CIGI QUALITA MOSIM 2023

Paramétrisation du DDMRP avec l'apprentissage par renforcement

LOUIS DUHEM[1], MAHA BENALI[1], GUILLAUME MARTIN[3]

[1] Département de mathématiques et génie industriel, Polytechnique Montréal
2500 Chemin de Polytechnique, Montréal, H3T 1J4, Québec, Canada
louis.duhem@polymtl.ca, maha.benali@polymtl.ca

[2] Département de génie industriel, Ecole des mines d'Albi-Carmaux
All. des sciences, Albi, 81000, France
guillaume.martin@mines-albi.fr

***Résumé* – Avec l'augmentation des exigences clients et de la diversité des produits conventionnels, les industriels font face à de nouveaux enjeux de production et de délais. Néanmoins, les méthodes de production ne peuvent plus répondre à ces nouveaux enjeux. Le *Demand Driven Material Requirements Planning* (ou DDMRP) est une méthode de production pilotée par la demande qui s'inscrit dans une nouvelle aire d'innovation industrielle. Cependant, peu d'attention a été accordée à la paramétrisation du DDMRP. Cette étude vise à apporter des ajustements dynamiques à des seuils et horizons de détection de pics afin d'améliorer la paramétrisation de la méthode. La méthodologie proposée est d'intégrer un algorithme d'apprentissage par renforcement au modèle de simulation d'un atelier hybride piloté en DDMRP soumis à des pics de demande. Nous étudions la performance de l'apprentissage, ainsi que l'évolution des indicateurs industriels de l'atelier. Nous parvenons à montrer qu'il est possible de piloter les paramètres de l'atelier tout en améliorant ses performances en termes de satisfaction client et de niveaux d'inventaire. Les résultats de l'étude démontrent la possibilité de piloter les paramètres d'un DDMRP avec une méthode automatique d'apprentissage par renforcement.**

***Abstract* – Considering more demanding customers and the diversity of the conventional products, the industrials face new stakes of production and lead times. Nevertheless, the production methods cannot reach those new goals anymore. The Demand Driven Material Requirements Planning (or DDMRP) is a demand-driven production method, which is included in a new era of industrial innovation. However, little attention has been given to the parametrization of DDMRP. This study aims to dynamically adjust order spike thresholds and horizons to improve the parametrization of the method. The proposed methodology is to integrate a reinforcement learning method to the simulation model of an hybrid DDMRP-run flowshop, subject to a peak demand distribution. We study the performance of the learning process and the industrial indicators of the flowshop. We manage to show that it is possible to adjust the parameters of the flowshop while improving its performance regarding customer satisfaction and inventory levels. The results of the study point out the possibility to drive DDMRP parameters with an automatic method using reinforcement learning.**

***Mots clés* – Gestion de production, Paramétrisation, DDMRP, Apprentissage par renforcement.**
***Keywords* – Production Management, Parametrization, DDMRP, Reinforcement learning.**

1 INTRODUCTION

Les industriels sont aujourd'hui confrontés à un nouveau contexte appelé par Ptak et Smith (2019) le « nouveau normal » en termes d'exigences de production. Cela rend parfois inadaptées des méthodes de production conventionnelles, telles que le *Material Requirements Planning* (MRP), une méthode de gestion de production pilotée par les prévisions de vente. Le nouveau normal se caractérise par une intolérance du client aux longs délais de livraison, une plus grande variété de produits, et également des demandes accrues en termes de sécurité client et de protection environnementale (Ptak, 2019).

Ces enjeux ont laissé place au développement de nouvelles méthodes de production qualifiées de *demand-driven* ou "pilotées par la demande", et plus récemment de la méthodologie DDMRP. Cette méthodologie "pilotée par la demande" est particulièrement intéressante dans un contexte de "marchés volatiles, avec une demande changeante, [...] avec des produits présentant une grande variété" (Azzamouri, 2021). Pour répondre à ces nouveaux besoins, le *Demand Driven Material Requirements Planning* ou DDMRP est apparu. Comme le MRP, le DDMRP est une méthode de gestion de production. Sa finalité est de réduire les stocks et limiter les temps de livraison. Le DDMRP se distingue du MRP comme étant une méthode de gestion de production davantage pilotée par la demande, avec une mise en avant des flux d'informations et de matières. Le DDMRP utilise des points de découplage appelés *buffers* placés tout au long de la chaîne de production et de la nomenclature des produits (Ptak, 2019). L'évaluation du niveau de stock et la projection sur les *buffers* vont permettre l'émission d'ordres d'approvisionnement ou de fabrication, tout en réduisant les stocks.

Un système DDMRP nécessite une paramétrisation pour être efficace, ce qui peut s'avérer difficile à fixer, notamment à cause du besoin de s'adapter à une demande changeante. Il faut par exemple qu'un *buffer* soit capable de détecter un pic de commande capable de venir menacer son intégrité. Ptak et Smith (2019) définissent un pic de commande par un seuil de détection de pics (*Order Spike Threshold* ou OST) et un horizon de détection de pics (*Oder Spike Horizon* ou OSH). Les auteurs proposent des règles générales pour calculer les OST et OSH, mais ces règles ne reposent pas sur des décisions et des algorithmes objectifs. Seuls Damand et al. (2022) ont récemment proposé une méthode pour ajuster l'OST et l'OSH d'un système DDMRP avec un algorithme génétique. Cependant, leur algorithme ne présente pas de composante d'apprentissage et présente une faible flexibilité à une demande inconnue.

Afin d'assurer un ajustement dynamique des paramètres, l'apprentissage par renforcement (R*einforcement Learning*, RL) est devenu de plus en plus populaire dans les problèmes de la chaîne d'approvisionnement (Xanthopolous, 2018). A notre connaissance, seuls Cuartas Murillo et Aguilar (2022) utilisent un agent d'apprentissage pour ajuster les niveaux d'inventaire et les temps de mise en opération d'un système DDMRP. Cependant, rien n'a été proposé pour l'ajustement des paramètres du DDMRP.

Dans ce projet, nous proposons d'utiliser l'apprentissage afin d'ajuster et d'optimiser la paramétrisation du DDMRP. L'objectif est de développer un outil d'apprentissage capable d'ajuster dynamiquement les paramètres du DDMRP et d'améliorer la performance de la méthode. Cet article est organisé de la façon suivante : Section 2 présente un état de l'art de la méthode DDMRP, ainsi que des concepts préliminaires sur les outils d'apprentissage utilisés. Ensuite, Section 3 expose notre cas d'étude ainsi que la méthodologie utilisée. Section 4 analyse les résultats. Tandis que Section 5 propose une conclusion de l'étude.

2 REVUE DE LITTERATURE

Cette section présente une brève revue de littérature du DDMRP et de quelques éléments concernant sa paramétrisation. Nous introduisons par la suite le RL, et exposons la contribution de cet article.

2.1 Littérature du DDMRP

Étant une méthode de production récente, la recherche portant sur le DDMRP est limitée. Azzamouri et al. (2021) ont dressé l'évolution du DDMRP via une revue de littérature et montrent que le DDMRP n'est pas encore bien validée. En particulier, la paramétrisation de cette méthode est largement délaissée. La littérature sur le DDMRP montre que la méthode n'est pas limité à un domaine industriel (Azzamouri, 2021). Bahu et al. (2019) affirment, à travers 30 études de cas, que le DDMRP est adapté dans de nombreux domaines industriels.

La performance du DDMRP a été souvent comparée à celle du MRP (Azzamouri, 2021). Miclo et al. (2015) ont publié une série d'articles portant à démontrer les intérêts du DDMRP face au MRP de façon quantitative. D'autres auteurs se sont intéressés à exposer une comparaison entre le DDMRP et le MRP en milieu industriel, notamment Kortabaria et al. (2018), Shofa et Widyarto (2017) ou Ihme et Stratton (2015). Ces recherches sont basées sur des analyses quantitatives (par exemple, niveaux d'inventaire, temps de mise en œuvre, etc.) et/ou qualitatives (par exemple, réalisation d'entretiens). Ces analyses ont montré l'efficacité du DDMRP en termes de réduction de l'inventaire et des retards.

Peu de littérature du DDMRP porte sur la paramétrisation et les ajustements dynamiques. Ptak et Smith (2019) proposent des équations simplistes pour calculer les paramètres. Or Miclo (2015) souligne que la modification des paramètres du DDMRP affecte la performance du système. Malgré ça, peu d'articles portent sur les paramètres et les ajustements dynamiques (Azzamouri, 2021). Par exemple, Dessevre et al. (2019) proposent un modèle d'ajustement dynamique au niveau des temps de mise en œuvre. Les pistes de recherche concernant l'ajustement des paramètres du DDMRP sont donc nombreuses.

2.2 Paramétrisation du DDMRP

Le DDMRP repose sur un ensemble de règles qui évolue dans un environnement dynamique et sensible à la variabilité de la demande client, permettant de calculer le niveau des zones des *buffers*. Ces *buffers* ont comme but d'absorber le choc induit par la demande, compresser les temps de livraison et gérer les priorités des ordres de commande (Ptak, 2019).

Afin de remplir ces fonctions, les *buffers* en *DDMRP* présentent quatre zones (voir Figure 1) de tailles différentes qui dépendent de différents facteurs propres à l'environnement de production et calculables en tout temps. Ces facteurs sont :

- *Average Daily Usage* (ADU) : la demande moyenne journalière ;
- *Decoupled Lead Time* (DLT) : le temps alloué aux ordres de fabrication pour être finalisés (Martin, 2020) ;
- *Lead Time Factor* (LTF) : calculé à partir de l'ADU et du DLT, ce facteur permet de considérer les incertitudes des délais d'approvisionnement et de production ;
- *Variability Factor* (VF) : permet de prendre en compte les incertitudes sur la demande ;
- *Minimum Order Quantity* (MOQ) : la taille minimale d'une commande.

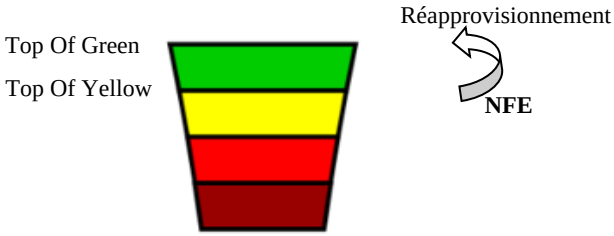

Figure 1 - Différentes zones d'un buffer en DDMRP

Ptak et Smith (2019) proposent de calculer les tailles des 4 zones (verte, jaune, rouge de base, rouge sécuritaire) d'un *buffer* comme suit :

$$\text{Zone verte} = \max(ADU * DLT * LTF, MOQ) \quad (1)$$
$$\text{Zone jaune} = ADU * DLT \quad (2)$$
$$\text{Zone rouge de base} = ADU * DLT \quad (3)$$
$$\text{Zone rouge sécuritaire} = ADU * DLT * LTF * VF \quad (4)$$

Afin de gérer les priorités des ordres de production, Ptak et Smith (2019) proposent de considérer :

Equation du flux net (EFN) = Stock en main + Stock en cours de production - demande qualifiée (5)

L'EFN reflète la position du stock et génère les ordres de réapprovisionnement. Lorsque l'EFN est inférieure à la valeur haute de la zone jaune (TOY), on génère un ordre de réapprovisionnement afin de refaire passer l'EFN à la valeur haute de la zone verte (TOG) (Ptak, 2019).

Dans l'équation de l'EFN, la demande qualifiée est la demande du jour à laquelle on ajoute la demande des pics dans l'horizon de détection (G. Dessevre, Ben Ali, M., 2020). Le calcul de la demande qualifiée proposée par Ptak et Smith (2019) requiert la fixation d'un seuil de détection de pic de demande (*Order Spike Threshold*, OST) et d'un horizon de détection de pic de demande (*Order Spike Horizon*, OSH). L'OST sert à détecter des pics de demande anormaux qui pourraient venir menacer l'intégrité du *buffer*. L'OSH définit la fenêtre de temps sur laquelle on va chercher à détecter les pics de demande (Ptak, 2019). La Figure 1 illustre un exemple d'OST et d'OSH sur une période de 8 jours. Dans la littérature, aucune solution n'est proposée pour ajuster ces deux paramètres.

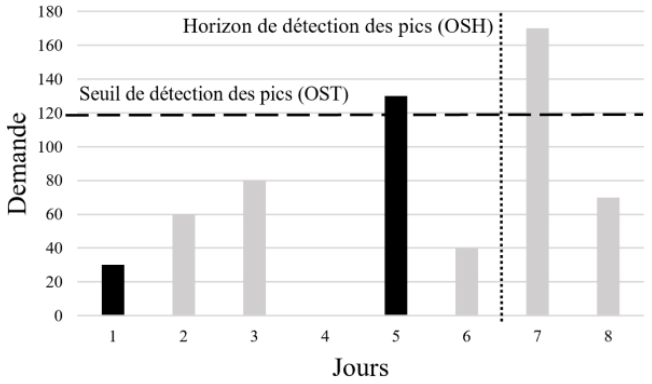

Figure 2. Exemple de calcul d'une demande qualifiée en utilisant l'OST et l'OSH

Ptak et Smith (2019) proposent différentes équations simplistes de calcul de l'OST et de l'OSH, sans en favoriser une par rapport à une autre et sans aucune étude qui justifie le choix de ces équations. À notre connaissance, seuls Damand et al. (2022) ont étudié la paramétrisation des OST et OSH et ont utilisé un algorithme génétique pour fixer les valeurs de OST et OSH sur une période donnée. Cependant, aucun ajustement dynamique n'est réalisé. Notre étude vise à combler ce gap de la littérature et propose un modèle DDMRP dans lequel l'OST et l'OSH vont s'ajuster de façon autonome à l'aide d'une entité d'apprentissage par renforcement.

2.3 Apprentissage par renforcement

L'apprentissage par renforcement (*Reinforcement Learning*, RL) est un type de problème d'apprentissage machine qui vise à apprendre à travers "l'essai et l'erreur" afin de trouver les meilleures solutions (Morales, 2020). L'idée est de prendre des actions et observer les conséquences de ces actions. Concrètement, une entité appelée un agent prend des décisions dans un environnement afin d'atteindre un but précis. Des valeurs numériques appelées des récompenses sont attribuées à ces décisions. Au fur et à mesure, l'agent "apprend" et tend vers de meilleures décisions. Ces décisions sont caractérisées par un changement d'état, un état étant un ensemble de variables décrivant l'environnement.

Les algorithmes de RL reposent sur la répétition de cycles. Un cycle représente le passage d'un état à un nouvel état. Le quadruplet composé de l'état, l'action, la récompense et le nouvel état est appelé l'expérience (Morales, 2020). Le cycle est composé des étapes suivantes :

1. Observation et récompense : l'agent observe l'environnement à travers un espace d'états et lui attribue une valeur numérique à travers une récompense ;
2. Amélioration de la politique π : l'agent améliore la politique afin qu'elle recommande de meilleures actions ;
3. Réalisation de l'action a : l'agent sélectionne une action a dans l'espace d'actions et la préconise à l'environnement ;
4. L'environnement réalise l'action a et effectue son changement d'état s.

Un cycle entier est appelé un pas de temps. La composition d'un pas de temps est représentée par la Figure 2.

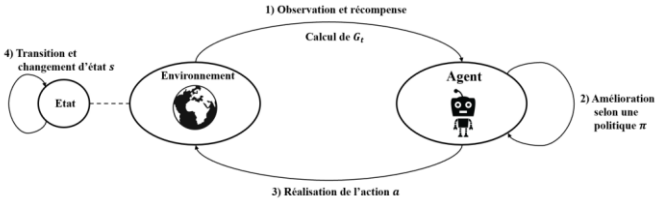

Figure 3. Un cycle d'apprentissage par renforcement

Une tâche est terminée lorsqu'un état terminal est atteint ou lorsqu'on a atteint un nombre maximal de pas de temps. Une séquence de pas de temps, déterminée par un début et une fin de tâche, est appelée un épisode.

Le RL s'appuie principalement sur l'utilisation de la fonction action-valeur, appelée Q-fonction. Cette fonction calcule le retour (i.e., la somme des récompenses collectées dans un seul épisode) attendu si l'agent, considérant une politique π, prend une action a à l'état s. Nous notons le gain G_t, la somme des retours au temps t, c'est-à-dire la somme des récompenses du pas de temps ($t + 1$) au pas de temps final T ($G_t = R_{t+1} + R_{t+2} + \dots + R_T$, avec R_t la récompense attribuée au pas de temps t). La Q-fonction permet de calculer le gain espéré d'une action a prise à un état s, autrement dit la somme des récompenses futures espérée lorsque l'action est prise dans un état particulier. La définition mathématique de la Q-fonction est donnée par l'équation (6) (Morales, 2020).

$$q_\pi(s, a) = E_\pi[G_t \mid S_t = s, A_t = a] \quad (6)$$

Maximiser la Q-fonction revient à trouver l'action qui donne la plus grande somme de futures récompenses espérée, c'est-à-dire l'action qui est la plus bénéfique à l'environnement. Dans la pratique, au lieu de calculer sa valeur exacte, on utilise généralement un approximateur (Morales, 2020). Dans notre cas, on a eu recours à un modèle de réseau de neurones comme approximateur.

Le RL peut être utilisé pour la planification de production en limitant les coûts (Stockheim, 2003) et en étant soumis à certaines contraintes de production (Cao, 2003). Dans plusieurs études telles que celle de Wang et al. (2017), la performance de l'algorithme est comparée à celles

d'heuristiques de planification de la production conventionnelles.

À notre connaissance, un seul article a utilisé le RL pour l'ajustement d'un modèle DDMRP. Cuartas Murillo et Aguilar (2022) proposent un algorithme hybride basé sur l'apprentissage par renforcement pour la gestion de l'inventaire d'un système géré en DDMRP. L'algorithme de RL proposé agit directement sur les niveaux d'inventaire et sur les temps de mise en œuvre. Dans cet article, nous proposons d'utiliser le RL dans un autre objectif, soit d'ajuster les paramètres du DDMRP qui sont habituellement calculés par des formules simplistes recommandées par Ptak et Smith (2019).

2.4 Contribution de l'article

Cet article s'intéresse à un problème rarement adressé dans la littérature, soit la gestion de profils de demande atypiques (demande avec pics). Il propose une approche innovante permettant d'ajuster via l'apprentissage par renforcement des paramètres du DDMRP qui sont habituellement calculés par des formules simplistes recommandées par Ptak et Smith (2019). Bien que le DDMRP repose sur plusieurs paramètres supposés ajustés dynamiquement, peu de littérature s'est intéressée à ce sujet. Seuls Damand et al. (2022) ont proposé un algorithme génétique d'ajustement des OST et OSH, mais en faisant une forte hypothèse que la demande est connue en avance. Or, la méthode DDMRP est apparue pour remédier au problème commun d'inexactitude de prévisions. Nous proposons donc une méthode permettant d'ajuster dynamiquement les OST et OSH sans avoir besoin de connaître la demande en avance.

3 METHODOLOGIE

Cette section présente l'étude de cas et la méthodologie utilisée. Nous évoquons également des éléments d'expérimentation.

3.1 Cas d'étude

Dans ce projet, nous traitons le cas d'un système de production piloté par DDMRP. Il s'agit d'un atelier de type hybride, où un ensemble de *p* produits sont fabriqués par *n* machines parallèles à partir de *m* matières premières. Nous nous basons sur le modèle de simulation à événements discrets développé par Martin (2020) illustré par la Figure 3. L'atelier est dit "hybride" car il fait intervenir des ordres de production en flux poussé et en même temps un flux de production tiré par la demande (Martin, 2020). Les stocks de *m* matières premières et *p* produits finis sont contrôlés par des *buffers* gérés en DDMRP.

La génération de la demande est inspirée du modèle de Dessevre (2020). Un profil de demande est caractérisé par le temps interarrivée de deux commandes. Les produits présentent des profils de demande différents générés de façon aléatoire. Les temps interarrivées sont distribués selon une loi exponentielle de paramètre 2 jours. La taille d'une commande est tirée uniformément à ±20% de la taille moyenne du profil de demande. La taille moyenne du profil de demande est également tirée de façon uniforme entre 50 et 200 produits. À cette demande "régulière" s'ajoutent des pics de demande. L'intervalle de temps entre les pics sont distribués selon une loi exponentielle de moyenne 5 ou 20 jours. La taille des pics de demande est uniformément distribuée entre 2, 3, 4 ou 5 fois la taille d'une demande régulière. Un exemple de profil de demande est illustré par la Figure 4 pour un pic de demande tous les 20 jours.

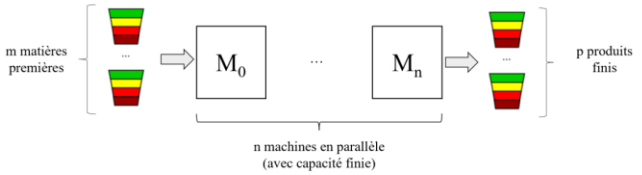

Figure 4. Modèle d'un atelier de production géré en DDMRP de manière hybride

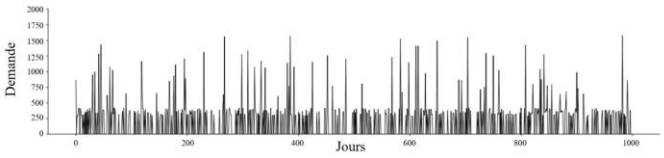

Figure 5. Profil de demande d'un produit avec un pic par 20 jours

Inspiré de Martin (2020), nous considérons trois niveaux de charge de travail (50%, 80% et 95%). Les temps d'opération et de mise en route des machines ont été calculés afin de correspondre avec ces charges de travail. Nous utiliserons ces paramètres afin d'analyser l'effet de la charge de travail sur le système étudié.

3.2 Intégration de l'agent

Nous introduisons de nouveaux éléments au modèle de simulation à événements discrets, qui sont les pas de temps et l'entraînement du réseau de neurones. La notion de pas de temps n'est pas vraiment adaptée à la SED, car elle se confond avec les pas de temps de la simulation. Par conséquent, nous remplaçons les pas de temps par les interactions de l'agent. Une interaction de l'agent correspond à une modification de l'espace d'état et un ajustement des paramètres.

Nous utilisons un algorithme de *Branching Dueling Q-Network* (BDQ) qui vise à régler des problèmes ayant des espaces d'actions à plusieurs dimensions. L'idée est de traiter chaque dimension d'action avec un degré d'indépendance. L'architecture du réseau de neurones se divise entre les différentes dimensions d'action pour représenter la fonction Q-valeur, tout en conservant un "module de décision partagé" afin de conserver une trace de l'état d'entrée. L'architecture du BDQ a été reconnue comme efficace dans des environnements ayant des actions similaires, et permet une exécution plus rapide dans des espaces d'actions de grande taille (Tavakoli, 2018). Une représentation du BDQ est proposée à la Figure 5.

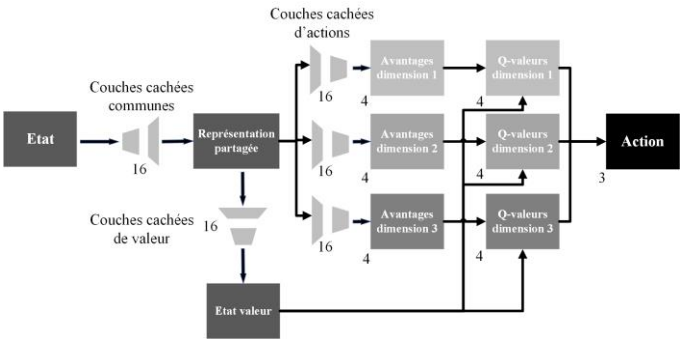

Figure 6. Architecture du BDQ inspirée de Tavakoli et al. (2018)

Nous considérons qu'une réplication est composée de 100 épisodes et qu'un épisode correspond à la simulation de 1000 jours. Nous travaillons sur 3 produits passant par des machines distinctes avec des temps d'opération et de réglage différents. L'espace d'état considéré correspond à la demande moyenne mobile des 30 derniers jours et les niveaux d'EFN de chaque produit.

L'espace d'action interagit avec les OST et les OSH. Nous discrétisons l'espace d'action afin de limiter la complexité, en utilisant un facteur α_{OST} et α_{OSH}. Ces facteurs peuvent prendre un nombre de valeurs finis, dans l'intervalle *[0, 1]*. Une action est donc un couple (α_{OST}, α_{OSH}), tel que :

Les facteurs représentent les variations des OST et des OSH, voir les équations (7) et (8), nous considérons que :

$$OST = \alpha_{OST} * TOR, (7)$$

$$OSH = DLT * (1 + \alpha_{OSH}), (8)$$

où TOR (*Top Of Red*) est le niveau supérieur de la zone rouge du *buffer* et DLT (*Decoupled Lead Time*) est le délai de découplage. Nous supposons que $\alpha_{OST} \in \{0.25; 0.5; 0.75; 1.0\}$ et $\alpha_{OSH} \in \{0.0; 0.15; 0.35; 0.5\}$. Ces valeurs respectent les recommandations de Ptak et Smith (2019), afin d'avoir *OST* < *TOR* et *OSH* > *DLT*.

Nous rappelons que l'objectif du RL est d'adopter une politique qui maximise la fonction récompense. La fonction récompense est une fonction qui est construite dans le but d'atteindre un objectif particulier. Pour notre atelier de production géré en DDMRP, nous considérons les trois objectifs suivants :

1. L'agent 1 tend à optimiser les stocks. Sa fonction récompense est définie par l'équation (9), où EFN est l'Equation du Flux Net présentée à l'équation (5), et TOG, TOY et TOR sont respectivement les valeurs supérieures des zones vertes, jaunes et rouges.

$$R_1 = \begin{cases} -\left(1 + \dfrac{EFN - TOG}{EFN}\right) si\ TOG < EFN \\ -1\ si\ TOY < EFN \leq TOG \\ 3\ si\ TOR < EFN \leq TOY \\ 0\ si\ 0 \leq EFN \leq TOR \\ -1\ si\ EFN < 0 \end{cases} \quad (9)$$

2. L'agent 2 vise à maximiser la satisfaction client. Sa fonction récompense est définie par l'équation (10).

$$R_2 = \frac{nombre\ de\ commandes\ complètes\ délivrées}{nombre\ de\ commandes\ reçues} \quad (10)$$

3. L'agent 3 concilie les deux premiers agents et tend à optimiser les stocks et maximiser la satisfaction client. Sa fonction récompense est formulée par l'équation (11) comme la somme des valeurs de R1 et de R2 normalisées.

$$R_3 = R_1 + R_2 \ (11)$$

Ces objectifs sont associés à trois agents distincts permettant d'explorer les possibilités du RL. Nous précisons que les 3 fonctions récompenses sont normalisées afin de pouvoir les comparer.

3.3 Expérimentation

Le plan d'expériences vise à explorer deux aspects : les limites de validité pour l'utilisation du RL et sa capacité à faire face à des variations de la demande. A cette fin, nous utilisons les quatre indicateurs clé de performance suivants pour soutenir notre analyse :

1. Le taux de service moyen. Il s'agit du ratio entre la demande livrée et la demande reçue.
2. Le niveau moyen des inventaires de produits finis après la période de réchauffement d'un épisode.
3. La récompense accumulée moyenne :
   $$RAM = (r_1 + r_2 + ... + r_t) / N\ (12)$$
   où N est le nombre d'épisodes et r_t est la récompense à l'épisode t. Cet indicateur est le principal d'un algorithme de RL et permet la vérification et la validation du modèle.
4. Le gain de récompense :
   $$GR = RAM_{100} - min(RAM)\ (13)$$
   la différence entre la valeur de la RAM à l'épisode 100 et la valeur minimale de la RAM. Cet indicateur nous permet d'évaluer l'efficience de l'apprentissage de l'agent.

Dans toutes les expériences, les OST initiaux prennent une des trois valeurs suggérées par Ptak et Smith (2019) : 50% du TOR (notée 50%), base de la zone rouge (notée RS) ou avec l'ADU (notée ADU). Comme les auteurs recommandent un OSH supérieur à un DLT, nous avons considéré 3 valeurs possibles pour l'OSH initial, soient : *1.0* × *DLT*, *1.25* × *DLT* ou *1.5* × *DLT*. Le plan d'expériences est résumé par le Tableau 1. Chaque scénario a été répliqué 3 fois afin de considérer l'aspect aléatoire.

Tableau 1 : Plan d'expériences

Expérience	**Expérience 1 :** Variation des OST et des OSH	**Expérience 2 :** Comparaison de différentes fonctions récompense	**Expérience 3* :** Variation de la charge de travail et la fréquence des pics
OST variable	Oui / Non	Oui	Oui
OSH variable	Oui / Non	Non	Non
Fonction récompense	R2	R1 / R2 / R3	R3
Valeur initiale d'OST	50% / RS / ADU	50% / RS / ADU	50% / RS / ADU
Valeur initiale d'OSH	1.0 /1.25 /1.5 * DLT	1.0 * DLT	1.0 * DLT
Fréquence des pics	1 tous les 5 jours	1 tous les 5 jours	Haute (1 tous les 5 jours) / Basse (1 tous les 20 jours)
Charge de travail	80%	80%	50% / 80% / 95%
Nombre de scénarios	27	9	18

* : Expérience où une comparaison avec une Baseline et un Know-It-All est faite

L'expérience 1 étudie les effets de la variation des OST et des OSH. L'objectif de cette expérience est de vérifier s'il est pertinent d'ajuster deux paramètres (OST, OSH) du modèle DDMRP et s'il est pertinent de les ajuster dynamiquement.

L'expérience 2 se concentre sur la notion d'objectif dans le RL et tente de comparer les résultats des trois fonctions récompense définies, chacune étant rattachée à un objectif précis.

L'expérience 3 est orientée métier et analyse l'effet de facteurs externes, soit la fréquence des pics de demande et la charge de travail dans l'atelier. Par conséquent, des profils de demande variés seront testés considérant différentes fréquences des pics et différents ratio charge-capacité. Cette expérience identifiera dans quel environnement le RL est plus efficace en comparant les résultats de l'agent avec ceux d'une Baseline (c'est-à-dire une simulation où les OST et les OSH sont fixes), et un agent Know-It-All, qui considère que toute la demande est connue d'avance.

4 Resultats et Discussion

4.1 Effets des paramètres ajustés sur le processus d'apprentissage

L'expérience 1 analyse l'effet de la variation des OST et OSH sur la performance du modèle DDMRP. Comme première étape de l'expérimentation, nous tentons de valider l'hypothèse d'un apprentissage possible avec les OST et les OSH. Pour cette étape, nous considérons juste la récompense R2 (maximiser le niveau de service), avec une fréquence de pics tous les 5 jours et une charge de travail de 80%. La Figure 6 représente la fonction RAM par épisode de scénarios à OST et OSH variables. Les résultats montrent que, pour tous les scénarios où l'OST est variable, le RL est fonctionnel, car la fonction RAM est croissante. Néanmoins, quand les OST sont fixes et les OSH variables, l'apprentissage n'est pas prononcé. On peut conclure que la variation des OSH seulement n'a pas d'impact sur le modèle de RL et ne permet pas l'apprentissage. Par contre, il est possible de faire de l'apprentissage par renforcement avec les OST.

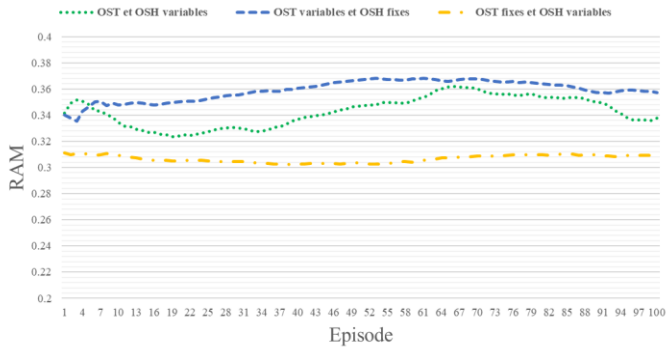

Figure 7. Evolution des RAM par épisode pour des scénarios de l'expérience 1

Pour les scénarios d'OST et OSH variables, l'agent apprend avec le pilotage des OST, mais est pénalisé par le pilotage des OSH. Dans ce cas, nous avons un espace à 16 actions, puisque $\alpha_{OST} \in \{0.25; 0.5; 0.75; 1.0\}$ et $\alpha_{OSH} \in \{0.0; 0.15; 0.35; 0.5\}$, ce qui donne 16 couples de (α_{OST}, α_{OSH}). Nous constatons que l'effet de la variation des OSH est négligeable vu que les actions visant à varier les OSH ont des effets similaires sur le système (voir Figure 6), ce qui nuit à l'apprentissage.

Enfin, nous précisons que le RL est davantage fonctionnel avec des espaces d'actions réduits. Cela permet d'avoir une meilleure exploration et de ne pas délaisser d'action. Un espace à 4 actions a plus de chances de réussir qu'un espace à 16 actions, car l'espace à 16 actions aura des chances de présenter des actions redondantes. Dans un environnement aussi complexe que celui du DDMRP, nous souhaitons favoriser un espace d'actions simple et de petite taille. Cela permet également de limiter la taille du réseau de neurones et donc de perdre en complexité. Pour envisager un espace d'actions plus grand, nous suggérons d'augmenter la longueur de la série de demande, mais cela augmente considérablement la complexité.

Pour le reste de l'étude nous considérons un espace d'actions de taille 4, en ne faisant varier que les OST et en considérant les OSH fixes à *1.0 × DLT*.

4.2 Choix d'un agent

La deuxième expérience compare les trois fonctions récompense R1, R2 et R3. Chacune a été définie avec un objectif précis et présente donc un processus d'apprentissage qui lui est propre. Le but est de déterminer la fonction récompense présentant une croissante satisfaisante de la fonction RAM. Le Tableau 2 affiche les gains de récompense et évalue l'efficacité de l'apprentissage. R3 ayant les meilleurs gains d'apprentissage, il semble que ce soit la meilleur fonction récompense. En d'autres termes, cela prouve que le RL est adapté à l'amélioration de la satisfaction client et à la minimisation des niveaux d'inventaire.

Tableau 2. Gains d'apprentissage de l'expérience 2 par fonction récompense et par valeur initiale d'OST

	R1	R2	R3
50%	0.014	0.022	0.031
RS	0.012	0.024	0.040
ADU	0.018	0.035	0.046

La Figure 7 affiche les taux de service (axe y) et les inventaires de produits finis moyens (axe x) pour les scénarios de la deuxième expérience. Nous considérons ces deux KPI à l'épisode 100 (le dernier épisode) puisque l'agent est supposé être bien entraîné à cette étape. Comme prévu, les scénarios utilisant R1 comme fonction récompense ont des niveaux de stocks plus bas que les scénarios avec R2. Cependant, il n'y a pas d'amélioration significative dans les taux de service avec R2. Les scénarios utilisant R3 (on rappelle que *R3 = R1 + R2)* présentent les avantages de R1 et R2 en termes de KPI industriels.

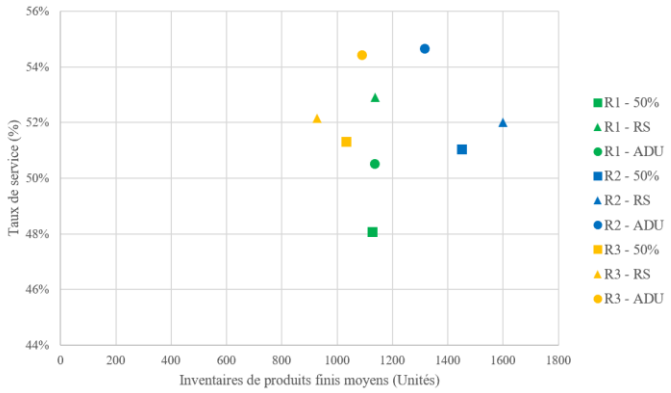

Figure 8. KPI industriels pour les scénarios de l'expérience 2 à l'épisode 100

Pour conclure, R3 est la meilleure fonction récompense en considérant des KPI industriels et d'apprentissage. Par conséquent, nous choisissons de continuer l'étude avec cette fonction récompense. Les deux premières expériences nous ont permis de construire un agent cohérent qui est adapté à faire du RL pour ajuster dynamiquement les paramètres d'un atelier géré en DDMRP.

4.3 Effets de la charge de travail et la fréquence des pics

Cette expérience vise à étudier les effets de la fréquence des pics et de la charge de travail. À cette fin, nous utilisons R3 comme fonction récompense avec des OST fixes à *1.0 × DLT* (voir Tableau 1, quatrième colonne) et nous analysons les KPI industriels (taux de service et inventaires moyens) du dernier épisode en comparant les résultats du RL avec ceux d'une Baseline et d'un agent Know-It-All.

Nous considérons deux niveaux pour la Fréquence des Pics (FP) et trois niveaux pour la charge de travail de l'atelier. Une basse FP et une haute FP correspondent respectivement à 1 pic tous les 20 jours et 1 pic tous les 5 jours. Trois niveaux de charge de travail sont testés : 50%, 80% et 95%. Nous présentons les résultats en considérant les moyennes sur les trois méthodes d'initialisation d'OST. Nous avons fait ce choix pour des questions de clarté, en vérifiant que la méthode d'initialisation a peu d'impact sur les indicateurs.

La Figure 8 présente les KPI industriels obtenus par un agent de RL, une Baseline et un agent Know-It-All à l'épisode 100 dans différents contextes. En premier lieu, nous remarquons que les performances de l'agent de RL sont proches de la performance du Know-It-All. Cela montre l'efficacité du processus d'apprentissage et démontre qu'un agent de RL est adapté à résoudre ce problème industriel quand la demande est inconnue ou difficile à prédire. Comme la demande est considérée comme inconnue du point de vue de l'agent lorsqu'il prend ses décisions à une certaine itération, il utilise uniquement l'entraînement réalisé avec la demande et les états des itérations précédentes.

En second lieu, nous essayons d'identifier les cas où le RL est meilleur que la Baseline. Nous constatons que pour les cas avec une haute FP (carrés dans la Figure 8), la Baseline est meilleure seulement pour une charge de travail à 50%, tandis qu'avec une basse FP (cercles dans la Figure 8), la Baseline est meilleure que l'agent pour des charges de travail à 50% et 80%. L'utilisation de l'agent de RL est pertinente d'autant plus que les pics sont fréquents (Haute FP), puisque l'agent est capable de détecter plus de pics. L'agent semble également être plus utile avec des charges de travail élevées (plus grand que 80% dans notre cas). Par conséquent, nous recommandons d'utiliser le RL avec des ateliers présentant des hautes charges de travail et faisant face à des pics fréquents.

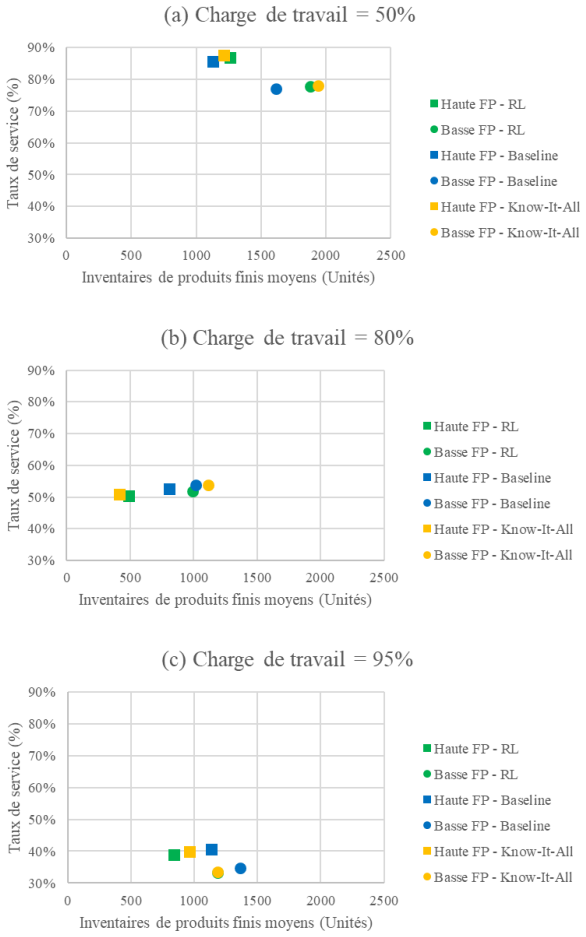

Figure 9. KPI industriels pour les scénarios de l'expérience 3 à l'épisode 100

5 CONCLUSION

Cet article présente une nouvelle approche de paramétrisation d'un système de production géré en DDMRP, faisant face à une demande atypique (avec pics). Un agent d'apprentissage par renforcement basé sur un *Branching Dueling Q-Network* a été développé afin d'ajuster deux paramètres du système (soit l'OST et l'OSH), et ainsi améliorer les performances d'un atelier géré en DDMRP.

D'abord, nous avons vérifié l'intérêt de faire varier l'OST et l'OSH. Nous avons montré que l'apprentissage est pertinent pour l'ajustement d'OST seulement. Ensuite, nous avons comparé les performances de trois fonctions récompenses différentes et avons démontré que l'utilisation de l'apprentissage permet de réduire les niveaux d'inventaire et augmenter les taux de service. Enfin, nous avons comparé les performances de l'agent développé à une Baseline et un agent Know-It-All. Les résultats montrent qu'il est pertinent d'utiliser le RL par rapport à la Baseline (diminution de 33% des stocks) quand l'atelier présente une haute charge de travail (plus de 80% du ration charge-capacité dans notre cas) et fait face à des pics fréquents. De plus, bien que l'agent de RL ait utilisé beaucoup moins d'information que l'agent Know-It-All (pour lequel nous avons supposé une demande connue à

l'avance, ce qui est irréaliste), nous avons démontré que l'agent de RL performe aussi bien que le Know-It-All.

Comme recherches futures, il serait possible d'explorer le RL ainsi que d'autres méthodes d'apprentissage automatique pour ajuster les paramètres du DDMRP. Il serait intéressant aussi de considérer plus de sources d'incertitude et de contraintes opérationnelles. Enfin, le RL pourrait être utilisé pour paramétrer le DDMRP pour des processus de production différents, tels que des processus divergents (par exemple, dans l'industrie de sciage ou l'industrie alimentaire) plutôt que des processus d'assemblage.

6 REFERENCES

Azzamouri, A., Baptiste P., Dessevre, G., Pellerin, R. (2021). Demand-driven material requirements planning (DDMRP): A systematic review and classification. *Journal of Industrial Engineering and Management, 14*(3), 439-456.

Bahu, B., Bironneau, L., Hovelaque, V. (2019). Compréhension du DDMRP et de son adoption : premiers éléments empiriques. *Logistique et Management, 27*(3), 20-32.

Cao, H., Xi, H., Smith, S.F. (2003). *A reinforcement learning approach to production planning in the fabrication/fulfillment manufacturing process*. 2003 Winter Simulation Conference, New Orleans, LA, USA.

Cuartas Murillo, C. A., Aguilar, J.L. (2022). Hybrid algorithm based on reinforcement learning and DDMRP methodology for inventory management. *Journal of Intelligent Manufacturing*.

Damand, D., Lahrichi, Y., Barth, M. (2022). Parametrisation of demand-driven material requirements planning: a multi-objective genetic algorithm. *International Journal of Production Research*, 1-22.

Dessevre, G., Ben Ali, M. (2020). *Modélisation et simulation d'un module d'ajustement de la capacité d'un système DDMRP*. 13ème Conférence Francophone de Modélisation, Optimisation et Simulation MOSIM'20, Agadir, Maroc.

Dessevre, G., Martin, G., Baptiste, P., Lamothe, J., Pellerin, R., Lauras, M. (2019). *Decoupled Lead Time in finite capacity flowshop: a feedback loop approach.* Paper presented at the 2019 International Conference on Industrial Engineering and Systems Management (IESM), Shangai, China.

Ihme, M., Stratton, R. (2015). *Evaluating Demand Driven MRP: a case based simulated study*. International Conference of the European Operations Management Association, Neuchatel, Switzerland.

Kortabaria, A., Apaolaza, U., Lizarralde, A., Amorrortu, I. (2018). Material management without forecasting: From MRP to demand-driven MRP. *Journal of Industrial Engineering and Management, 11*(4), 632-650.

Martin, G. (2020). *Contrôle dynamique du Demand Driven Sales and Operations Planning.* (PhD), Université de Toulouse, Toulouse, France.,

Miclo, R., Fontanili, F., Lauras, M., Lamothe, J., Milian, B. (2015). *MRP vs. demand-driven MRP: Towards an objective comparison*. 2015 International Conference on Industrial Engineering and Systems Management (IESM), Seville, Spain.

Morales, M. (2020). *Grokking Deep Reinforcement Learning.* (S. I. Manning Publications Co., NY, USA Ed.).

Ptak, C., Smith C. (2019). *Demand Driven Material Requirements Planning (DDMRP)* (I. Industiral Press, South Norwalk, Connecticut, USA Ed. Version 3 ed.).

Shofa, M. J., Widyarto, W.O. (2017). *Effective production control in an automotive industry: MRP vs. demand-driven MRP*.

Stockheim, T., Schwind, M., Koenig, W. (2003). *A reinforcement learning approach for supply chain management*. 1st European Workshop on Multi-Agent Systems (EUMAS), Oxford, UK.

Tavakoli, A., Pardo, F., Kormushev, P. (2018). *Action branching architectures for deep reinforcement learning*. Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA.

Wang, J., Qu, S., Wang, J., Leckie, J.O., Xu, R. (2017). *Real-time decision support with reinforcement learning for dynamic flowshop scheduling*. Smart SysTech 2017; European Conference on Smart Objects, Systems and Technologies, Munich, Germany.

Xanthopolous, A. S., Kiatipis, A., Koulouriotis, D.E., Stieger, S. (2018). Reinforcement learning-based and parametric production-maintenance control policies for a deteriorating manufacturing system. *IEEE Access, 6*, 576-588.