



HAL
open science

Problème d'affectation dynamique des emplacements de stockage chez Knapp : vers de l'apprentissage automatique ?

Paul Courtin, Axel Grimault, Mehdi Lhommeau, Jean-Baptiste Fasquel

► To cite this version:

Paul Courtin, Axel Grimault, Mehdi Lhommeau, Jean-Baptiste Fasquel. Problème d'affectation dynamique des emplacements de stockage chez Knapp : vers de l'apprentissage automatique ?. Plate-Forme Intelligence Artificielle, Jun 2020, Angers, France. hal-02527743v5

HAL Id: hal-02527743

<https://univ-angers.hal.science/hal-02527743v5>

Submitted on 23 Jun 2020 (v5), last revised 26 Jan 2021 (v6)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Problème d'affectation dynamique des emplacements de stockage chez Knapp : vers de l'apprentissage automatique ?

P. Courtin^{1,2}, 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

paul.courtin@knapp.com

20 mars 2020

Résumé

L'optimisation des performances de préparation des entrepôts automatisés passe par une affectation judicieuse des produits aux emplacements de stockage. Ce problème d'affectation des positions de stockage (SLAP), est généralement abordé par les méthodes relevant de la recherche opérationnelle. Cet article dresse un état de l'art de cette problématique en soulignant les liens possibles avec l'apprentissage automatique, et des perspectives envisageables de résolution combinant apprentissage automatique et recherche opérationnelle.

Mots-clés

entrepôt automatisé, problème d'affectation des positions de stockage, affectation de position de stockage, apprentissage automatique, apprentissage par renforcement, affectation dynamique de position, SLAP, SBS/RS

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

1 Introduction

La révolution numérique des entreprises, par la mise en place du paradigme *Industrie 4.0*, a apporté une profusion de données disponibles. L'exploitation de ces données peut constituer une aide à la prise de décision et à la pose de diagnostics, basée sur les données (approche *data-driven*). Néanmoins, il faut au préalable valoriser ces

données. Transformer les masses de données (*Big Data*) en informations (*Smart Data*) grâce, en outre, à l'utilisation d'algorithmes d'apprentissage automatique (*Machine Learning*) [2].

Dans ce contexte, nous nous intéressons aux problématiques d'optimisation de gestion d'entrepôts introduites par [9] et au coeur des préoccupations de la société Knapp. Ces problématiques traitent d'une manière générale de l'affectation d'articles à des conteneurs de stockage (*pallet assignment*), l'affectation de ces conteneurs à des emplacements de stockage (*storage assignment* ou *slotting*) et de l'ordonnancement des opérations de manutention (dépose/récupération) des conteneurs (*interleaving*). Dans notre étude, nous traiterons le problème d'affectation des conteneurs à des emplacements de stockage : *Storage Allocation Assignment Problem* (SLAP), souvent traité dans la littérature par des méthodes relevant de la recherche opérationnelle [9, 8, 21].

Comme ceci a été récemment souligné par [2], l'apprentissage automatique offre de multiples perspectives pour les applications industrielles (e.g. détection de pannes de composants, diagnostic de pannes, optimisation de production, détection de dommages). Néanmoins ce type d'approche est encore sous-exploité [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*).

L'objectif de cet article est, pour notre contexte applicatif et ses contraintes propres, d'identifier les points sur lesquels l'apprentissage automatique pourrait s'avérer pertinent. Notre contribution consiste donc en la proposition d'une stratégie de gestion d'entrepôt permettant de traiter la problématique d'affectation des positions de stockage de manière dynamique, en s'appuyant sur les perspectives offertes par l'apprentissage automatique, combiné à la recherche opérationnelle. Cette proposition sera contextualisée en s'inspirant d'exemples d'applications connexes s'intéressant à l'apprentissage automatique.

La section 2 est dédiée à la présentation détaillée de la problématique et de l'état de l'art, en soulignant des dif-

ficultés encore peu traitées, liées au manque d'anticipation des commandes. La section 3 se focalise sur les méthodes de résolution du domaine, basées sur la recherche opérationnelle et sur l'apprentissage automatique. Enfin dans la section 4, nous proposons également des nouvelles pistes d'investigations.

2 Problématique et état de l'art

2.1 Description du système

Nous nous plaçons dans le contexte d'un entrepôt automatisé *goods-to-person* mettant en œuvre un transtockeur de type *Shuttle-based Storage and Retrieval Systems* (SBS/RS), qui est un dérivé du *Automated Storage and Retrieval System* (AS/RS) utilisant des navettes (*Shuttle*). Le transtockeur est composé d'une ou plusieurs allées. Au centre des rayonnages des allées, une navette évolue à chaque étage (*tier-captive shuttle*). Les navettes sont responsables du transport des conteneurs de stockage depuis et vers l'ascenseur (le point I/O) ainsi que de leur maintenance dans les rayonnages (dépose et récupération). La figure 1 présente une vue de dessus d'un entrepôt. Le SBS/RS est configuré avec 4 allées, dont les rayonnages sont organisés en simple profondeur (*single deep*). Dans les rayonnages sont stockés les conteneurs de stockage contenant les unités de gestion du stock (*Stock Keeping Unit* ou SKU).

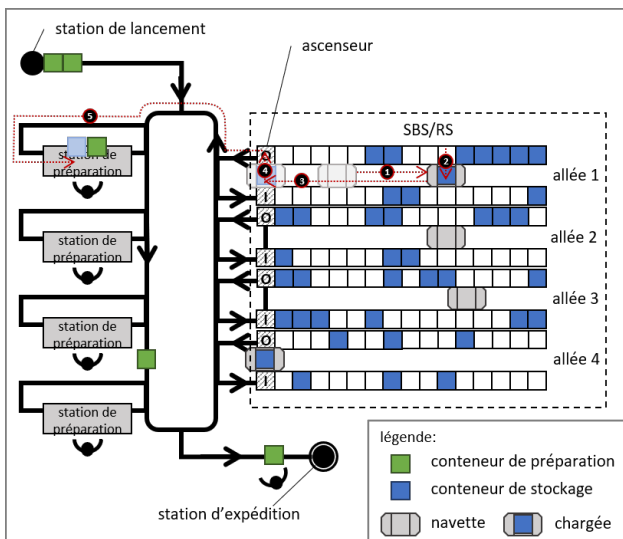


FIGURE 1 – Schéma d'un entrepôt avec un SBS/RS

La figure 1 illustre l'environnement associé à notre problématique. Dans un entrepôt logistique, plusieurs opérations sont nécessaires pour assurer la préparation des commandes :

1. *Goods-in* : étape d'entrée des produits dans le système. Au préalable, les SKU doivent être disponibles pour la préparation, i.e. stockés dans le SBS/RS.
2. *Picking* : étape de préparation. Les conteneurs de préparation sont acheminés depuis une station de lancement vers les stations de préparation, et les

conteneurs de stockage (contenant le stock) sont acheminés vers les mêmes stations de préparation (étapes 1,2,3 et 4 de la figure 1). Une fois les conteneurs de préparation et de stockage arrivés en station de préparation, les préparateurs peuvent alors transférer les produits des conteneurs de stockage vers les conteneurs de préparation.

3. *Shipping* : étape d'expédition. Le *picking* terminé, le conteneur de préparation est acheminé vers l'expédition et le conteneur de stockage retourne vers le SBS/RS pour être de nouveau stocké.
4. *Slotting* : étape d'affectation. Le système de gestion d'entrepôt *Warehouse Management System* (WMS) va sélectionner la meilleure position de stockage pour le conteneur de stockage, en fonction des caractéristiques des SKU présents dans le conteneur, de l'historique des commandes et de la stratégie de stockage implémentée.

2.2 Optimisation d'un entrepôt automatisé

Dans l'optique d'obtenir ou conserver un avantage concurrentiel, les industriels s'efforcent d'améliorer les performances de leurs entrepôts par une meilleure conception et l'optimisation de la conduite des opérations. L'optimisation d'un entrepôt automatisé peut être définie par plusieurs objectifs concomitants à minimiser : la distance moyenne de trajet d'un circuit de préparation, les coûts d'investissements et d'exploitation, le temps de préparation d'une commande ou d'un groupe de commandes, l'utilisation des équipements et des personnes, ou la perte d'espace [12]. Atteindre ces objectifs requière de prendre des décisions, que l'on peut hiérarchiser en niveaux stratégique, tactique et opérationnel [22]. Ces décisions portent sur différents points : conception des plans de l'entrepôt et dimensionnement (*layout design*), affectation des produits à des emplacements de stockage (*storage assignment*), affectation des commandes à des tournées et regroupement des allées en zone de travail (*batching and zoning*), routage des préparateurs de commande (*routing*), triage et accumulation des commandes (*order sorting and accumulation*) [12]. Comme ceci est illustré par la figure 2, les opérations de préparation de commande ont été identifiées [12] comme étant l'activité la plus coûteuse en temps et nécessitant une main d'œuvre conséquente.

55% des coûts opérationnels sont imputables à la préparation des commandes [27]. Le temps de préparation est lui-même impacté par (i) la durée de trajet (*Travel*) des SKU vers, depuis et entre les positions de préparations (à hauteur de 50%), (ii) la durée d'identification (*Search*) des SKU nécessaires à la préparation des commandes (à hauteur de 20%), (iii) la durée de prélèvement (*Picking*) des SKU depuis les conteneurs de stockage (à hauteur de 15%), (iv) la durée de préparation (*Setup*) de la tournée (à hauteur de 10%) et (v) d'autres tâches (*Other*) (à hauteur de 5%) [23], comme illustré dans la figure 2. Pour cette raison, nous nous focalisons sur la réduction du temps de préparation de commandes, et plus particulièrement sur la minimisation du temps (ou distance) de trajet des navettes du

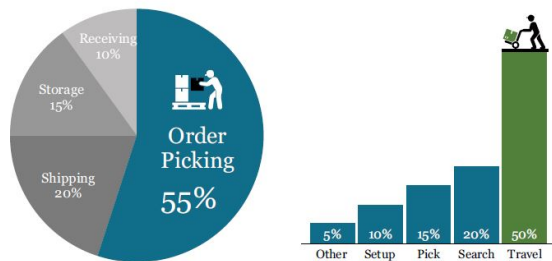


FIGURE 2 – Répartition des coûts opérations logistiques cumulées sur une année dans un entrepôt (gauche) et répartition du temps de chaque tâche sur l'activité de préparation de commandes (*order picking* - droite) (extrait de [11])

SBS/RS. Dans cette optique, nous allons étudier l'affectation des produits à des emplacements de stockage optimaux pour réduire le coût des opérations de préparation.

L'affectation de produits à des positions de stockage est connue dans la littérature comme étant le *Storage Location Assignment problem* (SLAP), dont la première formulation a été introduite par [9], et a été prouvé NP-hard par [5]. Le SLAP peut être abordé de différentes manières [8] selon la nature des données d'entrées : dates d'entrée et de sortie du produit connues dans le système, caractéristiques des produits (poids, dimensions), etc. Lors de l'affectation des conteneurs (et donc des produits qu'ils contiennent) à des emplacements de stockage (*Slotting*), le WMS s'efforce de choisir des emplacements en respectant une stratégie de stockage qui permet d'obtenir de bonnes performances lors du *Picking* tout en respectant des règles métiers et opérationnelles :

- la masse globale des produits doit être répartie sur toute la longueur du rayonnage pour limiter les contraintes mécaniques ;
- des espaces vides doivent être laissés pour faciliter les opérations de manutention intra-allées ;
- les produits comburants et combustibles ne doivent pas être stockés les uns à côté des autres ;
- le stock doit être distribué entre les allées et les étages pour ne pas bloquer le stock en cas d'indisponibilité d'un appareil (ascenseur ou navette).

Ces règles sont autant de contraintes à prendre en compte pour résoudre le SLAP. Mais, elles sont peu considérées dans les formalisations des problèmes, notamment la répartition de la masse, comme souligné dans l'état de l'art de [21]. Pour résoudre le SLAP, des méthodes exactes (programmation dynamique, programmation linéaire mixte en nombre entiers) et des méthodes approchées (heuristique et métaheuristiques) ont été proposées [26]. Plusieurs stratégies de stockage peuvent être envisagées, induisant des règles et une formulation particulières pour traiter le problème de SLAP [11]. Les stratégies les plus courantes sont :

- *Class-Based* (CB) : les produits sont séparés en plusieurs classes et une zone de stockage est affectée à chaque classe. La plus populaire est l'affectation

en 3 classes, ABC où les SKU sont répartis suivant un critère tel que le *turnover* (e.g. les produits les plus fréquemment commandés sont placés dans la classe A, associée à la zone la plus proche du point I/O), introduite par [9]. À l'intérieur de chaque classe, les produits sont rangés avec une règle simple (e.g. position disponible la plus proche du point I/O). D'autres classifications peuvent être envisagées. Par exemple, la classification XYZ (détaillée dans la section 2.3) s'appuie sur les fluctuations de commandes d'un produit [19, 24] ;

- *Duration-of-Stay* (DoS) : les produits sont affectés à un emplacement dont la distance depuis le point I/O est proportionnelle à la durée de séjour dans l'entrepôt. Sous certaines conditions (DoS connu à l'avance et nombre équilibré d'entrées/sorties de marchandises), cette règle est optimale [6] (par rapport aux autres stratégies, en terme de temps de trajet et d'occupation d'espace) ;
- *Cube-Order-per-Index* (COI) : cette métrique définit le ratio entre l'espace nécessaire pour stocker un produit et sa demande [6]. Les produits avec une faible COI (i.e. de petite taille et souvent demandé) sont placés près du point I/O ;
- *Random-Based* : cette politique affecte les produits à des emplacements de manière aléatoire. Elle peut être utilisée dans l'industrie et est considérée dans la littérature comme base de comparaison.

2.3 Problème de la fluctuation des commandes

Dans notre contexte opérationnel et industriel, au-delà des contraintes métiers présentées précédemment (répartition de la masse, espaces vides, compatibilité entre produits et distribution spatiale.), des complications supplémentaires ont été détectées, lors de l'exploitation des algorithmes de *Slotting* (implémentés dans les WMS). Ces complications résultent d'aspects qui sont souvent ignorés :

1. la fluctuation de la fréquence de demande et de la quantité des SKU dans les commandes ;
2. la survenue d'événements rares mais significatifs [1]. Par exemple, il peut s'agir de promotions conduisant à un pic de commande soudain et bref.

Ce dernier point est illustré par la figure 3 qui représente un historique des commandes observées par l'entreprise Knapp, dans un entrepôt donné et sur une période spécifique. La première ordonnée (à gauche) représente la quantité de pièces commandées en fonction de la semaine sur une période d'environ 6 mois. La seconde ordonnée (à droite) représente la performance de préparation, exprimée (par convention) en nombre de prélèvements par heure (*pick/h* - moyenne sur une semaine). Dans notre exemple, on constate une baisse de la performance de préparation, d'environ 30% en semaine 10. Cette chute de la performance pourrait s'expliquer par une réduction de la quantité de travail (nombre pièces commandées) en semaine 10, ce qui impliquerait une chute de la performance de prépa-

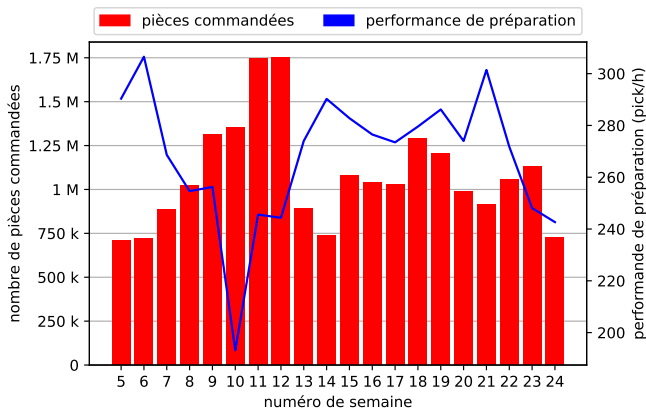


FIGURE 3 – Nombre de pièces commandées et performance de *picking* en fonction de la semaine : un exemple.

ration. Mais, on constate que la quantité totale de SKU à préparer augmente de +3% entre les semaines 9 et 10. Une variation aussi faible du nombre de pièces commandées ne devrait pas impacter autant la performance. La chute de performance résulte d'un pic de commande sur un type particulier de pièce, détaillé ci-après.

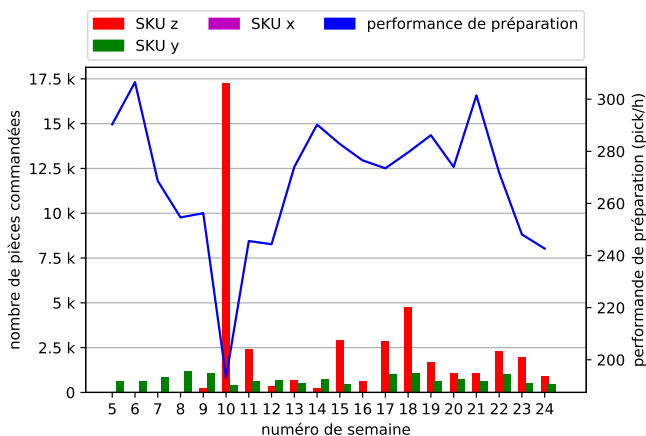


FIGURE 4 – Nombre de pièces commandées pour des SKU étiquetés X, Y et Z et la performance de *picking* en fonction de la semaine.

La figure 4 représente le nombre de pièces commandées pour 3 SKU en fonction de la semaine et la seconde ordonnée représente la performance de préparation. Les 3 SKU choisis sont représentatifs des types variations de demande observées dans les entrepôts. Le produit étiqueté x a une demande faible voire négligeable (moins de 0,005% de la demande totale), avec une faible variation dans le temps. Le produit étiqueté y se place à un niveau de demande plus important (près de 0,12%) et on observe une légère variation de quantité dans le temps. Enfin, le produit étiqueté z a une variation de demande brutale. On constate que ce produit, jamais commandé avant la semaine 10, devient en semaine 10 un produit très demandé (17 000 pièces sur un total de 1 352 273 requises soit près de 1,28% de la

mande totale) et ce pour une durée de seulement 1 semaine. Ensuite sa demande devient sporadique, mais reste significative (moins de 0,37% de la demande).

Ces comportements de variation de demande sont identifiés dans la littérature grâce à la classification par analyse XYZ [19, 24]. La classe X regroupe les produits dont la variation de demande (dans le temps) est stable, uniforme, et continue avec de faible variation. La classe Y regroupe les produits dont la demande fluctue davantage, notamment avec la saison. Enfin, la classe Z regroupe les produits dont la demande est sporadique, irrégulière voir anormale en terme de quantité par rapport aux autres produits. Les variations demande des produits Z ne sont peu ou pas prédictibles.

Cette classification des SKU s'applique à notre exemple, le produit étiqueté $x \in X$, celui étiqueté $y \in Y$ et étiqueté $z \in Z$. Le produit étiqueté z , a subi une variation soudaine et brève du nombre de produits demandés. Il a donc été identifié par le WMS comme un produit à forte rotation. Par conséquent ce SKU s'est vu assigner les emplacements de stockage au plus près du point I/O. Le fonctionnement du WMS est parfaitement normal, sur la base des données dont il dispose (historique des commandes, variation de demande passé des SKU) et de la stratégie de stockage utilisé. L'algorithme de *slotting* a assigné un produit à forte demande sur les emplacements de stockage dont l'accès est le plus "rapide". Le problème est néanmoins le suivant : cette explosion de commandes pour la référence z est valide pour une courte et brève période. Les semaines suivantes ne requièrent que peu de cette référence de produit, qui se retrouve dans le dernier quartile des produits demandés. Toutefois, le SKU étiqueté Z occupe les "meilleurs" emplacements de stockage, ce qui implique des déplacements plus long pour les navettes, qui doivent dépasser cette zone de stockage, pour atteindre les autres SKU.

Le problème illustré par cet exemple est : comment exploiter l'historique de commandes pour positionner les SKU dans l'entrepôt de manière optimale ? À noter que cette question pose le problème sous-jacent suivant : faut-il recalculer les affectations de positions pour chaque SKU et donc modifier dynamiquement l'agencement de l'entrepôt ? Ceci impliquerait des mouvements de manutention intra-allées pour repositionner les conteneurs et du temps d'indisponibilité du stock et des navettes (occupées à des tâches sans valeur ajoutée immédiate). Ceci sous-entend de définir une politique de gestion de l'entrepôt permettant d'effectuer cette manutention sans perturber la production. À notre connaissance, cet aspect n'est que trop peu abordé dans la littérature. On parle de *Dynamic Storage Location Assignment Problem (DSLAP)*. Ce problème a été abordé par des méthodes heuristiques [8]. Concernant l'exploitation de l'historique des commandes en particulier, et dans le cas d'une stratégie CB (ABC ou XYZ), l'industrie s'appuie essentiellement sur une analyse superficielle des données (moyenne des commandes passées pour ABC ou leur fluctuation moyenne pour XYZ), sans chercher à modifier l'agencement de l'entrepôt dynamiquement.

La section suivante se focalise sur une modélisation plus précise de cette problématique et des liens possibles entre

la recherche opérationnelle et l'apprentissage automatique.

3 Modélisation, recherche opérationnelle et apprentissage automatique

Compte tenu de la problématique décrite précédemment, incluant la fluctuation des commandes, nous proposons ici des pistes concernant la modélisation de ce problème d'optimisation et son couplage avec l'apprentissage automatique, en s'appuyant sur les travaux récemment publiés sur ce sujet. La section 3.1 est dédiée à la formulation du problème d'optimisation du SLAP. La section 3.2 concerne l'usage et les perspectives de l'apprentissage automatique dans notre contexte. Nous nous intéresserons en particulier à l'apprentissage non supervisé, l'apprentissage supervisé et l'apprentissage par renforcement, en soulignant les liens avec l'apprentissage par réseaux de neurones profonds.

3.1 Modélisation du problème d'optimisation

Dans cette partie, nous nous intéressons à une formulation de notre problème d'optimisation. Les formulations du SLAP sont basées sur le problème *General Assignment Problem* (GAP) [10].

Les entités souvent rencontrées dans la littérature sont les suivantes. Soit $\mathcal{N} = \{1, \dots, N\}$ l'ensemble des SKU (éléments) à stocker. Soit $\mathcal{L} = \{1, \dots, L\}$ l'ensemble des positions. Soit $\mathcal{O} = \{1, \dots, O\}$ l'ensemble des commandes. Nous définissons une variable binaire x_{ik} valant 1 si l'article de type $i \in \mathcal{N}$ est assigné à la position $k \in \mathcal{L}$, 0 sinon. Dans chacun des rayonnages des allées, la distance entre une position de stockage k et le point d'entrée/sortie I/O, situé au niveau de l'ascenseur, est notée $d_k \geq 0, k \in \mathcal{L}$. La séquence de sortie des SKU et la quantité requise sont données par l'ensemble des commandes de préparation $\mathcal{O} = \{1, \dots, O\}$. Cet ensemble permet de déterminer la probabilité de demande de chaque SKU i , notée $p_i \geq 0, i \in \mathcal{N}$. À noter que cette probabilité repose généralement sur une mesure simpliste (nombre moyen de commandes par article), et ne permet pas de détecter les pics si l'intervalle d'estimation est trop important.

Une liste de propriétés peut être ajoutée à chaque article appartenant à l'ensemble \mathcal{N} : classes d'éléments compatibles (e.g. problématique des produits inflammables), poids, dimensions, etc.

Il existe plusieurs manières de formuler le problème, selon la modélisation de l'entrepôt et le (ou les) critère(s) que l'on cherche à optimiser (e.g. voir les récents travaux intégrant une optimisation multi-objectif [28, 4]). Nous donnons ici un exemple simplifié, inspiré de [29] dans le cas où le critère d'optimisation est de minimiser la distance total de trajet des navettes et la répartition du poids. Pour cela, nous définissons w_i , correspondant à la masse du SKU i , et l'entité $\mathcal{M} = \{1, \dots, M\}$ l'ensemble des colonnes (voir la figure 5). Une colonne m_i est une partition de l'ensemble \mathcal{L} .

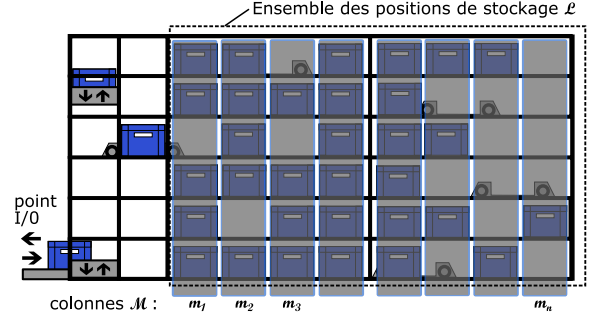


FIGURE 5 – Schéma explicatif des colonnes du SBS/RS considérées pour la contrainte de répartition de masse

La fonction objectif à optimiser peut être formulée par z :

$$z = \min \sum_{i \in \mathcal{N}} \sum_{k \in \mathcal{L}} x_{ik} \cdot d_k \cdot p_i \quad (1)$$

$$\sum_{i \in \mathcal{N}} x_{ik} \leq 1 \quad \forall k \in \mathcal{L} \quad (2)$$

$$\sum_{k \in \mathcal{L}} x_{ik} = 1 \quad \forall i \in \mathcal{N} \quad (3)$$

$$\left| \sum_{i|k \in m_1} w_i x_{ik} - \sum_{i|k \in m_2} w_i x_{ik} \right| \leq \varepsilon \quad \forall (m_1, m_2) \in M^2 \quad (4)$$

$$x_{ik} \in \{0, 1\} \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{L}$$

Les contraintes (2) imposent un seul produit pour chaque position, les contraintes (3) imposent une seule position pour chaque produit. Les contraintes (4) imposent de répartir la masse globale des SKU sur la totalité de la longueur du rayonnage (ε étant un paramètre associé aux écarts de poids entre colonnes). On pourrait étendre ces contraintes en s'appuyant sur d'autres propriétés associées aux SKU, par exemple le problème de compatibilité entre les produits (produits inflammables).

Cette formulation est assez similaire à la stratégie DoS (où notre p_i peut être associé au temps de présence dans le système), en lien avec de l'apprentissage profond [15].

Dans le cas d'une stratégie de type *Class-Based*, on pourrait classifier les produits en catégorie et adapter la formulation en intégrant une nouvelle entité $\mathcal{C} = \{1, \dots, C\}$, désignant l'ensemble des classes de stockage définies : les produits sont alors pré-affectés à des zones, induisant des contraintes supplémentaires sur les positions d'affectation (variables x_{ik}).

3.2 Apprentissage automatique

Comme souligné dans l'introduction, l'apprentissage automatique et en particulier l'apprentissage profond ont montré leur efficacité dans de nombreux domaines [7]. Cela fait notamment partie des perspectives récemment formulées concernant l'usage de l'apprentissage profond pour la prédiction des commandes et la catégorisation des produits [31, 21].

Plusieurs approches peuvent être envisagées pour exploiter l'historique des commandes, en s'appuyant sur l'apprentissage automatique. Nous allons considérer ci-après trois approches suscitant un intérêt récent de la communauté scientifique : l'apprentissage non-supervisé, supervisé et l'apprentissage par renforcement. Dans chaque cas, nous donnons quelques récents travaux, en soulignant les éventuels liens avec l'apprentissage profond, autour de notre problématique et autour de problématiques connexes, utilisant ces approches, soulignant ainsi leur potentiel pour traiter notre sujet.

3.2.1 Apprentissage non supervisé

L'apprentissage non supervisé permet, sur la base des caractéristiques des objets manipulés ou de leurs interactions, d'identifier des groupes d'objets présentant des caractéristiques similaires.

Dans notre cas, l'apprentissage non supervisé se retrouve sur le SLAP pour classer les SKU en catégories, lorsque la stratégie de stockage est de type *Class-Based*. Les SKU sont répartis dans des classes \mathcal{C} , mentionnées dans la section précédente, grâce à l'apprentissage automatique [30]. Les classes de produits peuvent être établies par des techniques de *clustering* s'appuyant sur les attributs des différents SKU commandés et l'historique des commandes. Dans l'étude du SLAP d'un entrepôt de stockage par *clustering* [30] dans le e-commerce, se base sur des attributs standard de la classification *Class-Based*, tels que le prix et le volumes des ventes, auxquels il associe des attributs complémentaires : le taux de commentaires favorables des SKU, le type de paiement autorisé à la livraison (paiement en espèce ou non), si les frais de port sont offerts ou non. Il a pu ainsi guider le processus de *clustering*, et répartir 109 SKU en 3 catégories.

Cette méthode de *clustering* est basée sur les attributs intrinsèques des SKU, d'autres méthodes de classification se basent sur les interactions entre les SKU. Des règles d'associations *association rule mining* permettent d'extraire des relations d'intérêts entre les SKU. Ces règles fournissent alors des métriques (e.g *weighted support count* (WSC) [18], *product affinity* [14]), qui sont utilisées pour catégoriser les SKU et les assigner aux positions de stockage.

Les méthodes de classification étudiées font peu appel aux réseaux de neurones. Or les méthodes de classification par apprentissage profond font parties des perspectives récemment formulées pour la catégorisation des produits [31, 21].

3.2.2 Apprentissage supervisé

L'apprentissage supervisé a pour objectif la prédiction et/ou la classification, sur la base d'un entraînement réalisé sur des données annotées.

L'apprentissage supervisé peut également se retrouver dans la problématique du SLAP, par exemple dans le cas d'une stratégie de type *Duration of Stay* [15]. Ces travaux introduisent le terme \hat{p} désignant la durée de séjour (prédite) d'un SKU dans le système, le *Duration of Stay* (DoS), qui est estimé avec un réseau de neurones profond convolutif

(*convolutional neural networks* ou CNN) et un réseau de neurones récurrent (*recurrent neural networks* ou RNN) de type *long short-term memory* (LSTM) [3]. Ce réseau est entraîné sur un historique de commandes, correspondant à l'entité \mathcal{O} introduite précédemment. Pour un conteneur avec un DoS prédit par apprentissage automatique, noté \hat{p} , la fonction d'affectation des positions de stockage $A : \mathbb{R} \rightarrow \mathcal{W}$ est, selon [15] :

$$A(\hat{p}) = \arg \min_{w \in \tilde{\mathcal{W}}} d(N\hat{r}\hat{W}(\hat{p}), w) + c(w) \quad (5)$$

Avec $d(v, w)$ la distance entre deux positions v et w . $c(w)$ représente les coûts connexes associés au stockage à cette position w . Cela prend en compte les coûts des contraintes spécifiques de chaque entrepôt, (e.g : *stock-mixing*, *stock-splitting*, répartition de la masse,...). La position de stockage optimale approchée dans le cas du DoS est donnée par $N\hat{r}\hat{W}(\hat{p})$. $\tilde{\mathcal{W}}$ représente l'ensemble des positions de stockage disponibles lorsqu'un conteneur de stockage entre dans l'entrepôt. $d(N\hat{r}\hat{W}(\hat{p}), w)$ représente le coût de non stockage d'une palette à la position de stockage optimale pour le DoS. La position de stockage $A(\hat{p})$ est alors choisie en fonction de ces deux coûts (les coûts connexes $c(w)$ et le coût de non stockage $d(N\hat{r}\hat{W}(\hat{p}), w)$) [15].

Le principe de cette approche est illustrée par la figure 6, dans le cas présent (DoS) et dans la perspective d'une utilisation dans le cas d'une stratégie de type *class-based*. On peut noter ici que cette approche combine apprentissage profond et optimisation sous contraintes.

À notre connaissance, l'article de [15] constitue le premier article qui traite du SLAP avec des réseaux de neurones profonds, en prenant en compte le caractère dynamique de l'entrepôt et les incertitudes d'exploitation. À noter que ces travaux sont associés à la mise à disposition de l'une des seules base de données publiques contenant des données d'entrepôt pour traiter le problème d'affectation.

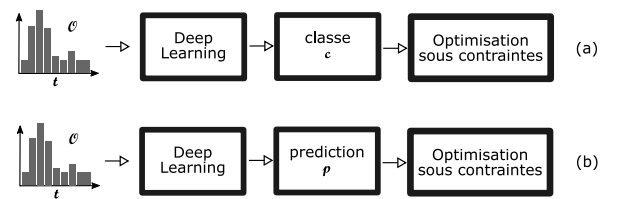


FIGURE 6 – Apprentissage profond et Optimisation par contraintes. Variante (a) les réseaux de neurones retournent la classe de stockage c pour les SKU. Variante (b) les réseaux de neurones prédisent la valeur \hat{p} du DoS pour les SKU

La figure 6 présente deux axes de recherche envisagés pour résoudre le SLAP. La variante (a) représente l'utilisation de *Deep Learning* pour catégoriser les produits en classes c_i en se basant sur l'historique des commandes, inspiré de [30, 14, 18]. La variante (b), correspond aux travaux de [15], utilise du *Deep Learning* pour prédire la valeur d'une métrique (le DoS pour [15], la probabilité de demande des

SKU dans notre cas). Dans les deux variantes (a) et (b), des techniques de *Deep Learning* vont transformer/utiliser les données, ici l'historique des commandes \mathcal{O} , pour fournir des variables d'entrée pour résoudre notre problème d'optimisation sous contrainte.

3.2.3 L'apprentissage par renforcement : prendre en compte le long terme

L'apprentissage par renforcement [25] est une autre piste intéressante n'ayant, à notre connaissance, pas encore été étudiée dans notre contexte.

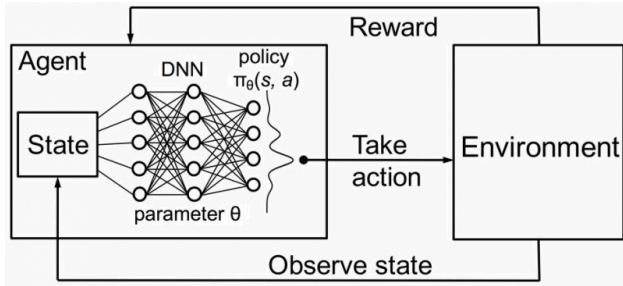


FIGURE 7 – *Deep reinforcement learning* (extrait de [17])

Le principe est illustré par la figure 7. Néanmoins, cette classe d'approche fait actuellement l'objet de travaux pour des problèmes connexes. On peut citer la répartition en charge de l'information [16] (apprentissage pour l'affectation temps réel des tâches sur des serveurs distribués), la replanification de tâches (i.e. des commandes passées par les clients) dans les environnements de production [20], la gestion des ressources en énergie en fonction des besoins (avec incertitude sur la demande) [13]. Elle repose sur une observation de l'environnement afin d'identifier les actions à entreprendre afin de maximiser les récompenses sur moyen ou long terme. s correspond à l'état observé et a à l'action optimale à entreprendre, sur la figure 7 l'action optimale est donnée par $\underset{a}{\operatorname{argmax}}(\pi_{\theta}(s, a))$ (θ est déterminé par la phase d'entraînement du réseau de neurones). Pour notre étude, une difficulté est de correctement modéliser le système et de définir les rôles : environnement, agent, action (e.g. réagencement de l'entrepôt dynamiquement) et récompenses (e.g. minimiser, sur un ensemble de commandes, le temps ou la distance de trajet des navettes de l'SBS/RS pour récupérer des SKU).

Dans notre cas, les différents rôles pourraient être attribués de la manière suivante :

- l'**environnement** est constitué du SBS/RS ;
- l'**agent** engage l'action optimale, apprise par renforcement ;
- l'**action** consiste à stocker le SKU $i \in N$ à la position $k \in L$;
- l'**état** du système est constitué de l'état d'occupation des positions de stockage \mathcal{L} , l'ensemble des commandes courantes, l'ensemble des SKU à stocker \mathcal{N} et de l'ensemble des classes de stockage \mathcal{C} dans le cas d'une stratégie *class-based* ;
- la **récompense** (*reward*) est associée au bénéfice

obtenu suite à l'action entreprise (associée à la baisse de la distance moyenne des trajets des navettes par exemple).

4 Conclusion

Nous avons vu que l'optimisation d'un entrepôt automatisé utilisant un SBS/RS, passe par la réduction du temps de préparation et donc par la réduction du temps/distance de trajet des navettes. Améliorer l'affectation des positions de stockage (*Slotting*) est un moyen d'action pour minimiser les trajets des navettes, connu dans la littérature comme le *Storage Location Assignment Problem* (SLAP). Nous avons vu, à travers un exemple concret, certaines difficultés associées à la fluctuation, difficilement prévisible, des commandes. Ceci requiert de tendre vers un système d'allocation dynamique basé sur l'apprentissage de la dynamique des commandes passées.

Nous avons ensuite présenté les approches considérées pour aborder cette problématique, généralement formulée comme un problème d'optimisation sous contraintes, relevant de la recherche opérationnelle. Nous avons également fait l'état de l'art des travaux connexes impliquant de l'apprentissage automatique. Cette étude des récents travaux publiés dans la littérature montre le potentiel de l'apprentissage automatique pour ce problème, et en particulier de l'apprentissage supervisé par réseaux de neurones profonds et également de l'apprentissage par renforcement. Cette étude montre également que peu de travaux ont été réalisés sur l'usage de l'apprentissage profond dans ce contexte. Enfin, ce travail de synthèse illustre comment l'apprentissage profond peut-être avantageusement combiné à la modélisation du problème, posé sous forme d'un problème d'optimisation de contraintes : l'apprentissage peut permettre de modéliser les commandes afin de guider le processus d'optimisation. La suite à moyen terme de ces travaux concernera l'approfondissement de cette étude et la réalisation d'un simulateur avant l'évaluation sur un environnement réel.

Références

- [1] Kamil Ciosek and Shimon Whiteson. Off-Environment RL with Rare Events. *NIPS workshop on Optimizing the Optimizers*, 2016.
- [2] Alberto Diez-Olivan, Javier Del Ser, Diego Galar, and Basilio Sierra. Data fusion and machine learning for industrial prognosis : Trends and perspectives towards industry 4.0. *Information Fusion*, 50 :92 – 111, 2019.
- [3] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description, 2014.
- [4] Marcele Elisa Fontana and Vilmar Santos Nepomuceno. Multi-criteria approach for products classification and their storage location assignment. *The Inter-*

national Journal of Advanced Manufacturing Technology, 88(9) :3205–3216, Feb 2017.

- [5] Edward H. Frazelle. *Stock location assignment and order picking productivity*. PhD thesis, Georgia Institute of Technology, 1989.
- [6] Marc Goetschalckx and H. Donald Ratliff. Shared storage policies based on the duration stay of unit loads. *Management Science*, 36(9) :1120–1132, 1990.
- [7] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016.
- [8] Jinxiang Gu, Marc Goetschalckx, and Leon F. McGinnis. Research on warehouse operation : A comprehensive review. *European Journal of Operational Research*, 177(1) :1–21, 2007.
- [9] Warren Hausman, Leroy Schwarz, and Stephen Graves. Optimal Storage Assignment in Automatic Warehousing Systems. *Management Science*, 22(6), 1976.
- [10] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, Berlin, Germany, 2004.
- [11] Monika Kofler. *Optimising the Storage Location Assignment Problem Under Dynamic Conditions*. PhD thesis, Johannes Kepler Universität Linz, 2014.
- [12] René de Koster, Tho Le-Duc, and Kees Jan Roodbergen. Design and control of warehouse order picking : A literature review. *European Journal of Operational Research*, 182(2) :481 – 501, 2007.
- [13] Jesus Lago, Ecem Sogancioglu, Gowri Suryanarayana, Fjo De Ridder, and Bart De Schutter. Building day-ahead bidding functions for seasonal storage systems : A reinforcement learning approach. *IFAC PapersOnLine*, 52 :488–493, 06 2019.
- [14] Jiayi Li, Mohsen Moghaddam, and Shimon Y. Nof. Dynamic storage assignment with product affinity and abc classification—a case study. *The International Journal of Advanced Manufacturing Technology*, 84(9) :2179–2194, Jun 2016.
- [15] Michael Lingzhi Li, Elliott Wolf, and Daniel Wintz. Duration-of-stay storage assignment under uncertainty, 2019.
- [16] Long Mai, Nhu-Ngoc Dao, and Minho Park. Real-time task assignment approach leveraging reinforcement learning with evolution strategies for long-term latency minimization in fog computing. *Sensors*, 18(9), 2018.
- [17] Hongzi Mao, Mohammad Alizadeh, Ishai Menache, and Srikanth Kandula. Resource management with deep reinforcement learning. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks - HotNets '16*, pages 50–56. ACM Press, 2016.
- [18] David Ming-Huang Chiang, Chia-Ping Lin, and Mu-Chen Chen. Data mining based storage assignment heuristics for travel distance reduction. *Expert Systems*, 31(1) :81–90, 2014.
- [19] Irena Nowotyńska. An application of XYZ analysis in company stock management. *Modern Management Review*, 2013.
- [20] Jorge A. Palombarini and Ernesto C. Martínez. Closed-loop rescheduling using deep reinforcement learning. *IFAC-PapersOnLine*, 52(1) :231 – 236, 2019.
- [21] J. Reyes, E. Solano-Charris, and J. Montoya-Torres. The storage location assignment problem : A literature review. *International Journal of Industrial Engineering Computations*, 10 :199–224, 2019.
- [22] B. Rouwenhorst, B. Reuter, V. Stockrahm, G.J. van Houtum, R.J. Mantel, and W.H.M. Zijm. Warehouse design and control : Framework and literature review. *European Journal of Operational Research*, 122(3) :515–533, 2000.
- [23] André Scholz, Sebastian Henn, Meike Stuhlmann, and Gerhard Wäscher. A new mathematical programming formulation for the single-picker routing problem. *European Journal of Operational Research*, 253(1) :68–84, 2016.
- [24] Milan Stojanović and Dušan Regodić. The significance of the integrated multicriteria ABC-XYZ method for the inventory management process. *Acta Polytechnica Hungarica*, 14(5) :20, 2017.
- [25] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*. The MIT Press, second edition, 1998.
- [26] El-Ghazali Talbi. Combining metaheuristics with mathematical programming, constraint programming and machine learning. *Annals of Operations Research*, 240(1) :171–215, 2016.
- [27] James A Tompkins, John A White, Yavuz A. Bozer, and J.M.A Tanchoco. *Facilities planning*. John Wiley & Sons, 4 edition, 2010.
- [28] Wei Wang, Jing Yang, Li Huang, David Proverbs, and Jianbin Wei. Intelligent storage location allocation with multiple objectives for flood control materials. *Water*, 11(8), 2019.
- [29] Ren-Qian Zhang, Meng Wang, and Xing Pan. New model of the storage location assignment problem considering demand correlation pattern. *Computers and Industrial Engineering*, 129 :210 – 219, 2019.
- [30] Li Zhou, Lili Sun, Zhaochan Li, Weipeng Li, Ning Cao, and Russell Higgs. Study on a storage location strategy based on clustering and association algorithms. *Soft Computing*, 2018.
- [31] E. Zunic, H. Hasic, K. Hodzic, S. Delalic, and A. Besirevic. Predictive analysis based approach for optimal warehouse product positioning. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 0950–0954, 2018.