousing systems. The tool implements an iterative procedure, based on a set of analytical models and algorithms, to address the identification of the proper lane depths that comply with given space efficiency requirements. A database architecture and a set of graphic user interfaces (GUIs) are also illustrated to show the potential functionalities enabling the application of the procedure with real world storage systems. Section 4 illustrates the outcomes from a real world testbed tackled through the proposed procedure and tool.

2. Support-design procedure

We introduce an iterative procedure for the determination of the storage layout which corresponds to the proper configuration of lane depth and the number of lanes that meet given production and demand profiles.

2.1 Lane depth

The first step of the iterative procedure deals with the determination of the optimal lane depth per each SKU to respond to the production and demand profiles of a given time horizon. This step builds on an extant analytical model that calculates the lane depth of a SKU that minimizes the overall unoccupied space due to honey combing and accessibility. The model is defined as follows (Bartholdi and Hackman, 2013).

Let the lane be k pallet position deep, the space in front of the lane be of area a (measured in equivalent pallet positions), while the total area charged to one lane be $k + a/2$. The generic SKU i experiences constant demand of D_i pallets during an inventory cycle (e.g., 1 month) with a reorder quantity of q_i pallets (and an order cycle with a duration of q_i/D_i). Lastly, let the stackable column per SKU i be z_i. The parameter z_i measures the so-called stackability, as the number of levels a pallet of SKU i can be stacked by maintain the storage safety. Thus, the optimal lane depth k for the incoming lot q_i of the SKU i, depends on the adopted storage mode, and is defined in Table 1.

Storage mode	Lane depth	Rack level
Floor storage	$k_i = \sqrt{\left(\frac{a}{2}\right)\left(\frac{q_i}{z_i}\right)}$	Depends on stackability
Drive-in rack	$k_i = \sqrt{\left(\frac{a}{2}\right)\left(\frac{q_i}{z}\right)}$	Equal per all SKUs
Drive-Through rack	$k_i = \sqrt{a\left(\frac{q_i}{z}\right)}$	Equal per all SKUs
Flow-rack	$k_i = \sqrt{a \cdot q_i}$	1
ASRVS	$k_i = \sqrt{\left(\frac{a}{2}\right)q_i}$	1

Table 1. Lane depth model

Assuming as constant the daily demand D_{it} of SKU i during the inventory cycle t, the optimal lane depth for the lot q_i is k_i.

At this step the designer can try different lane depths, storage modes, and storage levels and establishes through Equation (1) the number of lanes l_i^t required by each SKU i at each inventory cycle t. As a result, the daily optimal layout configuration $Layout^t$ cumulates the lanes allocated to each incoming lot, according to Equation (2).

$$l_i{}^t = \frac{q_i}{z_i \cdot k_i} \tag{1}$$

$$Layout^t = \sum_i^N l_i{}^t \tag{2}$$

In order to meet the production and demand profiles during an inventory cycle, Equation (1) set the number of lanes of depth k to allocate that SKU. The overall number of required lanes for the incoming lots during the inventory cycle t measures the requirements of pallet locations (i.e., the capacity) of the block stacking system.

2.2 Storage/Retrieving operations scheduling

The first step deals with the determination of the optimal lane depth per each lot in a given inventory cycle, and cumulates the required lanes to quantify the space to be allocated in the layout. The model assumes a constant demand rate during the inventory cycle *t*. In real-world, the inventory mix changes and the honey combing and accessibility costs effectively depends by the SKU's turnover. As example, the honey combing for slow moving SKUs tends to be higher than for fast-moving SKUs. The number of lanes l_i^t result in depicting the layout at the generic inventory cycle *t* according to the expected constant demand D_{it} without taking into account the real trend of demand.

In order to address these aspects, the procedure implements the *dynamic scheduling* phase, that involves the put-away and retrieving tasks experienced by a lane during the inventory cycle *t*. This time, also named *release horizon*, influences the requirements of lane to allocate the incoming production and related expected storage capacity of the warehouse. Assuming the release horizon as the average turnover of each lot of SKU *i*, the procedure copes with seasonality and the changing inventory mix to obtain a more accurate measure of the overall storage space to be guaranteed.

While the analytical models of Table 1 provides the optimal lane depth per each incoming lot of SKU *i* during inventory cycle *t*, the *dynamic scheduling* provides the rolling storage capacity to respond to the daily production and the demand profiles.

2.3 Simulation

The third step develops and uses a simulation approach to study the impact of put-away and retrieving operations on the space efficiency of the designed layout. These performances include the ratio of occupied storage locations to the total capacity, the ratio of occupied lanes to the overall number of lanes, and the saturation of each open lane. The historical inventory is used to simulate the storage and retrieving activities in a what-if multi-scenario analysis, comparing alternative storage layout configurations resulting by the previous steps. Whether the space efficiency metrics do not satisfy the designer expectation (i.e. low values of space saturation), the procedure iterates the *dynamic scheduling* and reshapes the layout accordingly.

The developed simulation adopts a greedy heuristics to simulate the first-in-first-out (FIFO) assignment of incoming lots to the available lanes.

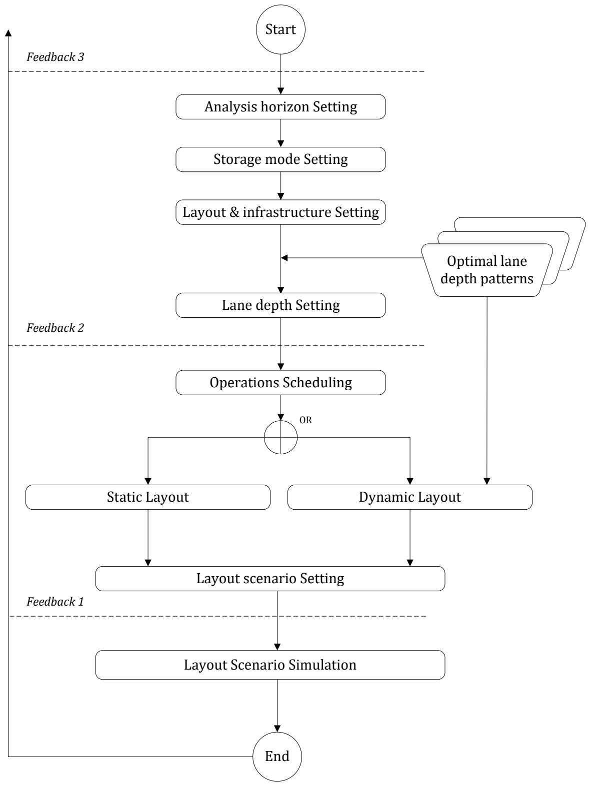

Figure 1. top-down procedure

Figure 1 represents the flow-chart of the procedure steps. The preliminary phase deals with data collection and is connected to gathering the available inventory, production and demand profiles from the company warehouse management system (WMS). These data are stored in a tailored database architecture. The first step deals with the settings of the time horizon, the set of available storage modes (i.e. floor storage, drive-in, drive-through, flow-rack, ASRVS), the set of available lane depths, as inputs for the aforementioned analytical model (see Section 2.1).

The second step enables extending the *static* model by considering the so-called *release horizon* in-between put-away and retrieving tasks. According to the *release horizon* the daily requirement of lanes is calculated in a rolling approach and the $Layout^t$ at the generic period *t* calculated period by period.

Two options have been developed and are therefore available for the designer. The so-called *static layout* approach inherits the optimal lane depth assigned to the average lot of each SKU *i*, and holds these lanes for the average SKU turnover. The so-called *dynamic layout* approach quantifies the optimal number of lanes per each lot *qi* of SKU *i* and holds these lanes for the average SKU turnover. The output of both approaches consists on a set of layout configurations (i.e., layout *t)*, rolling day by day, characterized by different value of storage capacity.

The third step simulates a set of alternative layout configurations resulting by the dynamic scheduling module, and enables assessing the space efficiency performances during a given time horizon.

3. A tool

The procedure is applied to a dataset that draws the historical behaviour of the block-stacking system in terms of production and demand profiles. To this purpose an E-R diagram has been developed as follows. The proposed E-R diagram is built on four tables, named SKU, INVENTORY, INBOUND and OUTBOUND. These tables are quickly gatherable from a company WMS. In the following, a summary of each table is given, with the detail of the main fields.

• *SKU*. This table represents the SKU master file that contains the code, carton volume, carton weight, description, demand class of each SKU. It typically counts tens thousands of rows. The class of demand reports the classification of the SKUs according to their turn over (i.e. A, B, C classes of Pareto curve). The procedure uses these classes to classify the SKUs to devote to each storage areas (or storage modes) and clusterizes the layout accordingly.

• *OUTBOUND*. This table represents the demand profile of a specific time horizon and usually counts millions of lines. Each row is composed by the due date of order, the order code, the SKU code and the picked quantity in terms pallets. The field BatchCode is of interest in the determination of the release horizon of each lot.

• *INBOUND*. This table reports the historical inbound profile composed by the incoming lots received by manufacturing lines or docks. Each record is made by the receiving date, the batch code and the SKU code, as well as the pallet quantities to be stored in the system. This table is key for the implementation of the static models for the setting of the optimal lane depth for each SKU.

• *INVENTORY*. This table reports the inventory master file for the overall SKUs population. The historical stocks of the SKUs enable to assess the space efficiency and saturation performances of the designed unit-load warehousing system in the third step of the procedure.

• *LAYOUT*. This is a snapshot of the storage configuration resulting by the second step of the procedure. Through this table the designer may also import an *as-is* configuration, bypassing the first two steps, in order to measure its space efficiency via simulation as a benchmark for further improvements.

• *UL*. This table indicates the type and size of unit load produced by the manufacturing lines. This table aims to define the aisle width in term of equivalent pallet locations.

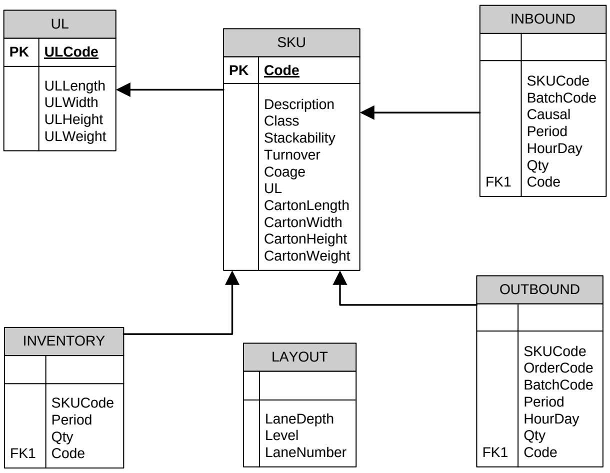

Figure 2. E-R diagram

The E-R diagram is illustrated in Figure 2. The proposed procedure is implemented through a decision-support tool developed in C# .NET language. The adopted DB provider is MS Access. The management and control of the tool is allowed to the designer, through a set of developed GUIs, which support the his interaction with the proposed iterative procedure.

3.1 Lane setting

A first module (see Figure 3), named lane setting, supports the determination of optimal lane depths for the SKUs given a time horizon. This horizon, defined by the calendar panels, filters the considered rows of the table INBOUND. The average quantity *qi* of the incoming lots for a generic SKU *i*, is used to calculate the optimal lane depth per each SKU according to Equation (1).

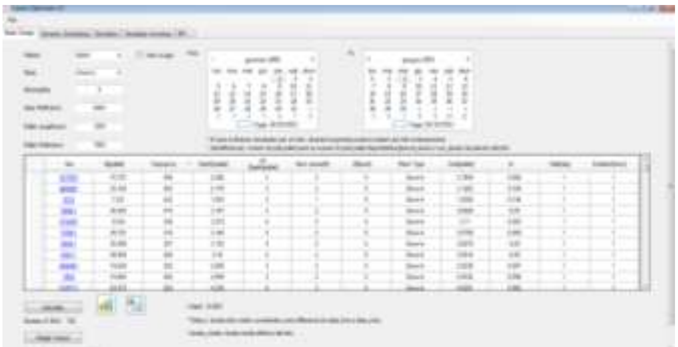

Figure 3. Lane setting input GUI

The output by this step consists on the analysis of the lane depth frequency of the block-stacking layout. It counts per each depth, the total number of lanes required to allocate the average incoming lot of the SKUs. It provides (Figure 4) a *static* picture of the average storage configuration layout, composed by the sum of the required lanes.

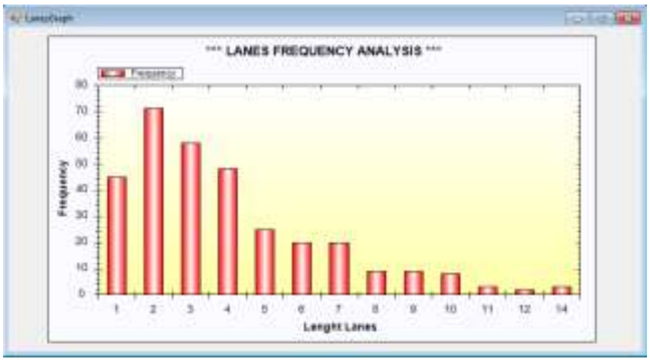

Figure 4. Lane setting output GUI

3.2 Operations scheduling

This module extends the results of the previous step, by involving the inventory turnover in terms of *release horizon* of each lane. The GUI of Figure 5 shows the *release horizon* of each lot represented by a coloured line. The longer the line, the wider the *release horizon* of a lot is. For example, the lanes held by the lot 2153LA of the SKU 112200 remain occupied for 5 days, and so on.

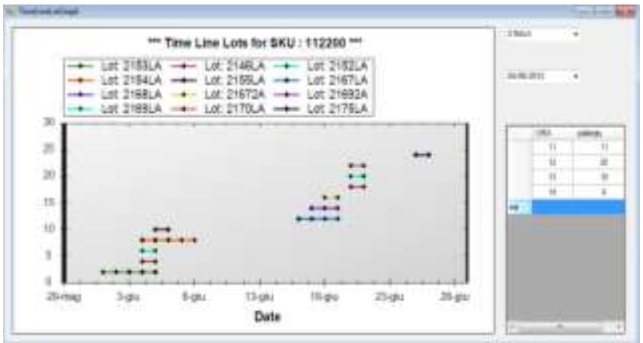

Figure 5. Release horizon GUI

The number of allocated lanes of each depth, cumulated for the all the SKUs, represents the rolling storage capacity required by the production system, in response to the production and demand profiles.

Through the GUI of Figure 6, the tool implements two options: the *Static Layout* and the *Dynamic Layout* approaches.

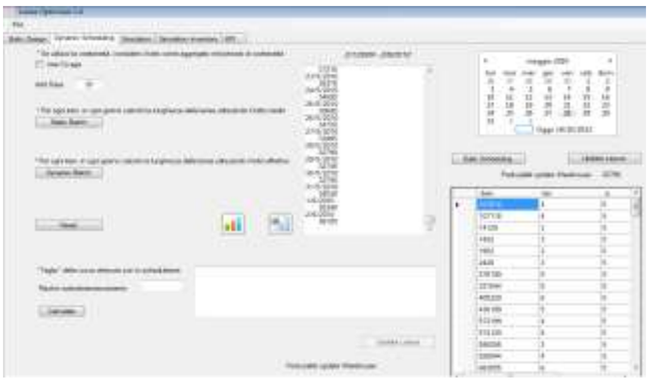

Figure 6. Operations scheduling GUI

The latter quantifies the overall storage system capacity, considering the optimal lane depth for each incoming lot *qi*. Then cumulates rolling the overall lanes requirements per day considering the release horizon of each lot. The resulting total storage capacity, day by day, is shown in Figure 7.

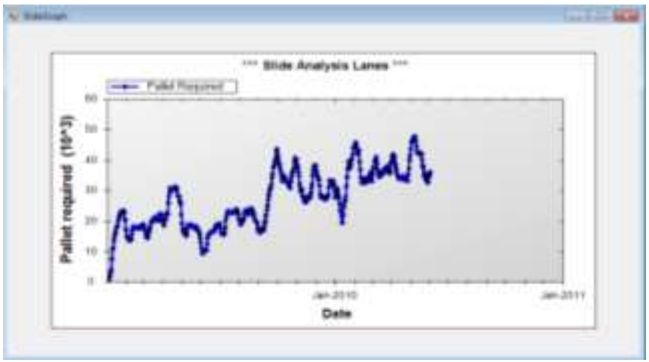

Figure 7. Rolling system storage capacity

Per each day of time horizon, the storage capacity (i.e. a point of the graph) can be split into a frequency analysis of the required lane depths, as illustrated in Figure 4.

Lastly, the calendar panel of Figure 6 enables to select a specific storage configuration corresponding to a day. This consists on a list of different lane depth, and the related number of available lanes. This storage configuration is stored in the table LAYOUT and pre-set for the simulation analysis.

3.3 Simulation

This final GUI enables to assess the performances of space efficiency and saturation of a selected layout configuration. The analysed layout results by the application of the procedure, rather than being set ex-ante by the designer and imported from the database.

The tool analyses the impact of a set of historical inventory snapshots on the given storage configuration. Each lot of SKU *i* fills its optimal lanes. Whether these are not available in type of number, a greedy heuristics to match the available lanes with the lot is implemented. The performances, in terms of occupied lanes, occupied storage locations and lane saturation, are computed per each inventory snapshot in the database. Figure 8 reports the obtained performances in areal-world application of the beverage industry.

Figure 8. Simulation results GUI

4. Application and discussion

This section deals with the application of the illustrated tool to the warehouse of a worldwide renowned company of Italian pasta. The analysis focuses on the determination of the optimal lane depths in a Drive-In rack storage area, and on aiding the design of an effective storage layout. In such zone, 152 items compose the whole SKUs population, and the observed time horizon is of 6 months.

We adopted the operations scheduling module with the dynamic layout approach which resulted in the multi-dimensions lane-depth frequency analysis of Figure 9.

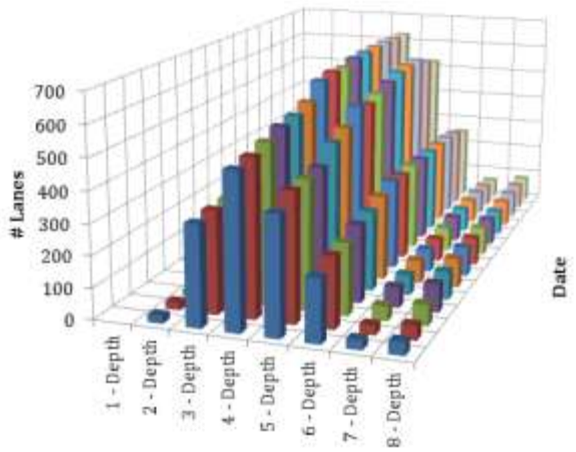

Figure 9. 3D frequency analysis of the daily storage configuration.

The selected storage configuration (i.e. the layout highlighted in the graph) corresponds to a storage capacity of 46,260 pallets. It has been validated via simulation and the resulting space efficiency performances are reported in Figure 10.

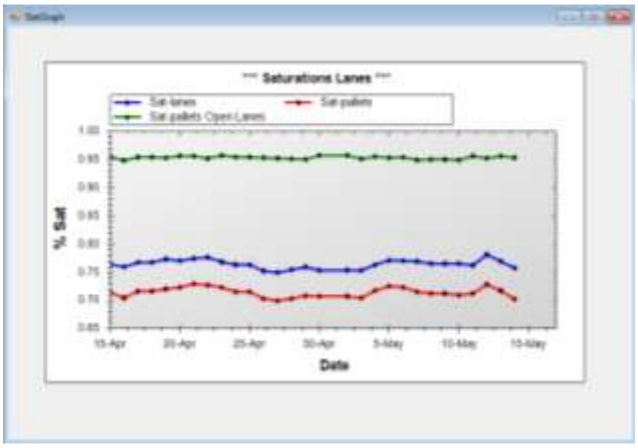

Figure 10. Space efficiency metric resulting by simulation.

Figure 10 reports the values of space saturation in terms of the ratio of occupied storage locations (i.e. red line), the ratio of occupied lanes (i.e. blue line), and the ratio of the occupied storage locations in the open lanes to the overall storage locations of the open lanes (i.e. green line). Whether the obtained results do not satisfy the designer or the company management, the second step of the procedure can be iterated to identify an alternative layout. Each scenario represents a benchmark to evaluate the space efficiency performance of the storage system.

Even though the obtained results refer to the given case study and are not generalizable, the insight benefit of this tool is providing a set of guidelines for managers, practitioners and researcher in facing with real world block storage systems.

5. Conclusions

The proposed top-down iterative procedure gains traction by the extant literature that provides analytical models for the determination of the optimal lane depth in block storage systems. This procedure and the proposed decision-support tool go beyond the assumptions undertaken by the existing models, and provide a useful and practical devices not only to set the optimal lane depth of a set of SKUs, but also to explore the impacts that this has on the space efficiency performances of the storage layout.

Further researches are expected to support the configuration of the optimal layout scenario through the adoption of optimization models (as in Accorsi et al., 2017) for the maximization of the space efficiency in presence of infrastructural constraints (e.g., existing facility).

References

Accorsi, R., Baruffaldi, G., Manzini R., (2017). Design and manage deep lane storage system layout. An iterative decision-support model. *International Journal Advanced Manufacturing Technology*, 1-11. Published online February 16, 2017. DOI 10.1007/s00170-016-9962-9.

Bartholdi, J.J., Hackman, C.S., (2013). *Warehouse and Distribution Science*.

Goetschalckx M., Ratliff, H. D. (1991). Optimal Lane Depths for Single and Multiple Products in Block Stacking Storage System. *IIE Transactions*, 23 (3), 245-258.

Goetschalckx, M. (2003). Logistics System Design. *Lecture Notes 1994-2003*. Georgia Institute of Technology, Atlanta, Georgia.

Gu J., Goetschalckx, M., McGinnis, L.F. (2010). Research on warehouse design and performance evaluation: A comprehensive review. *European Journal of Operational Research*, 203 (3), 539-549.

Jang, D.-G., Kim, S. W., Kim, K.H., 2013. The optimisation of mixed block stacking requiring relocations. *International Journal of Production Economics*, 143 (2), 256-262.

Kim, K.H, Hong, G-P, 2006. A heuristic rule for relocating blocks. *Computers and Operations Research*, 33 (4), 940-954.

Kind, D.A., (1975). Elements of Space Utilization, *Transportation and Distribution Management*, 15 (1), 29-34.

Manzini, R., Accorsi, R., Baruffaldi, G., Cennerazzo, T., Gamberi, M. (2016). Travel time models for deep-lane unit-load autonomous vehicle storage and retrieval system (AVS/RS), *International Journal Of Production Research* 2016, 54, 4286-4304.

Marsh, W. H. (1979). Elements Of Block Storage Design. *International Journal of Production Research*, 17 (4), 377-394.

Matson J.O., White L.A. (1981). Storage system optimization. *Production and Distribution Research Centre Report*, 81-89, Georgia Institute of Technology, Atlanta, Georgia.

Matson, J.O., Sonnentag, J.J., White, J.A., Imhoff, R.C., (2014). An analysis of block stacking with lot splitting. *IIE Annual Conference and Expo 2014*, 497-506.

Meneghetti, A., (2009). Optimizing allocation in floor storage systems for the shoe industry by Constraint Logic Programming. *Proceeding of 9th International Conference on Intelligent Systems Design and Applications* (ISDA 2009), IEEE Computer Society, ISBN 978-0-7695 3872-3, 2009.

Sonnentag, J.J., Imhoff, R.C., White, J.A., Matson, J.O., (2014). A consideration of the block stacking multi-product storage problem. *IIE Annual Conference and Expo 2014*, 3221-3230.

Van den Berg, J.P., Zijm, W.H.M., (1999) Models for warehouse management: classification and examples, *International Journal of Production Economics*, Vol. 59, pp. 519-528.

White, J.A., Sonnentag, J.J., Matson, J.O., (2013). New insights regarding block stacking. *IIE Annual Conference and Expo 2013*, 1225-1234.

Yang, J.H., Kim, K.H., (2006). A grouped storage method for minimizing relocations in block stacking systems. *Journal of Intelligent Manufacturing*, 17, 453-463.