The effect of slot-code optimisation on travel times in common unit-load warehouses

Andrea Fumi, Laura Scarabotti* and Massimiliano M. Schiraldi

Department of Engineering and Management, ‘Tor Vergata’ University of Rome, Via del Politecnico, 00133, Rome, Italy
E-mail: andrea.fumi@uniroma2.it
E-mail: laura.scarabotti@uniroma2.it
E-mail: schiraldi@uniroma2.it
*Corresponding author

Abstract: The main aim of this paper is to estimate material handling times reductions in one-block unit-load warehouse organised with an optimal slot-code allocation, rather than with a uniform pick/store locations distribution, while comparing single and dual-command cycles from a travel distance perspective; results are calculated through multiple what-if analysis based on random scenarios simulations assuming variable input/output positions and warehouse shapes. Simulations helped in the effective quantification of travel times reductions, gaining a result of extreme importance for those manufacturing, distribution and retailing companies which aim at both designing their warehouse and determining the right type and number of transportation resources. Because of currently used warehouse management systems (WMS), companies do not reckon so needful existing literature relying on uniform pick/store distribution: this paper seems the first to address a precise estimation of material handling times when fast-movers items are more or less effectively placed nearby warehouses entrance.

Keywords: single command cycles; dual command cycles; travel distance; random product allocation; optimised product allocation; one-block warehouse; unit-load warehouse; material handling; slot-code allocation optimisation; storage assignment.


Biographical notes: Andrea Fumi graduated with honours in Management Engineering at ‘Tor Vergata’ University of Rome, is currently attending an MBA and PhD candidate in Engineering and Management. He is also working with the Industrial Engineering Research Group of ‘Tor Vergata’s’ University Enterprise Engineering Department as a Research Assistant in Manufacturing Systems Engineering. His main research interests: operations management, production management, project management, logistics and transportation.

Laura Scarabotti graduated with honours in Management Engineering at ‘Tor Vergata’ University of Rome, is currently attending an MBA and PhD candidate in Engineering and Management. She is also working with the Industrial Engineering Research Group of ‘Tor Vergata’s’ University Engineering Department as a Research Assistant in Manufacturing Systems Engineering.
1 Introduction

“Good or bad, the warehouse ultimately portrays the efficiency or inefficiency of the entire supply chain” (Frazelle, 2002). Warehousing is a strategic aspect of supply chain management and plays a crucial role in the success, or failure, of businesses (More and Babu, 2009, 2012); thus is decisive that warehouses are well managed to decrease the overall function cost (Richards, 2011; Pettersson and Segerstedt, 2012; Dharmapriya and Kulatunga, 2011). Effective warehouse management is, to a large extent, determined at the organisational phase (Strack and Pochet, 2010; Gu et al., 2010) and the related costs seem to be mainly influenced by the usage of tracing technologies (Mancini et al., 2012; Wamba and Boeck, 2008; Smith and Offodile, 2009) on top of the a proper items allocation (Gagliardi et al, 2007; Grant and Fernie, 2008; Battista et al., 2012). According to literature, the main aims of an efficient warehouse optimisation is to minimise the average travel distance and required space to store items; secondarily, other goals are maximising the utilisation of available space, equipment, labour, as well as the accessibility to all items (De Koster et al., 2007; Van den Berg, 1999). Order picking has long been known as the most labour-intensive and time-consuming warehouse activity (Tompkins et al., 2002) and, at the same time, the main candidate for productivity improvement studies. As De Koster et al. (2007) underlined in 2007, up to the 55% of the total warehouse operating cost results from order picking operations. Given the extreme relevance of warehouse operations costs saving, this paper estimate material handling distance reductions, while comparing single and dual-command cycles, obtaining an effective quantification through a what-if analysis based on simulations.

2 Previous research

Some authors have devoted publications to the modelling of effective approaches to improve the efficiency of material handling times to achieve better performance in the supply chain; literature is now full of important scientific contributions to minimise material handling costs and storage/retrieval operations, studying new optimisation criteria (Hou et al., 2010; Kutzelnigg, 2011; Ceylan et al., 2012; Routroy et al., 2011;
Sangwan and Kodali, 2009; Zhang et al., 2009). Several authors have focused their researches on the general case of order picking cycle that is a fundamental topic in order to increase operation efficiency and decrease the labour workload (Qiana and Jie, 2011). Some authors have determined the shortest travel distance and the optimal pick tour for a given set of pick locations in two cross aisles’ warehouses (Ratliff and Rosenthal, 1983) or with more than two cross aisles (Vaughan and Petersen, 1999) with a dynamic programming approach. Then, Roodbergen and De Koster (2001a) applied the Ratliff and Rosenthal’s algorithm to middle cross aisles’ warehouses. From the non-random storage policy perspective instead, Hwang et al. (2004) and Caron et al. (2000) modelled an analytical expression assuming turnover based storage policies. Some comparisons on the optimal routing and heuristics for picking problems are also present in literature (Petersen, 1997; De Koster and Van der Poort, 1998; Pan et al., 2012). It seems though, that no closed-form evaluation technique is yet available to estimate the optimal tour length for a general number of picks (Pohl et al., 2009) and that so far, the simulation required to forecast this length has been performed using routing heuristics under the assumption of a random storage policy (Roodbergen and Vis, 2006; Le-Duc and De Koster, 2007; Roodbergen et al., 2008; Petersen and Aase, 2004). Some authors have also developed a mechanism for reducing order picking travel distance through a class-based storage method based on integer programming (Muppani and Adil, 2008a).

Afterwards, Muppani and Adil, (2008b) tried to reduce order picking operations for class-based storage arrangement developing a non-linear integer programming using the branch and bound algorithm. Ho et al. (2008) proposed a further method focused on developing distance or area-based rules to minimise the travel distance of pickers; this method relies on order batching techniques for an order-picking warehouse. In the last two years, Burinskiene (2010) suggested some new approaches for optimisation of order picking processes: the volume-based storage method and the usage of correlation between order picking efficiency and stock accuracy. Ene and Öztürk (2012) instead used both the batching and routing problems to minimise travel costs in warehouse operations. Order picking can be considered as a general case of ‘dual-command’ procedure: for manual picking systems, a single-command cycle means that workers travel from a common pickup/Deposit (P/D) or input/output (I/O) point to a single location and back again to execute a storage or retrieval request (out and back order picking discipline, see Malmborg et al., 1988; Malmborg and Bhaskaran, 1990); the dual-command cycle, instead, usually includes both a storage and retrieval request, meaning that order picking vehicles first move pallets from the I/O point to a location, performing a storage operation, then carry on to a second location picking a pallet before returning to the I/O point (interleaving practice, see Malmborg et al., 1988). Interleaving storage and retrieval requests makes a more efficient use of time, minimising unloaded – and, therefore, unproductive – travel times (Chen and Li, 2011; Salah, 2011; Pohl et al., 2011).

Some authors have devoted publications to the single-command travel distance estimation (Francis, 1967; Bassan et al., 1980), presenting results on optimal warehouse shape and I/O position; however, few contributions seem to be present on the dual-command travel distance estimation in manual pick systems (Mayer, 1961; Malmborg and Krishnakumar, 1987). As a matter of fact, a closed expression to describe optimal dual-command travel distance under the assumption of random storage policy only seems to have been recently developed by Pohl et al. (2009): according to these authors, concerning single command procedure, dual-command cycles can reduce empty
forklift travels from 50% of the total travel distance to about 33%. The single-command and dual-command travel distance/time estimation problem have also been applied to automated storage and retrieval system (AS/RS): Hwang and Lee (1988) for example, have modelled this problem assuming a crane simultaneously travelling in both horizontal and vertical direction, as reasonable; Azzi et al. (2011) instead suggested a new method to estimate the travel time of a new version of multi-shuttle systems using different scenarios with Monte Carlo simulation. Furthermore, Hausman et al. (1976), Graves et al. (1977), Schwarz et al. (1978) and Hwang and Song (1993) have compared random, dedicated and class-based storage policies in single-command and dual-command AS/RS using both analytical models and simulations. The aforementioned contributions are summarised in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>Single command</th>
<th>Dual-command</th>
<th>Picking</th>
</tr>
</thead>
</table>

Nowadays, the majority of industrial companies use warehouse management systems (WMS) to monitor and optimise the position of their stock-keeping-units (SKUs) in storage locations. WMSs are helpful in granting that fast-movers items are located nearby warehouses I/O points, thus minimising travel times for stacking/picking operations. Though, analytical result of estimated travel times computation assuming randomly distributed SKUs, only provide an upper bound of the correct value: this upper bound can be much higher than the actual value performed by the company. Indeed, results of investigations where the randomly-distributed SKUs hypothesis has been made, are not considered very useful by industrial companies. The main aim of this work is thus to
estimate material handling times reductions using optimal items allocation rather than a pick/store locations uniform distribution, while comparing single and dual-command cycles in terms of travel distance.

Results are obtained through multiple what-if analysis based on variable I/O positions and warehouse shape simulations: the simulation tool has been validated on a random-picking scenario, both considering single and dual command cycles, thus to compare it with an already existing analytical formulation (Pohl et al., 2009).

3 Warehouses, layout design and routing methods

In a traditional unit-load warehouse, materials or products are shipped in single discrete units (e.g., pallets) and racks are organised as to create parallel picking aisles, usually with one or more orthogonal cross aisles, with the aim of reducing overall internal travel times. The warehouse layout design problem has been deeply treated in literature: warehouse design and performance have been analysed respectively by Gu et al. (2010) as well as by Meller and Gue (2009) who showed different aisles design such as ‘Flying-V’ design, ‘Fishbone’ and ‘Chevron’ aisles. Because of their widespread use in industry, this paper specifically focuses on traditional one-block unit-load warehouses (Figure 1).

Figure 1 One-block warehouse vs. two-block warehouse

In the most general case of single/dual-command cycles – as to general order-picking – workers can pick/store on both sides of the aisle as well as change direction (no one-way aisles). In order to minimise the internal travel distance with dual-command cycles, a specific routing policy has to be chosen. Many order-picker routing policies can be also applied to dual-command cycles for retrieval requests: a detailed description of the ‘s-shape’, ‘return’, ‘mid-point’, ‘largest gap’, ‘combined’ and ‘optimal’ methods is given by De Koster et al. (2007). The ‘s-shape’ is one of the easiest heuristics that implies a complete aisle crossing in case the aisle contains at least one pick; aisles without picks are not entered at all and once performed the last pick workers return to the I/O point. In the ‘return’ heuristic instead, workers are able to enter and leave each aisle from the same side and, as in the ‘s-shape’ method, only aisles where at least one pick is required are entered. According to the ‘midpoint’ methodology, the warehouse is ideally split in two areas: aisles containing items to pick in the front half are entered from the front cross aisle while picks in the back half are entered from the back cross one. An improvement of the ‘midpoint’ strategy is the ‘largest gap’, in which a worker enters an aisle as far as the
largest gap within the aisle is: the gap is defined as the distance between any two adjacent pick locations, between the first pick and the front aisle, or between the last pick and the back aisle. The ‘largest gap’ part of the aisle is the one not visited by the picker.

**Figure 2** Order-picker routing policies

![Order-picker routing policies](image)

**Figure 3** Example of paths in dual-command cycles

![Example of paths in dual-command cycles](image)

All of these techniques were originally developed for one-block warehouses but they can be used for multiple-block ones by implementing specific changes. Vaughan and Petersen (1999), Roodbergen and De Koster (2001a, 2001b) also showed that the combined heuristic returns the best results in 93% of 80 analysed instances. According to the aforementioned ‘combined’ method, a decision between entirely traverse an aisle with at least one pick, or enter and leave the aisle from the same side, should be computed using dynamic programming (Roodbergen and De Koster, 2001a). Considering dual-command cycles instead of the general order picking case though, there’s no need of dynamic programming to choose between the two alternative decisions: after the storage phase,
workers access the second location pick from the front or the back cross aisle depending on the minimum travel distance between the two store/pick locations (Figure 2). Obviously – as in all the heuristics – aisles without picks are not entered.

4 Simulation methodology

The aim of this paper is to underline the differences, in terms of storage and retrieval distances, among the multiple scenarios of optimised slot-code location (OPT) and the case of random uniform location (RAN) of products inside a warehouse. As it has been stated, modern information systems tend to grant an optimisation of slot-code allocation and this clearly reduces stacking/picking travel times. However, a precise estimation of this reduction entity is considered to be of extreme importance for manufacturing, distribution and retailing companies, since it can help in the design of the warehouse as well as in determining the most appropriate type and number of handling vehicles.

The comparison is performed through simulation, assuming that forklifts operations can be performed in single (SC) or in dual (DC) command mode. In each scenario, the average distance to complete a SC or a DC cycle – respectively $D_{SC}$ and $D_{DC}$ – have been computed, along with their standard deviation over a number of 10'000 runs. This paper offers an original contribution to estimate the cycle time reduction opportunities coming from the slot-code location optimisation, since no theoretical formula to compute $D_{DC}$ or $D_{SC}$ exists under this hypothesis. The simulation model has been validated in a random (RAN) location scenario: here, simulations results show a 0.6% gap with theoretical results by Bassan et al. (1980) and by Pohl et al. (2009) respectively in the SC and DC cases.

4.1 Hypotheses on warehouse layout and input parameters

A generic stacking warehouse has been considered. According to literature, the rectangular shape is the optimal geometric shape to store pallets (Berry, 1968), thus the storage area is assumed to be rectangular, with one input and one output point. Be:

- $X$ the storage area longitudinal width
- $Y$ the storage area lateral depth
- $(x_{out}, y_{out})$ the output coordinates
- $(x_{in}, y_{in})$ the input coordinates.

In the developed simulator, all these variables are independent and can be varied to perform multiple what-if analysis. Clearly, $x_{out} \leq X$ and $y_{out} \leq Y$ as well as $x_{in} \leq X$ and $y_{in} \leq Y$. As far as the simulation tool capabilities are concerned, the input and output points could even be placed inside the storage area (e.g., to represent an elevator moving loading units inside the warehouse from an upper/lower floor); however, in order to consider the most common warehouse case, all simulation runs have been performed under the assumption of a single I/O point located in the middle of the warehouse long side. This choice is also supported by evidences shown by Bassan et al. (1980): indeed, they displayed how this configuration represents the optimal solution to minimise storage/retrieval travel times in warehouses with a longitudinal width twice as lateral.
depth. Moreover, a middle I/O point configuration has also been used by Goetschalckx and Ratliff (1998), Hall (1993) and Petersen (1999). A corner located depot instead, was used by Chew and Tang (1999), De Koster et al. (1999), Gibson and Sharp (1992), and Rosenwein (1994), while both middle and corner options are shown in Jarvis and McDowell (1991), Petersen (1997), and Petersen and Schmenner (1999): these contributes may lead to possible future researches as stated in the conclusion section.

Without loss of generality, the storage location area has been assumed to be square, with side $\delta$. The 2D coordinates of a single storage location in the warehouse can thus be identified by a couple $(x; y)$. Forklifts are assumed to move following the ‘Manhattan’ (i.e., ‘rectilinear’, ‘right-angle’ or ‘l1’) metric (De Rezende et al., 1985), thus, workers are able to travel along the two main orthogonal aisles at the front and back of the storage area as well as along cross ones.

4.2 Random generation of input variables

For the dual-command case, one storage $(x_{sto}; y_{sto})$ and retrieval $(x_{ret}; y_{ret})$ point inside the warehouse are randomly identified for each run, while for the single-command case only one storage or retrieval point is randomly generated. The random generation rule, as performed by Hall (1993) in the development of an heuristic to estimate a shortest tour length lower bound, followed a uniform distribution for the case in which no slot-code allocation optimisation was performed; correspondingly, a specific custom distribution was used for the case with fast-movers items located nearby the I/O point. Thus, for the first case, we have:

$$x_{sto} = \xi_1 \text{ where } \xi_1 \text{ is a random variable } 0 \leq \xi_1 \leq X$$

$$y_{sto} = \xi_2 \text{ where } \xi_2 \text{ is a random variable } 0 \leq \xi_2 \leq Y$$

$$x_{ret} = \xi_3 \text{ where } \xi_3 \text{ is a random variable } 0 \leq \xi_3 \leq X$$

$$y_{ret} = \xi_4 \text{ where } \xi_4 \text{ is a random variable } 0 \leq \xi_4 \leq Y.$$ 

In the second case, in order to represent an optimised slot-code allocation, the $x$ and $y$ coordinates of the storage/retrieval points were generated with a random inverse transform sampling from normal distribution. Considering that simulations assumed a single I/O point located on the warehouse front side ($x_{in} = x_{out}; y_{in} = y_{out} = 0$), we have:

- $x_{sto} = \text{ results from } \xi_1 = \frac{1}{\sigma_x \sqrt{2 \cdot \pi}} \cdot \exp \left[ -\frac{(x_{sto} - x_{in})^2}{2 \cdot \sigma_x^2} \right] \text{ where } \xi_1 \text{ is a random variable } 0 \leq \xi_1 \leq 1$

- $y_{sto} = \text{ results from } \xi_2 = \frac{1}{\sigma_y \sqrt{2 \cdot \pi}} \cdot \exp \left[ -\frac{(y_{sto} - y_{in})^2}{2 \cdot \sigma_y^2} \right] \text{ where } \xi_2 \text{ is a random variable } 0 \leq \xi_2 \leq 1$

- $x_{ret} = \text{ results from } \xi_3 = \frac{1}{\sigma_x \sqrt{2 \cdot \pi}} \cdot \exp \left[ -\frac{(x_{ret} - x_{in})^2}{2 \cdot \sigma_x^2} \right] \text{ where } \xi_3 \text{ is a random variable } 0 \leq \xi_3 \leq 1$
The effect of slot-code optimisation on travel times in common unit-load

- \( y_{ret} \) = results from 
  \[
  \zeta_t = \frac{1}{\sigma_y \sqrt{2 \pi}} \cdot \exp\left( -\frac{(y_{ret})^2}{2 \cdot \sigma_y^2} \right)
  \]
  where \( \zeta_t \) is a random variable

  \[ 0 \leq \zeta_t \leq 1. \]

The user-specified input parameters \( \sigma_x \) and \( \sigma_y \) measure the spreading of products all over the storage area. Thus, the condition

\[
\begin{align*}
\sigma_x & >> X \\
\sigma_y & >> Y
\end{align*}
\]

represents a case in which few or no products have been correctly assigned to slots (i.e., the slot-code allocation is far from being optimised) while

\[
\begin{align*}
\sigma_x & << X \\
\sigma_y & << Y
\end{align*}
\]

displays a situation in which few fast-mover products have been located far from the I/O point. Clearly, due to normal distribution right and left tails, generating the random coordinates with the inverse transform method led to the unacceptability of some coordinates: more specifically, values were discarded if at least one of the following conditions was verified:

- \( x_{sto} < 0 \) and \( x_{sto} > X \)
- \( x_{ret} < 0 \) and \( x_{ret} > X \)
- \( y_{sto} < 0 \) and \( y_{sto} > Y \)
- \( y_{ret} < 0 \) and \( y_{ret} > Y \)

4.3 Hypotheses on forklift paths

In the designed warehouse, single-command and dual-command storage/retrieval operations were simulated with the following approach:

- for the single-command case, a 50% probability to generate a storage or retrieval command was set in order to equally compare it to the dual-command case. Thus, for a storage command, a forklift truck is assumed to:
  1. start at the input point \((x_{in}; y_{in})\), already loaded
  2. horizontally and vertically move to the location \((x_{sto}; y_{sto})\)
  3. once reached the storage location \((x_{sto}; y_{sto})\), perform the storage operation (lifts were not considered)
  4. horizontally and vertically move to the location \((x_{out}; y_{out})\)
  5. exit from the output point.

Analogous operations are performed for the retrieval command. Thus, the distance \( D_{SC} \) travelled to perform one single-command (storage) cycle results to be equal to:

\[
D_{SC} = |x_{in} - x_{sto}| + |y_{in} - y_{sto}| + |x_{sto} - x_{out}| + |y_{sto} - y_{out}|
\]
For the dual-command case, the forklift truck is assumed to:

1. start at the input point \((x_{in}; y_{in})\), already loaded
2. horizontally and vertically move to the location \((x_{sto}; y_{sto})\)
3. once reached the storage location \((x_{sto}; y_{sto})\), perform the storage operation
4. horizontally and vertically move to the location \((x_{ret}; y_{ret})\)
5. once reached the retrieval location \((x_{ret}; y_{ret})\), perform the retrieval operation
6. horizontally and vertically move to the location \((x_{out}; y_{out})\)
7. exit from the output point.

When moving from \((x_{sto}; y_{sto})\) to \((x_{ret}; y_{ret})\), the forklift is assumed to always choose the best alternative (shorter distance) between reaching the front or rear longitudinal aisle. Thus, the overall distance \(OD_{DC}\) travelled to perform one dual-command cycle results to be equal to:

\[
OD_{DC} = \left| x_{in} - x_{sto} \right| + \left| y_{in} - y_{sto} \right| + \min \left\{ \left| y_{sto} + 2 \cdot Y - y_{ret} - y_{out} \right|, \left| y_{ret} - y_{out} \right| \right\}
\]

When comparing the dual and single-command case, a \(D_{DC} = \frac{OD_{DC}}{2}\) has been computed to represent the average distance for a single operation, in order to equally compare it to \(D_{SC}\) value, which refers to the movement of one single loading unit. Thus:

\[
D_{DC} = \frac{OD_{DC}}{2}
\]

An overall number of 10,000 runs have been simulated, computing \(D_{DC}\) and \(D_{SC}\). When calculating \(D_{DC}\) and \(D_{SC}\) in the optimised slot-code location case, according to what previously mentioned, more than the 50% of the 10,000 samples was discarded in each simulation run, due to storage/retrieval coordinates outside warehouse boundaries: as a consequence, \(D_{DC}\) and \(D_{SC}\) were calculated basing on acceptable values only.

5 Simulation results
5.1 Results of random storage/retrieval points generation

As already stated, these results refer to simulations performed under the hypothesis of a unique I/O point located in the middle of warehouse rear side, assuming a longitudinal width twice the lateral depth:

\[
(x_{in}; y_{in}) = (x_{out}; y_{out}) = (X / 2; Y)
\]

\[
X = 2 \times Y
\]

The random generation rule followed a uniform distribution, or a random inverse transform sampling from the normal one, to respectively represent the case in which no slot-code allocation optimisation was performed, or the case in which fast-movers items were located nearby the I/O points.
Figure 4 shows one of the several theoretical probability surfaces resulting from the normal distribution used in the random inverse transform sampling: higher slot access probabilities are related to slots nearer to the warehouse I/O point, located in the centre of the back side of an 80 × 40 metres storage area.

The surface depicted in Figure 4 has the following equation in \( z = f(x, y) \) form:

\[
z = \frac{1}{\sigma_y \sqrt{2\pi}} \exp\left[-\frac{(y + \mu_y)^2}{2\cdot\sigma_y^2}\right] + \frac{1}{\sigma_x \sqrt{2\pi}} \exp\left[-\frac{(x + \mu_x)^2}{2\cdot\sigma_x^2}\right]
\]

With the following parameters:

\[
\sigma_x = \sigma_y = 20
\]

\[
\mu_x = \mu_y = 40
\]

These specific parameters create a smooth shape of the relative surface, suitable to describe a general case of warehouse in which the difference between the turnover rate of slow and fast-mover items is significant but not extreme; different values of these parameters have been tested while simulating to examine the opportunity of choosing a different scenario. As a result, in each simulation of the optimal slot-code allocation case, fast movers codes were located nearer to the I/O point. Figure 5 shows the results of the frequency distribution in one simulation run: as expected, the resulting frequency distribution followed the shape of the theoretical probability distribution depicted in Figure 4.
On the contrary, to simulate a random slot-code allocation, the theoretical probability surface depicted in Figure 6 provided frequency distributions similar to that depicted in Figure 7.

**Figure 5** A frequency distribution of the random points generation for OPT case

**Figure 6** Theoretical probability surface for RAN case
The effect of slot-code optimisation on travel times in common unit-load

Figure 7 A frequency distribution of the random points generation for RAN case

Figure 8 Results of random generation of storage or retrieval points (RAN case)

A clearer 2D representation of storage/retrieval points random generation is depicted in the following Figure 8 and Figure 9 (1,000 out of 10,000 points of a single simulation run are represented for graphical clearness). A uniform distribution is adopted in the first case to underline the absence of any functional slot-code allocation; in the second case, the storage/retrieval points were generated with the random inverse transform sampling from normal distribution, in order to represent the optimised slot-code allocation, similarly to what represented in 3D in Figure 4.
5.2 Results of times calculation

Numerical results of forklifts travel distance simulations are presented in the following tables, comparing the optimised slot-code allocation case (OPT) and the random one (RAN). In the former, three scenarios were considered:

- a first scenario (LOW) characterised by a ‘low effective’ slot-code optimisation (e.g., fast-mover items turnover ratio similar to slow-movers), represented by $\sigma_x = \sigma_y = 26.6$ theoretical probability surface value depicted in Figure 4
- a second scenario (MED) characterised by a ‘medium effective’ slot-code optimisation, represented by a $\sigma_x = \sigma_y = 30.0$ value
- a third scenario (HI) characterised by a ‘high effective’ slot-code optimisation (e.g., fast-mover items turnover ratio deeply different from slow-mover ones), represented by a $\sigma_x = \sigma_y = 16.0$ value.

This comparison led to an effective underlining of differences in terms of:

$\Delta_1$ percentage decrease from $D_{SC}$ to $D_{DC}$, in the RAN case

$\Delta_2$ percentage decrease from RAN to OPT (LOW) case, for $D_{SC}$ or $D_{DC}$ values

$\Delta_3$ percentage decrease from RAN to OPT (MED) case, for $D_{SC}$ or $D_{DC}$ values

$\Delta_4$ percentage decrease from RAN to OPT (HI) case, for $D_{SC}$ or $D_{DC}$ values.

Table 2 and Table 3 respectively show these gaps considering both an 80 × 40 metres rectangular storage area (RECT) and an equivalent 56.6 × 56.6 metres square storage area (SQUA).

Simulation results for RAN scenarios fit almost perfectly expected results both for $D_{SC}$ or $D_{DC}$, both on the RECT and the SQUA storage area. Indeed, given:

$$D_{SC\ (theoretical)} = \frac{X}{4} + \frac{Y}{2}$$
The effect of slot-code optimisation on travel times in common unit-load

\[ D_{DC}^{(theoretical)} = \frac{OD_{DC}^{(theoretical)}}{2} = \frac{\left(\frac{5}{6}X + \frac{5}{3}Y\right)}{2} \]

Simulation results showed:

- for the RECT storage area:
  - an experimental average \(D_{SC}\) value equal to 80.4, given an 80.0 theoretical value
  - an experimental average \(D_{DC}\) value equal to 66.9, given a 66.7 theoretical value

- for the SQUA storage area:
  - an experimental average \(D_{SC}\) value equal to 84.6, given an 84.9 theoretical value
  - an experimental average \(D_{DC}\) value equal to 70.6, given a 70.8 theoretical value.

Indeed, a pattern of simulated \(D_{SC}\) and \(D_{DC}\) frequency distributions, both in the RAN and OPT case, is shown in Figure 10 and 11, respectively for the RECT and the SQUA storage area.

**Table 2** Simulation results of average travelled distances on RECT area

<table>
<thead>
<tr>
<th>Metres</th>
<th>(D_{SC})</th>
<th>(D_{DC})</th>
<th>(\Delta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAN</td>
<td>80.4</td>
<td>66.9</td>
<td>16.8%</td>
</tr>
<tr>
<td>OPT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOW</td>
<td>71.7</td>
<td>63.0</td>
<td>12.1%</td>
</tr>
<tr>
<td>AVG</td>
<td>64.2</td>
<td>57.4</td>
<td>10.6%</td>
</tr>
<tr>
<td>HI</td>
<td>58.0</td>
<td>52.6</td>
<td>9.3%</td>
</tr>
<tr>
<td>(\Delta_2)</td>
<td>10.8%</td>
<td>5.8%</td>
<td></td>
</tr>
<tr>
<td>(\Delta_3)</td>
<td>20.1%</td>
<td>14.2%</td>
<td></td>
</tr>
<tr>
<td>(\Delta_4)</td>
<td>27.9%</td>
<td>21.4%</td>
<td></td>
</tr>
</tbody>
</table>

**Table 3** Simulation results of average travelled distances on SQUA area

<table>
<thead>
<tr>
<th>Metres</th>
<th>(D_{SC})</th>
<th>(D_{DC})</th>
<th>(\Delta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAN</td>
<td>84.6</td>
<td>70.6</td>
<td>16.5%</td>
</tr>
<tr>
<td>OPT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOW</td>
<td>68.8</td>
<td>61.7</td>
<td>10.3%</td>
</tr>
<tr>
<td>AVG</td>
<td>57.9</td>
<td>52.1</td>
<td>10.0%</td>
</tr>
<tr>
<td>HI</td>
<td>50.7</td>
<td>45.9</td>
<td>9.5%</td>
</tr>
<tr>
<td>(\Delta_2)</td>
<td>14.4%</td>
<td>7.8%</td>
<td></td>
</tr>
<tr>
<td>(\Delta_3)</td>
<td>28.0%</td>
<td>22.1%</td>
<td></td>
</tr>
<tr>
<td>(\Delta_4)</td>
<td>36.9%</td>
<td>31.4%</td>
<td></td>
</tr>
</tbody>
</table>
Clearly, the dual-command mode always allows to decrease the average travelled distance; nevertheless, the dual-command over-performance decreases if using an effective slot-code allocation criteria. As expected, both DSC and DDC values showed to be higher in the SQUA case rather than in the RECT one, according to what evidenced by Berry (1968). On the contrary, in each OPT case, both the DSC and the DDC presented a higher value considering a RECT area rather than a SQUA one: this because the OPT case is based on the random generation of storage and retrieval points through the inverse transform sampling from normal distribution with the same variance, i.e., from a
The effect of slot-code optimisation on travel times in common unit-load

probability surface with circular contour lines that best fit on square areas than on rectangular ones. This result, potentially reckoned as a pure theoretical consideration, may actually entail many practical implications: when managing a warehouse, several circumstances may push employees to merely consider geometrical distances from the I/O location, thus intrinsically assuming circular contour lines on the storage area and performing an item-slot assignment in accordance to these perimeters. Indeed, elliptic perimeters would hardly be considered by warehouse operators, thus ignoring the effect of a non-unitary ratio between warehouses width and length. However, when adopting an optimised slot-code allocation criteria, differences in terms of travelled distances between optimal-ratio rectangular storage areas and square ones decrease significantly.

6 Conclusions

For traditional unit-load warehouses, the scientific literature seems to be only focused on random slot-code allocation, mainly for single-command operations: from an industrial point of view this turns out to be a critical issue, especially for companies not making use of information systems to grant SKUs optimal placement in their storage areas. Moreover, existing research contributions mainly aim at reducing warehouse operations costs through the minimisation of material handling average travel distance obtainable by optimising items location. Indeed, given that up to the 55% of a storage area overall expense results from order picking operations, this paper, through multiple simulations based on what-if analysis, aims to be the first to address a precise and measurable estimation of material handling times when fast-movers items are more or less effectively placed nearby warehouses entrance. This work is thus intended to give some insights about the advantages coming from a slot-code optimisation rather than a random one, comparing the performance of material handling systems both in single and dual-command cycles. Each comparison was carried out in three scenarios representing a low, medium and high level of effectiveness of the slot-code optimisation. Despite a predictable reduction of travel times switching from the random scenario to the optimal one, simulations helped in an exact determination of these decrements, which turns out to be the main strength of the whole work: indeed, quantifications showed a reduction in the overall distance travelled by forklifts varying from 6% to 21% in dual-command operations, and from 11% to 28% in single-command ones on a $2 \times 1$ rectangular storage area. Considering a square warehouse though, reductions vary from 8% to 31% in the dual-command case and from 14% to 37% in a single-command one. These results can be of extreme importance for manufacturing, distribution and retailing companies seeking both for an efficient design of their warehouse and the most appropriate type and number of material handling vehicles. The research limitations concern the assumptions of an unique size of storage locations (single shelf type) and of two-ways aisles, hypothesis that may not be valid for every industrial warehouse. However, presented results turn out to be extremely helpful in supporting the storage area design phase. On top of removing the single shelf type assumption and evaluating the influence of one-way aisles presence, future research should imply simulations performed varying the shape of the storage area, the location and number of orthogonal aisles, the positions of the input and of the output point. Most important, more interesting insights could come from the application of the presented methodology to a multiple-picking warehouse.
Acknowledgements

We would like to thanks the reviewers for their constructive and helpful comments.

References


The effect of slot-code optimisation on travel times in common unit-load


The effect of slot-code optimisation on travel times in common unit-load


