Dynamic Storage Location Assignment Problem at Knapp company: toward Reinforcement Learning?

P. Courtin¹², A. Grimault¹, M. Lhommeau¹, J.-B. Fasquel¹
¹ Laboratoire Angevin de Recherche en Ingénierie des Systèmes (LARIS)
Université Angers, 62 avenue Notre Dame du Lac, 49000 Angers, FRANCE
² Knapp France, 23 Rue de la Maison Rouge, 77185 Lognes, FRANCE
paul.courtin@knapp.com
20 mars 2020

Abstract
Automated warehouses picking performance optimization is strongly influenced by a wise assignment of product to storage location. The storage location assignment problem (SLAP), is usually solved with Operational Research methods. This article draws up a literature review to solve this problem by highlights link with Machine Learning. It also present plausible prospect to solve this problematic by combining Machine Learning and Operational Research

Keywords
automated warehouse, storage location assignment problem, storage allocation, machine learning, reinforcement learning, dynamic slotting, SLAP, SBS/RS

Zusammenfassung
Die Optimierung der Kommissionierleistung in automatischen Lagern wird stark durch eine kluge Zuordnung von Produkt und Lagerort beeinflusst. Das Lagerortzuordnungs-Problem (SLAP) wird in der Regel mit Methoden der Betriebsforschung (Operational Research) gelöst. In diesem Artikel wird ein Literaturüberblick zur Lösung dieses Problems erstellt, wobei die Verbindung zum maschinellen Lernen hervorgehoben wird. Er stellt auch eine plausible Perspektive für die Lösung dieses Problems durch die Kombination von maschinellem Lernen und operationeller Forschung dar.

Schlüsselwörter
Automatisches Lager, Problem der Lagerortzuordnung, Lagerzuordnung, Maschinelles Lernen, Bestärkendes Lernen, dynamisches Slotting, SLAP, SBS/RS

1 Introduction
Digital revolution in business, through the implementation of the Industry 4.0 paradigm, has brought a wealth of data to the table. Exploitation of these data can be used to support data-driven approach for decision making and diagnostics. Nevertheless, these data must be first exploited. This large amount of data (Big Data) must be transformed into information (Smart Data) thanks, in particular, to the use of machine learning algorithms [2].

In this context, we are focusing on the warehouse management optimization issues introduced by [9] and at the heart of the concerns of the Knapp company. In general, these problems deal with the assignment of article to storage containers (pallet assignment), the assignment of these containers to storage locations (storage assignment or slotting) and the scheduling of handling operations (deposit/retrieval) of the containers (interleaving). In our study, we will focus on the problem of assigning containers to storage locations: Storage Allocation Assignment Problem (SLAP), often addressed in the literature by operational research methods [9,8,21].

Néanmoins ce type d’approche est encore sous-exploitée [15], malgré les perspectives offertes, dans de multiples domaines, par l’apprentissage profond (Deep Learning) [7], et l’apprentissage profond par renforcement (Deep Reinforcement Learning). As recently pointed out, machine learning offers multiple perspectives for industrial applications (e.g. component failure detection, fault diagnosis, production optimization, damage detection) [2]. Nevertheless, this type of approach is still under-exploited [15]. In spite of the perspectives offered in multiple others fields, in Deep Learning [7], and Deep Reinforcement Learning.

The objective of this article is, for our application context and its own constraints, to identify points on which machine learning could be relevant. Our contribution consists in proposing a warehouse management strategy that allows us to deal with the problem of assigning storage positions in a dynamic way, based on the perspectives offered by machine learning combined with operational research. This proposal will be contextualized by drawing on examples of related applications involving machine learning.

Section 2 is dedicated to a detailed presentation of the problem and the state of the art, highlighting difficulties still little addressed, linked to the lack of anticipation of orders. Section 3 focuses on methods to solve the problem, based on operational research and machine learning. Eventually,
in section 4 we present new investigation tracks.

2 Problematic and State of the Art

2.1 System description

For this study we will focus on automated warehouse of type goods-to-person implementing a Shuttle-based Storage and Retrieval Systems (SBS/RS), which is a derivative of the Automated Storage and Retrieval System (AS/RS) using shuttles. The SBS/RS is composed of one or more aisles. In the center of the aisle shelves, on each floor, runs a tier captive shuttle runs. Shuttles are responsible of storage containers transport from and to the lift (the I/O point) and of their handling in the racks (storage and retrieval).

The figure 1 shows a top view of a warehouse. The SBS/RS is configured with 4 aisles, each racks are configured in single deep. In the racks are stored the storage containers, holding the Stock Keeping Units (SKUs).

Figure 1 illustrates the environment associated with our problem. In a logistics warehouse, several operations are necessary to ensure the preparation of orders:

1. **Goods-in**: Products are entering the system. Beforehand, the SKUs must be available for picking, i.e. stored in the SBS/RS.

2. **Picking**: This is the preparation step. The picking containers are routed from a start station to the picking stations, and the storage containers (holding the stock) are routed to the same picking stations (steps 1, 2, 3, and 4 in the figure 1). Once the picking and storage containers have both arrive at the picking station, the picker can transfer the products from the storage containers into the picking containers.

3. **Shipping**: Expedition step. After picking is completed, the picking container is conveyed to the shipping station and the storage container returns to the SBS/RS for further storage.

4. **Slotting**: Storage location assignment step. The Warehouse Management System (WMS) will select the best storage position for the storage container, based on the characteristics of the SKUs in the container, the order history and the implemented storage strategy.

2.2 Automated Warehouse Optimization

In order to gain or maintain competitive advantage, manufacturers are on the move to improve warehouses performance through better design and optimized run of operations. The optimization of an automated warehouse can be defined by several concomitant objectives to be minimized: average distance of a picking circuit, investment and operating costs, orders or a group of orders picking time, use of equipment and people or loss of space [12].

Achieving these goals requires to take decisions that can be prioritized into strategic, tactical and operational levels [22]. These decisions include: warehouse layout design and sizing (layout design), product allocation to storage locations (storage assignment), order allocation to routes and aisle grouping into work areas (batching and zoning), order picker routing (routing), order sorting and accumulation [12]. As illustrated by the figure 2, order picking operations have been identified [12] as the most time consuming and labor intensive activity.

55% of operational costs can be attributed to order preparation [27]. Picking time is itself impacted by (i) travel time (Travel) of the SKUs to, from and between pick positions (up to 50%), (ii) identification time (Search) of the SKUs needed for order picking (up to 20%), (iii) time for picking (Picking) the SKUs from the storage containers (up to 15%), (iv) preparation time (Setup) of the tour (up to 10%) and (v) other tasks (Other) (up to 5%) [23], as shown in Figure ?? For this reason, we focus on reducing the order picking time, and in particular on minimizing the travel time (or distance) of the SBS/RS shuttles. From this perspective, we will study the allocation of products to optimal storage locations in order to reduce the cost of picking operations.

Assignment of products to storage positions is known in the literature as Storage Location Assignment problem (SLAP),

**FIGURE 1 – schematic of a warehouse with an SBS/RS**

**FIGURE 2 – Distribution of the logistics operation costs cumulated over a year in a warehouse (left) and distribution of the time of each task over the order picking activity (order picking - right) (extract from [11])**
witch was first formulated by [9], and was proven NP-hard by [5]. SLAP can be approached in different ways [8] depending on the nature of the input data: known dates of entry and exit of the product into and out of the system, product characteristics (weight, dimensions), etc. When assigning containers (and thus the products they contain) to storage locations (Slotting), the WMS endeavour to choose the best storage locations. According to a storage strategy that allows good performance during Picking while respecting business and operational rules:

- the overall mass of the products must be distributed over the entire length of the rack to limit mechanical stresses;
- Empty spaces must be kept to facilitate intra-aisle handling operations;
- combustive and combustible products must not be stored next to each other;
- stock must be distributed between the aisles and levels, so no stock reference is unavailable in case of unavailability of a device (lift or shuttle).

All these rules must be taken into account while solving the SLAP. However, some are not often considered in the literature in the formalization of the problems, in particular the distribution of the mass, as underlined in the state of the art of [21]. To solve SLAP, exact methods (dynamic programming, integer mixed linear programming) and approximate methods (heuristics and metaheuristics) have been proposed [26]. Several storage strategies can be considered, inducing particular rules and formulation to deal with the SLAP problem [11]. The most common strategies are:

- **Class-Based (CB)**: the products are separated into several classes and a storage area is dedicated to each class. The most popular is the assignment in 3 classes, ABC where the SKUs are distributed according to a criterion such as the turnover (e.g. the most frequently ordered products are placed in the class A, associated with the area closest to the I/O point), introduced by [9]. Within each class, the products are arranged with a simple rule (e.g. closest available position to the I/O point). Other classifications may be considered. For example, the classification XYZ (detailed in the section 2.3) is based on order fluctuations for a product [19, 24];
- **Duration-of-Stay (DoS)**: products are assigned to a location where the distance from the I/O point is proportional to the time spent by the SKU in the warehouse. Under certain conditions (DoS known in advance and a balanced number of goods input-output), this rule is optimal [6] (compared to other strategies, in terms of travel time and space occupation);
- **Cube-Order-per-Index (COI)**: This metric defines the ratio between the space needed to store a product and its demand [6]. Products with a low COI (i.e. small size and often in demand) are placed near the I/O point;
- **Random-Based**: This policy, assign product to storage location randomly. It’s used in the industry and is used in academic study as comparison basis.

### 2.3 Problem of fluctuating orders

In our operational and industrial context, beyond the business constraints presented above (mass distribution, empty spaces, compatibility between products and spatial distribution), additional complications have been detected when using the Slotting algorithms (implemented in WMS). These complications result from aspects that are often ignored:

1. SKUs frequency of demand and the quantity in pick orders varies and fluctuates over time;
2. Arise of rare but significant events [1]. For example, promotions sales leading to a sudden and brief peak in orders quantity.

**Figure 3 – The number of parts ordered and the performance of the picking by week is an example.**

This last point is illustrated by the figure [2] which represents a history of orders observed by the Knapp company, in a given warehouse and over a specific period. The first ordinate (left) represents the quantity of pieces ordered by week over a period of approximately 6 months. The second ordinate (right) represents the picking performance, expressed (by convention) as the number of picks per hour (pick/h - average over a week). In our example, there is a drop in preparation performance of about 30% in week 10. This drop in performance could be explained by a reduction in the amount of work (number of pieces ordered) in week 10, which would imply a drop in picking performance. However, we can see that the total quantity of SKUs to be prepared increases by +3% between weeks 9 and 10. Such a small variation in the number of parts ordered should not have such an impact on performance. The drop in performance is the result of a peak order for a particular type of pieces, as detailed below.

The figure [4] represents the number of pieces ordered for 3 SKUs as a function of the week and the second ordinate represents the picking performance. The 3 SKUs selected are representative of the types of demand variations observed in the warehouses. The product labelled X has a low or
negligible demand (less than 0.005% of overall demand), with little variation over time. The product labelled $y$ has a higher level of demand (close to 0.12%) and there is a slight variation in quantity over time. Finally, the product labelled $z$ has a sharp variation in demand. We notice that this product, never ordered before week 10, becomes in week 10 a very demanded product (17 000 pieces out of a total of 1 352 273 required, that is to say nearly 1.28% of the overall demand) and this for a duration of only 1 week. Then its demand becomes sporadic, but remains significant (less than 0.37% of the demand).

These behaviors of demand variation are identified in the literature by means of the classification by analysis $XYZ$ [19, 24]. Class $X$ includes products where the variation in demand (over time) is stable, uniform, and continuous with little variation. Class $Y$ regroups products where the demand fluctuates more, especially with the season. Finally, Class $Z$ regroups products where the demand is sporadic, irregular or even abnormal in terms of quantity compared to other products. Variations in demand for $Z$ products are unforeseeable or unpredictable.

This SKU classification applies to our example, the one labeled $x \in X$, the one labeled $y \in Y$ and labeled $z \in Z$. The product, labeled textttz, underwent a sudden and brief change in the number of products requested.

It was thus identified by the WMS as a high turnover product. Therefore this SKU was assigned storage locations as close as possible to the I/O point. The operation of the WMS is perfectly normal, based on the available data (order history, variation in past demand for SKUs) and the storage strategy used. The $slotting$ algorithm assigned a high-demand product to the storage locations with the “fastest” access. Nevertheless, the problem is the following : this explosion of commands for the reference $z$ is valid for a short and short period. The following weeks require only a small amount of this product reference, which is in the last quartile of requested products. However, the SKU labeled $Z$ uses the “best” storage locations, which implies longer trips for the shuttle, which have to overtook this storage area, to reach the other SKUs.

The problem illustrated by this example is : how to use the order history to choose optimal position for the SKUs in the warehouse? This implies following underlying questions : should the position assignments for each SKU be recalculated and thus dynamically change the warehouse layout? This would imply intra-aisle handling movements to relocate containers and downtime for stock and shuttles (busy with tasks with no immediate added value). This means specifying a warehouse management policy that allows this handling to be carried out without disrupting production.

To the authors best knowledge, this aspect is little covered in the literature. It is called Dynamic Storage Location Assignment Problem (DSLAP). This problem has been tackled by heuristic methods cited above [8]. Concerning the exploitation of order history in particular, and in the case of a CB strategy ($ABC$ or $XYZ$), the industry relies essentially on a superficial analysis of the data (average of orders placed for $ABC$ or their average fluctuation for $XYZ$), without seeking to modify the warehouse SKU allocation dynamically.

The following section focuses on a more specific modeling of this problem and possible links between operational research and machine learning.

### 3 Modelling, Operational Research and Machine Learning

With problem described above, including the fluctuation of the controls, we propose a way to couple optimization problem modelling with machine learning, based on recently published works on this subject. Section 3.1 is dedicated to the formulation of the SLAP optimization problem. Section 3.2 deals with the use and perspectives of machine learning in our context. We will be particularly interested in unsupervised learning, supervised learning and reinforcement learning, highlighting the links with learning be means of deep neural network.

#### 3.1 Optimization Problem Modeling

In this section, we focus on a formulation of our optimization problem. The SLAP formulations are based on the General Assignment Problem (GAP) problem [10].

The entities often encountered in the literature are the following. Let $\mathcal{N} = \{1, ..., N\}$ be the set of SKUs (elements) to be stored. Let $\mathcal{L} = \{1, ..., \mathcal{L}\}$ be the set of positions. Let $\mathcal{O} = \{1, ..., \mathcal{O}\}$ be the set of commands. We define a binary variable $x_{ik}$ equal to 1 if the item of type $i \in \mathcal{N}$ is assigned to the position $k \in \mathcal{L}$, 0 otherwise. In each of the aisle racks, the distance between a storage position $k$ and the I/O entry/exit point, located at the elevator level, is noted as $d_k \geq 0$, $k \in \mathcal{L}$. The exit sequence of the SKUs and the required quantity are given by the set of picking orders $\mathcal{O} = \{1, ..., \mathcal{O}\}$. This set determines the probability of demand for each SKU $i$, denoted $p_i \geq 0$, $i \in \mathcal{N}$. Note that this probability is generally based on a simplistic measure (average number of orders per item), and does not detect
peaks if the estimation interval is too large.
A list of properties can be added to each item belonging to
the $\mathcal{N}$ set: compatible item classes (e.g. flammable product
problem), weight, dimensions, etc.

Several ways to formulate the problem can be considered,
depending on the modeling of the warehouse and the cri-
teron(s) one is trying to optimize (e.g. see recent work inte-
grating multi-objective optimization [28, 34]). We give here
a simplified example, inspired by [29] in the case where the
optimization criterion is to minimize the total travel dis-
tance of the shuttles and the weight distribution. For this,
we define $w_i$ corresponding to the mass of the SKU $i$, and
the entity $\mathcal{M} = \{1, ..., M\}$ the set of columns (see figure
5). A column $m_k$ is a partition of the set $\mathcal{L}$.

The objective function to be optimized can be formulated by $z:

$$
z = \min \sum_{i \in \mathcal{N}} \sum_{k \in \mathcal{L}} x_{ik} \cdot d_k \cdot p_i$$

(1)

$$
\sum_{i \in \mathcal{N}} x_{ik} \leq 1 \quad \forall k \in \mathcal{L}
$$

(2)

$$
\sum_{k \in \mathcal{L}} x_{ik} = 1 \quad \forall i \in \mathcal{N}
$$

(3)

$$
| \sum_{i | k \in m_1} w_i x_{ik} - \sum_{i | k \in m_2} w_i x_{ik} | \leq \varepsilon \quad \forall (m_1, m_2) \in \mathcal{M}^2
$$

(4)

Constraint (2) imposes only one product for each position,
constraint (3) imposes only one position for each product.
Constraints (4) imposes to distribute the overall mass of
the SKUs over the entire length of the rack ($\varepsilon$ being a pa-
parameter associated with the weight differences between co-
olumns). These constraints could be extended based on other
properties associated with SKUs, such as the problem of
compatibility between products (flammable products).

This formulation is quite similar to the DoS strategy (where
our $p_i$ can be associated with the time of presence in the
system), in connection with deep learning [13].

In the case of a strategy of type Class-Based, one could
classify the products in categories and adapt the formula-
tion by integrating a new entity $C = \{1, ..., C\}$, designa-
ting the set of defined storage classes: the products are
then pre-assigned to zones, inducing additional constraints
on the assignment positions (variables $x_{ik}$).

3.2 Machine Learning

As pointed out in the introduction, machine learning and
in particular deep learning have proven to be effective in
many areas [7]. In particular, this is part of the recently for-
mulated perspectives on the use of deep learning for order
prediction and product categorization [31, 21).

Several approaches can be considered to exploit pick orders
history based on machine learning. We will consider below
three approaches of recent interest to the scientific com-
community: unsupervised learning, supervised learning and
reinforcement learning. In each case, we give some recent
work, highlighting the possible links with deep learning,
around our issue and around related issues, using these
approaches, thus underlining their potential to address our
topic.

3.2.1 Unsupervised Learning

Unsupervised learning makes possible to identify groups
of objects with similar characteristics on the basis of the
characteristics of the manipulated objects or their interac-
tions.

In our case, unsupervised learning can be used on the
SLAP to classify SKUs into categories, when the storage
strategy is of type Class-Based. The SKUs are distributed
in $C$ classes, mentioned in the previous section, be means
of the machine learning [30].

Product classes can be established by clustering techniques
based on the attributes of the different SKUs ordered and
the order history. In the study of the SLAP of a storage wa-
rehouse by clustering in e-commerce, [30] is using stan-
dard attributes of the Class-Based classification, such as
price and sales volumes, to which it associates complemen-
tary attributes: the rate of favorable comments from SKUs,
the type of payment authorized on delivery (cash payment or
not), whether the shipping costs are offered or not. He
was thus able to guide the process of clustering, and divide
109 SKU into 3 categories.

This method of clustering is based on the intrinsic attri-
butes of the SKUs, other classification methods are based
on the interactions between the SKUs. Association rules
association rule mining allow to extract relationships of in-
terest between SKUs. These rules then provide metrics (e.g.
weighted support count (WSC) [18], product affinity [14]),
which are used to categorize SKUs and assign them to sto-
rage positions.

The classification methods studied make little use of neural
networks. However, deep learning classification methods
are part of the recently formulated perspectives for product
categorization [31, 21].

3.2.2 Supervised Learning

Supervised learning aims at prediction and/or classifica-
tion, based on training on labeled data.

Supervised learning can also be used in the SLAP prob-
lem,
for example in the case of a strategy such as Duration of Stay [15]. This work introduces the term \( \hat{p} \) for the (predicted) duration of stay of a SKU in the system, the Duration of Stay (DoS), which is estimated with a deep convolutional neural network (CNN) and a recurrent neural network (RNN) of the long short-term memory type. (LSTM) [3]. This network is trained on commands history, corresponding to the \( O \) entity introduced previously. For a container with a DoS predicted by machine learning, noted \( \hat{p} \), the \textit{Astoragepositionassignmentfunction} : \( \mathbb{R} \rightarrow \mathbb{W} \) is, according to [15] :

\[
A(\hat{p}) = \arg \min_{w \in \mathbb{W}} d(NrW(\hat{p}), w) + c(w)
\]  

(5)

Where \( d(v, w) \) is the distance between two \( v \) and \( w \) positions. \( c(w) \) represents the associated costs associated with storing at that \( w \) position. This takes into account the costs of the specific constraints of each warehouse, (e.g.: stock-mixing, stock-splitting, mass distribution,...). The approximate optimal storage position in the case of DoS is given by \( NrW(\hat{p}) \). \( \mathbb{W} \) represents the set of storage positions available when a storage container enters the warehouse. \( d(NrW(\hat{p}), w) \) represents the cost of not storing a pallet at the optimal storage position for DoS. The storage position \( A(\hat{p}) \) is then chosen based on those two costs (the associated costs \( c(w) \) and the cost of not storing \( d(NrW(\hat{p}), w) \))[15].

This approach assumption is illustrated by the figure [6] in the present case (DoS) and from the perspective of use in the case of a strategy of type class-based. We can note here that this approach combines deep learning and optimization under constraints.

To our knowledge, the paper of [15] is the first paper dealing with SLAP with deep neural networks, taking into account the dynamic nature of the warehouse and the uncertainties of operation. Note that this work is associated with the provision of one of the only public databases containing warehouse data to address the allocation problem.

The figure [6] presents two lines of research envisaged to solve the SLAP. Variant (a) represents the use of Deep Learning to categorize products into \( c_i \) classes based on order history, inspired by [30] [14] [18]. Variant (b), corresponding to the work of [15], uses *Deep Learning* to predict the value of a metric (the DoS for [15], the probability of demand for SKUs in our case). In both variants (a) and (b), *Deep Learning* techniques will transform/use the data, here the \( O \) command history, to provide input variables to solve our constrained optimization problem.

### 3.3 Reinforcement Learning : Taking the Long Term into Account

Reinforcement learning [25] is another interesting lead that, to our knowledge, has not been studied yet in our context.
shuttle trips for example).

4 Conclusion

We have seen that the optimization of an automated warehouse using an SBS/RS, requires the reduction of the preparation time and therefore the reduction of the time/distance of the shuttle travel. Improving the allocation of storage positions (Sloting) is a means of action to minimize shuttle trips, known in the literature as the Storage Location Assignment Problem (SLAP). We have seen, through a concrete example, some difficulties associated with the hardly predictable fluctuation of orders. This requires to move towards a dynamic allocation system based on dynamic learning from past orders.

Then we presented some approaches considered to tackle this problem, generally formulated as an optimization problem under constraints, falling within the scope of operational research. We also presented the state of the art of related work involving machine learning. This review of recent work published in the literature shows the potential of machine learning for this problem, and in particular of learning supervised by deep neural networks and also of reinforcement learning. This study also shows that little work has been done on the use of deep learning in this context. Finally, this synthesis work illustrates how deep learning can be advantageously combined with the modeling of the problem, done in the form of a constraint optimization problem: learning can be used to model commands in order to guide the optimization process. The medium-term continuation of this work will concern the deepening of this study and the realization of a simulator before the evaluation on a real environment.

About

This article was translated from french with the help of DeepL.Translator

Références


