Un système d'aide à la décision pour les négoces de matériaux

Marc Souply

Normandie Université

UNIVERSITÉ CAEN NORMANDIE

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité **INFORMATIQUE**

Préparée au sein de l'**Université de Caen Normandie**

Un système d'aide à la décision pour les négoces de matériaux

Présentée et soutenue par

MARC SOUPLY

Thèse soutenue le 06/03/2024

devant le jury composé de :

M. MAURICIO CAMARGO	Professeur des universités - UNIVERSITE DE LORRAINE	Rapporteur du jury
M. STEPHAN CLÉMENÇON	Professeur des universités - Telecom Paris	Rapporteur du jury
MME CLAUDIA FRYDMAN	Professeur des universités - Ecole polytechnique Marseille	Membre du jury
M. TOMAS MENARD	Maître de conférences HDR - Université de Caen Normandie	Membre du jury
MME CECILIA ZANNI-MERK	Professeur des universités - INSA de Rouen Normandie	Membre du jury
M. GRÉGORY ZACHAREWICZ	Professeur - IMT Mines d'Alès	Président du jury
M. FRANCOIS RIOULT	Maître de conférences HDR - Université de Caen Normandie	Directeur de thèse

Thèse dirigée par **FRANCOIS RIOULT** (GREYC ALGORITHMIQUE)

MIIS

GREYC
Laboratoire de recherche en sciences du numérique

Dédicace

À ma mère **Anita** pour son courage, sa confiance et son sens du sacrifice, qui m'a soutenu même dans ses moments les plus difficiles.

À **Lou**, qui m'accompagne depuis ces presque 10 années d'études supérieures, pour sa gentillesse, sa sensibilité et son humour sans lesquels le chemin à parcourir aurait paru bien long.

Il me serait difficile de remercier sans une fastueuse, mais néanmoins fastidieuse énumération tous les amis, collègues et enseignants que j'ai eu l'occasion de rencontrer à Le Mans, Laval, Caen et Toulouse, et il me semble qu'un tel exercice me placerait aux antipodes des mots que je souhaite leur dédier.

Aux amis qui m'ont rendu la vie douce-amère,
À toi qui chaque jour me tires vers le haut ;
Tu fais d'inutilité fugace un tableau
Vous travestissez ma traversée en croisière

Du fond du cœur, merci.

Remerciements

C'est ainsi que ces trois années et demie se terminent, sur un sprint final qui n'est certainement pas recommandé après un marathon. Ces années de doctorat m'ont apporté autant de bagages techniques que de réflexion sur le monde auquel ceux-ci doivent apporter des solutions. Et je dois dire que cette thèse CIFRE qui m'a permis de joindre la recherche académique et opérationnelle, a été une opportunité passionnante d'apprendre, comprendre, expliquer, justifier et parfois capituler.

C'est bien sûr aux côtés de mes encadrants **François Rioult** et de **Bertrand Cuissart** que j'ai appris à synthétiser, justifier et affiner mon approche de la recherche académique. Sans leurs soutiens et leurs questionnements permanents, je doute que le présent manuscrit et les deux articles soumis aient été lisibles par un autre que moi-même. La patience et la concentration dont ils ont fait preuve à la relecture m'ont à de maintes reprises laissé humble. Enfin, je remercie encore François pour son ouverture d'esprit, qui m'a guidé sans me restreindre durant ces trois ans où il m'a conseillé sans y ajouter d'urgence inutile ou de formalisme étouffant. Je le remercie conjointement avec Bertrand pour le temps qu'ils ont trouvé à m'accorder, même dans des périodes intenses pour eux. Je remercie également le laboratoire du GREYC et l'équipe CODAG pour leur sympathie à mon égard, et plus encore mes compères **Etienne Lehembre**, **Alexis Mortelier** et **Mathias Dehais** qui m'ont aidé à me sentir à ma place au laboratoire alors que je passais la majeure partie de mon temps en entreprise. Je souhaite également remercier **Tomas Ménard** pour le temps qu'il m'a accordé lors de mes recherches en automatisation, celles-ci seraient sans nul doute tombées à l'eau par manque de temps sans un guide pour m'accompagner.

Pour leur lecture que je devine minutieuse de ce manuscrit, dont les enjeux peuvent sembler peu conventionnels, et leurs suggestions sagaces dans leurs rapports, notamment sur les états de l'art et les motivations derrière certaines décisions, je remercie chaleureusement **Mauricio Camargo** et **Stephan Clémençon**.

Je suis reconnaissant aux membres du jury de cette thèse, **Claudia Frydman**, **Tomas Ménard**, **Gregory Zacharewicz**, **Cecilia Zanni-Merk** pour le temps qu'ils m'accordent, et à Gregory et Cecilia de m'avoir accompagné avec bienveillance pendant mes

CSI.

Je remercie bien sûr toute l'entreprise **RMAN Sync**, qui a bien grandi depuis le début de cette aventure où nous sommes passés de 5 à plus de 30 employés. C'est bien sûr avec une certaine fierté que je leur présente ce manuscrit dont certaines lignes représentent pour eux tantôt des périodes difficiles, tantôt d'allégresse. J'ai eu la chance d'être valorisé et très libre dans la direction de mes recherches, facilitant un aspect que je sais souvent difficile dans les thèses CIFRE : l'adéquation entre le besoin fonctionnel et la recherche académique. C'est donc en toute sincérité que je remercie **Marc Malmaison**, **Renaud Delcoigne** et **Antoine Morace** pour leur compréhension des enjeux qui entourent ma thèse malgré les besoins toujours grandissant de notre jeune entreprise, et pour la confiance qu'ils m'ont accordée du premier jour à aujourd'hui. Enfin, il me faut remercier mes collègues et amis **Gautier, Grégoire, Hugo, Cyril et Marvin** qui nous ont rejoints au fur et à mesure de l'aventure, pour leur aide pratiquement journalière sur les problématiques qui entourent le déploiement de mes expérimentations. Ils m'ont permis de gagner un temps précieux sur tous les aspects de mise en production des outils que j'ai eu l'occasion de développer, il m'est aussi bien plus facile de me lever le matin pour les rejoindre.

Enfin, à cheval entre l'université et l'entreprise, je remercie mon ami **Anatole Gaudin** pour les nombreuses fois où il m'a accompagné sur des questionnements de modélisation mathématique et de fonctionnement du monde académique.

TABLE DES MATIÈRES

Chapitre 1

Introduction

Cette thèse a été effectuée dans le cadre d'un dispositif CIFRE au sein de l'entreprise RMAN Sync[1].

Dans l'optique d'ancrer concrètement ce travail de thèse, les constituants principaux du métier et les préoccupations fondamentales rencontrées par les acteurs du milieu sont détaillés dans ce chapitre, de même que les enjeux humains et matériels d'un système d'aide à la décision pour le négoce de matériaux.

1.1 Contexte industriel

Dans toutes les entreprises de production et/ou de distribution de biens, la gestion de la chaîne d'approvisionnement (« supply chain ») est la clé de voûte sur laquelle reposent toutes les problématiques d'optimisation, de prises de risques, de développements futurs et de gestion des relations commerciales. Cette chaîne concentre les questionnements sur les diminutions des coûts et l'augmentation des bénéfices sur tous ses maillons : choix produits, approvisionnement de la matière première, production, distribution, livraison, gestion de la ressources humaine et des stockages.

La chaîne d'approvisionnement Ce modèle de production est prépondérant dans nos modes de consommations actuels dès lors que le secteur secondaire est concerné. L'optimisation de cette chaîne sous tous ses aspects est largement étudiée depuis les années 50 [32] (bien que le terme «supply chain» n'apparaisse qu'en 1982). Il en résulte une littérature du sujet fournie et vaste, chaque entreprise ayant tendance à travailler avec ses spécificités émanant de contraintes terrain et d'un mode de fonctionnement qui lui est propre. Le plus souvent, les maillons qui composent la supply chain sont décomposés afin de diminuer la combinatoire et les rapprocher de problèmes connus dans le domaine comme

1. https://www.rman-sync.com/

le problème du sac à dos pour le stockage d'entrepôts ou de chargements, le problème du voyageur de commerce pour les livraisons et le problème de réapprovisionnement conjoint (joint replenishment problem - *JRP* [92]) pour la production de produits.

Prédire une demande La gestion du besoin est un point sensible de la suppply chain. Il existe deux façons de gérer cette demande : *poussée* (*push*) ou *tirée* (*pull*), respectivement impliquant l'attente d'un besoin pour initier une production, ou initiant la production en prévision de la demande.

Évidemment, dans le cas d'une stratégie de vente aussi élevée que possible, une stratégie *push* est de mise. Celle-ci comporte alors des risques importants de surstockage et amène le plus souvent un manque de réactivité sur la chaîne en cas de besoin non prévu. La prédiction du besoin est alors cruciale pour diriger la chaîne. Dans les travaux plus anciens, notamment ceux sur le JRP, la demande était le plus souvent considérée connue et continue. Cela s'explique assez facilement : avec moins de références produit et une application long terme, une prévision de ventes au mois, ou même à l'année, n'est pas très risquée. À moins d'un évènement catastrophique, les ventes suivent une tendance qu'il est possible de prédire et l'erreur finale cumulée sera faible.

Ce n'est pas le cas pour des références nombreuses et des volumes de ventes faibles voire chaotiques. La granularité temporelle est un facteur déterminant dans la recherche d'une solution adaptée, notamment pour l'évaluation de celle-ci, puisqu'en fonction des produits comparés, les pourcentages tendent à cacher les implications réelles : une erreur de 10 % sur un volume vendu de 10 000 se trouve être plus critique et lourde de conséquence qu'une erreur de 30 % sur un volume de vente de 100 unités.

S'approvisionner pour la demande Les *négociants de matériaux* (on parlera par la suite de *négoce*) sont peu connus du public alors qu'ils jouent un rôle essentiel dans le secteur de la construction. En apparence, le principe de base de ces métiers semble simple : acheter en grandes quantités auprès de fournisseurs et de grossistes des matériaux de construction tels que des plaques de plâtre, des tuyaux, de la peinture, et revendre ces produits aux artisans pour leurs chantiers.

Dans ce contexte, la gestion de la chaîne ne comprend pas la partie production, puisque les négoces sont principalement des revendeurs. Ils opèrent donc uniquement entre le fournisseur des produits et les consommateurs finaux, comme les artisans ou les clients particuliers qui passent commande ou font des achats en magasin. Il n'est alors pas question

d'optimiser la production de produits en fonction de la demande, mais plutôt d'optimiser les commandes fournisseurs que le négociant de matériaux doit passer. On dépendra du besoin des artisans, des relations et de la fiabilité des fournisseurs avec lesquels des commandes sont passées. La confiance des clients est primordiale pour assurer la pérennité de l'entreprise ; un chantier qui démarre en retard aura des implications bien plus importantes que celles induites par la rupture d'un produit dans une grande surface.

1.2 Du stratégique au tactique

Une variation importante dans la littérature au fil des années est l'apparition de décisions à effectuer à court terme. L'amélioration des capacités de traitement des ordinateurs permet de résoudre des problèmes plus spécifiques et nombreux qu'auparavant. Il est possible de passer d'une optimisation *stratégique*, c'est-à-dire une vision générale à suivre pour optimiser ses rendements, à une vision court terme et donc *tactique* : l'ensemble des actions qui permettront de mettre en œuvre la stratégie. C'est un pas supplémentaire vers une aide à la décision concrète mais pour laquelle il est difficile de proposer des solutions généralistes efficaces.

Des références nombreuses Concrètement, le négoce se heurte à des complications qui lui sont propres : le nombre de références produits et de fournisseurs est important, mais le volume de vente en lui-même ne l'est pas, menant à de nombreux besoins très chaotiques. Pour cette raison, la gestion du besoin est difficile et les marges déjà faibles du milieu sont largement amputées par le sur-stockage.

Un stockage complexe Ces nombreuses références sont une des forces des négociants, elles permettent de proposer une gamme de produits toujours plus importante, au détriment cependant des coûts de stockage (espaces, assurances, ressources humaines) et de transport des ressources. Il est fréquent de devoir effectuer une cession, à savoir déplacer d'un dépôt à l'autre des stocks qui ne se vendent pas là où ils sont entreposés. En effet, les négoces disposent de plusieurs agences et dépôts de stockages sur un territoire, gérés par des acteurs différents. Bien qu'une volonté d'uniformité existe au sein de ces entreprises, ces dépôts sont souvent en concurrence avec un microcosme local, ce qui mène à des *prix tapés*, à savoir des prix personnalisés en négociation directe avec le client. Ces prix tapés sont le plus souvent bien en-dessous de la grille tarifaire de l'entreprise, car le vendeur a

des objectifs de quantité plutôt que de rentabilité. De même, la gestion de stock est laissée à la discrétion des gérants de ces entrepôts. Ce stock mène notamment à l'immobilisation des capitaux pour l'entreprise, l'empêchant d'investir dans des projets plus porteurs pour l'avenir.

1.3 Une généralisation difficile

Bien que la plus-value d'un système d'aide à la décision soit évidente, aussi bien d'un point de vue externe que pour les dirigeants, le métier de négociant est par essence imprécis : les négoces sont des entreprises de taille variable, mondiales ou locales à une région, qui négocient des prix directement au comptoir ou par téléphone avec des clients. Il est impensable de réformer le métier pour le faire entrer dans une case informatique parfaitement codifiée comme le sont les grandes enseignes de la distribution.

Des employés et des habitudes Les acteurs experts de leurs domaines, parfois imprécis, ont la main sur les prix vendus dans leurs agences et les réapprovisionnements prévus ; ils n'aspirent pas à entrer dans un mode de fonctionnement régi par un système informatique. L'idée n'est donc pas d'automatiser les maillons de la chaîne, mais plutôt de proposer un système d'aide sur ces maillons pour garantir une prise de décision optimale des responsables.

Un système vieillissant Ce manque de processus automatisé a des impacts importants sur les possibilités d'essor du domaine : contrairement à de nombreuses entreprises, les négoces ne sont pas intégrés dans des modèles industriels 4.0, notamment au travers de l'IOT, à savoir des capteurs et des équipements connectés générant de la donnée pour la prise de décision. Dans le cas des négoces, c'est l'AS 400 qui est majoritairement utilisé depuis les années 90. Bien que fonctionnel, celui-ci limite grandement la récolte et le stockage des données, d'autant plus que les entreprises du domaine n'en ont de toute façon pas fait leur priorité. Loin de la collecte et du traitement *big data* avec le souci du volume et de la qualité, le négoce, comme de nombreuses entreprises de petite taille, se heurte à une problématique de *small data*. En effet, ces entreprises ne récoltent que les données dont elles ont besoin pour l'exercice strict de leur travail.

Une concurrence efficace Il est donc très difficile de proposer les solutions mises en place par les géants du domaines, comme les GAFAS du négoce de matériaux (Mr Bricolage, Leroy Merlin...). La plus-value du négoce sur de telles entreprise est la relation avec les clients et les fournisseurs, mais à mesure que l'inflation augmente, que les marges des négociants sont de plus en plus faibles, tandis que la construction immobilière cherche toujours à réduire les coûts de construction pour dégager des profits, cette proximité pourrait finir par ne plus suffire face à une économie toujours plus serrée.

Un manque d'informations objectives Pour améliorer la compétitivité de telles entreprises, il est nécessaire de connaître les indicateurs clés permettant de déterminer où mobiliser du capital. Bien que les responsables aient des avis sur les points faibles des entreprises (catégorisation clients, prévision du besoin, fiabilité fournisseur), il est difficile de cibler clairement et efficacement les maillons faibles de la chaîne sans données précises sur les processus. Lorsqu'un produit est vendu à un prix faible, vérifier le bien-fondé de cette offre est difficile. Prospection d'un nouveau client, prix d'appel pour les spécialistes, concurrence, prix obsolète ou commercial visant des objectifs de volume de vente sont autant de bonnes et mauvaises raisons.

Le plus souvent, la marge est calculée entre le prix d'achat et de revente, sans considérer les coûts humains supplémentaires de stockage, d'inventaire, de déchargement, qui peuvent rapidement mener à des ventes au bénéfice négatif. Mais l'étude des élasticités tarifaires ou de l'intérêt de disposer de nombreuses marques d'un même type de produits est difficile sans historique.

De nombreux autres sujets gravitant autour du négoce de matériaux existent : extraire des motifs fréquents des paniers consommateurs pour les recommandations [95], déterminer les profils consommateurs et y adapter des tarifs préférentiels [90], définir les produits substituables en cas de rupture [39], caractériser les produits principaux et les produits secondaires qu'ils impliquent comme le carrelage et sa colle [19].

Bien que cette thèse s'articule sur l'axe principal qu'est la chaîne d'approvisionnement – déterminer le besoin et optimiser les commandes fournisseur – il n'est pas envisageable d'espérer proposer une solution adéquate sans une compréhension approfondie du microcosme des négociants et des interactions entre les travailleurs qui y opèrent.

1.4 Rentabiliser le temps humain

Le domaine du négoce de matériaux ne dispose d'aucune solution automatisée pour tous les points évoqués ci-dessus, ce qui demande une énergie et un temps de travail considérable aux responsables qui étudient tous ces comportements sur Excel. L'application des concepts de l'industrie 4.0 doit mener à une meilleure mobilisation de la main d'œuvre. Par exemple, réduire la commande à passer chez un fournisseur à l'ajustement et la validation, plutôt que devoir se renseigner sur les produits en rupture actuellement, d'estimer ceux à venir et déduire les quantités à commander en prenant en compte le stock de sécurité et le maximum stockable.

Affiner la granularité Les études sont le plus souvent menées au niveau de l'entreprise dans sa globalité, et pas à celui des agences locales. S'il est possible de prédire la consommation d'une entreprise entière sur un produit selon une granularité temporelle (l'année) et spatiale (toute l'entreprise régionale), menant comme expliqué précédemment à des prises de décision stratégique, la tactique n'est pas maîtrisée : il conviendrait d'étaler le renouvellement d'un stock annuel sur des mois voire des semaines, et spécifiquement pour certains entrepôts et agences plutôt que d'autres. Une répartition homogène du stock annuel entre tous les dépôts de part équivalente impliquerait de coûteux mouvements de cession interne pour pallier des ruptures de stocks et du sur-stockage.

Accélérer les décisions fines par des indicateurs concrets Pour toutes ces raisons, l'automatisation ou la mise un place d'un système pour conseiller, bien que difficile, est indispensable à moyen-long terme malgré les réticences humaines à entrer dans un système pré-établi et le manque de donnée disponible à l'heure actuelle. Sur l'aspect strictement opérationnel, les logiciels propriétaires utilisés pour planifier les réapprovisionnements de ventes sont lents, les prédictions sont le plus souvent calculées à la demande pendant les heures de travail. Il est donc fréquent pour les responsables des approvisionnements de lancer le logiciel puis de quitter leur poste de travail en attendant que le contexte décisionnel se mette à jour.

1.5 Un contexte informatique stimulant

Le contexte multi-fournisseur, multi-produit, multi-dépôt et temporalité courte (pour garder une bonne réactivité), implique des complexités et des temps de traitement très

élevés dans la résolution ou l'exécution de processus informatiques, nécessitant des notions d'optimisation calculatoire importantes et la détention de serveurs nombreux et puissants. Cette complexité est une autre explication de la difficulté pour les négoces à engager eux-mêmes les ressources nécessaires pour disposer d'un système de gestion compétitif. Cette complexité implique de disposer de collaborateurs qualifiés et de serveurs de calculs puissants, l'investissement en interne est élevé pour une réussite incertaine et financièrement difficile à évaluer.

1.6 Un contexte adapté aux PME : le *small data*

Bien que ce travail de thèse soit présenté spécifiquement pour le négoce de matériaux, il nous semble que les thématiques abordées concernent de nombreuses petites et moyennes entreprises. Il n'est pas raisonnable d'espérer appliquer à des entreprises de ces tailles des concepts de la science de la donnée pensés pour fonctionner avec un flux constant de donnée : elles n'en ont ni les moyens, ni les besoins. Les notions de données massives parasitent actuellement l'industrie et le grand public, et découragent les acteurs de se lancer dans une rationalisation de leurs activités. Des coûts élevés, des processus et des installations complexes et peu ou pas de compréhension en interne amènent à considérer cette rationalisation comme inabordable. Enfin, le retour sur investissement de disposer d'historique de donnée ne pouvant se faire qu'à long terme, il condamne définitivement l'intérêt pour cette transition vers un système décisionnel. Pourtant, comme les sites internet en leur temps, il ne parait pas avisé d'ignorer cette transition dans le futur : le concept de *small data*, bien que moins en vogue, tend à se répandre et montre la volonté pour ces entreprises d'améliorer leurs processus. L'impératif de maîtrise de son empreinte carbone en est une porte d'entrée, l'augmentation des coûts des matières premières, notamment dans la construction, en est une autre.

Ce travail de thèse montrera notamment que la puissance de calcul actuelle permet, outre l'existence du big data, de renforcer, spécialiser et généraliser des méthodes plus anciennes et plus adaptées à des volumes de données modestes, notamment grâce au *meta-learning*.

1.7 Organisation du mémoire

Cette thèse a pour but d'adapter et de comparer les méthodes existantes de gestion de la chaîne d'approvisionnement afin de proposer une approche alternative pour les entreprises non-compatibles avec l'ère du *big data* ainsi que ses pré-requis en volume de donnée et puissance de calcul. Nos études montrent qu'il est possible, pour un investissement modeste, d'améliorer la prise de décision dans le négoce de matériaux au travers de choix de méthode et d'allocation de ressource appuyés par des concepts basiques du *meta-learning* et une compréhension du contexte industriel étudié. En effet, la compréhension de ce contexte permet de déterminer les aspects clé de précision attendue, de granularité temporelle et de coûts qui permettront, malgré le nombre de références produit élevé et la nécessité d'un fonctionnement en temps réel, de simplifier certains aspects des problématiques rencontrées. Si les grandes entreprises parviennent à adapter le travail des employés au système informatique pour rationaliser leurs activités, dans le cadre des plus petites entreprises qui dépendent de relations plus profondes entre vendeurs et artisans spécialistes, c'est le système qui doit s'adapter aux employés et à leur façon de travailler.

Notre proposition d'un système d'aide à la décision adapté au négoce plus économe en ressource et en donnée sera articulée sur deux axes que nous considérons comme primordiaux : la *prédiction du besoin* et l'*optimisation du réapprovisionnement* pour satisfaire ce besoin. Ces deux aspects seront abordés au travers de deux états de l'art distincts (Chapitres 2 et 3) qui s'intéresseront particulièrement au contexte social motivant leurs évolutions, ceci pour situer les négociants de matériaux sur l'échelle du besoin et des contraintes techniques intrinsèques à celui-ci. Nous montrerons ainsi la pertinence et la proximité entre ce qui est aujourd'hui appelé le *small data*, et les méthodes utilisées jusqu'en 2010 avant l'investissement massif dans la collecte de données.

Ces deux états de l'art seront suivis par deux contributions. La première (Chapitre 4) compare différentes approches pour prédire les ventes et propose une approche pour satisfaire le besoin prédictif de milliers de produits. La deuxième contribution (Chapitre 5) compare l'optimisation de commande par solveur avec des méthodes heuristiques, dont la consommation de ressource stable est plus raisonnable que pour le solveur. Ces deux contributions sont fortement influencées par le contexte du négoce de matériaux, visant des résultats *rapides* pour des prédictions et des optimisations de réapprovisionnement de commandes nombreuses et fréquentes dans un contexte *multi-produit multi-fournisseur*. Dans les deux cas, en prenant en compte un contexte opérationnel où le temps d'exécution

est une donnée critique, le volume de donnée disponible faible et le nombre de prédictions et d'optimisations élevé, nos contributions se fondent sur des méthodes traditionnelles ou viennent compléter des méthodes plus récentes.

Finalement, une conclusion et une réflexion sur les pistes d'amélioration seront proposées.

Chapitre 2

État de l'art : prédiction de séries temporelles

2.1 La prédiction des ventes

L'introduction de ce manuscrit a mis en exergue l'importance d'une prédiction de la demande efficace comme socle immuable d'une chaîne logistique optimisée. Bien que ce domaine ait évolué avec le temps et des capacités de calcul toujours plus importantes, le marché a lui aussi suivi cette course et exige des modèles prédictifs toujours plus précis, rapides et adaptés au contexte, avec toujours plus de variables à prendre en compte. L'état de l'art suivant est composé d'un point sur les définitions nécessaires à une bonne compréhension, suivi d'une présentation des travaux pré-existants menant à une conclusion sur les limites inhérentes au contexte de la prédiction de volume de vente dans le cadre du négoce de matériaux.

2.1.1 Série temporelle

Définition 1 (Série temporelle) *Une série temporelle est une suite de données indexée par une variable entière de sémantique temporelle.*

On parlera également de *série chronologique*. La suite de valeurs numériques représente l'évolution temporelle d'une quantité.

Une telle suite peut être manipulée à l'aide d'outils mathématiques qui permettent d'en analyser le comportement, de prévoir ses variations futures ou passées.

Par exemple, les ventes regroupées à la semaine réalisées par une entreprise sont une série temporelle, de même que le résultat journalier obtenu par un lancer de dé. En effet, une série temporelle peut tout aussi bien représenter une suite numérique aléatoire qu'une suite de valeurs constante, le contexte de la variable suivie n'influençant d'aucune manière sa définition en série temporelle.

L'indexation temporelle peut concerner n'importe quelle période de temps, du moment qu'elle est de progression constante. On parle d'un choix *de granularité temporelle*, elle peut donc s'exprimer en heures, demi-journées, années...

Mathématiquement, une série temporelle se décompose en trois composantes, avec t la granularité temporelle de la série (heure, jour, mois...) :

— la tendance notée T_t, une fonction à variation lente ;

— la saisonnalité notée S_t, une fonction périodique ;

— un bruit noté ε_t de moyenne nulle et à variabilité faible.

Une série temporelle X_t peut être *additive* :

$$X_t = T_t + S_t + \varepsilon_t$$

ou *multiplicative* :

$$X_t = T_t * S_t * (1 + \varepsilon_t).$$

La décomposition est la clé des modèles prédictifs traditionnels comme ARMA ou les lissages exponentiels simple, double ou triple, bien que les représentations puissent varier et des composantes disparaître ou s'ajouter. C'est le cas du *niveau* (la valeur de prédiction) dans le cadre d'un lissage exponentiel, ou de la différenciation pour stationnariser une série temporelle, étape indispensable à l'utilisation d'un modèle ARMA. Une série temporelle est dite *stationnaire* si ses propriétés statistiques (espérance, variance, auto-corrélation) ne varient pas dans le temps. Notamment si sa Covariance ne dépend que de la différence en temps :

$$\exists u \in \mathbb{N}, Cov(X_s, X_t) = Cov(X_{s+u}, X_{t+u}) \forall (s,t) \in \mathbb{Z}^2$$

Il est utile de considérer la stationnarité car elle est une composante de la décomposition pour certains modèles. En effet, la plupart des modèles statistiques qui permettent de prédire les séries temporelles, par exemple ARMA et ses dérivés, ainsi que les méthodes de lissage exponentiel double et triple cherchent à déterminer et utiliser des variations cycliques entre des périodes de temps. Cependant, cette recherche de relation linéaire dans les valeurs passées des processus auto-regressifs (AR) ou à moyenne mobile (MA) sera faussée si les propriétés statistiques de la série temporelle changent au fil du temps, autrement dit, si celle-ci n'est pas stationnaire. De manière générale, il est recommandé de rendre stationnaire les séries temporelles pour utiliser des modèles prédictifs qui ne modélisent pas la tendance ou la saisonnalité. Enfin, l'utilisation d'une régression linéaire

avec des variables explicatives sur une série temporelle à expliquer non stationnaire pourra mener à une *régression fallacieuse*. Une explication plus en détail de la régression fallacieuse et des différences entre corrélation, causalité et causalité indirectes est proposée dans l'annexe A. De manière générale, si la stationnarité n'est pas nécessaire pour tous les modèles prédictifs, la question doit toujours se poser. Dans la suite de cette étude, cette notion disparaît car les modèles de forêt aléatoire, de réseau de neurones et de moyenne annuelle ne nécessitent pas la stationnarité de la série temporelle, et le modèle ARIMA utilisé effectue une différenciation pour stationnariser la série temporelle si nécessaire.

2.1.2 Prédiction du volume des ventes

La *prédiction du volume des ventes* est un problème qui consiste à utiliser l'historique de ventes précédentes pour *prédire les ventes à venir*, soit les volumes écoulés à venir.

Modèles prédictifs et variables prédictives

L'analyse prédictive consiste à utiliser des données pour prédire les valeurs futures. Ce processus utilise l'analyse des données, le machine learning, l'intelligence artificielle et les modèles statistiques pour identifier des tendances susceptibles de prédire les comportements futurs, en utilisant un *modèle prédictif*.

Définition 2 (Modèle prédictif) *Étant donnée une série temporelle X_t, un modèle prédictif est une* fonction du temps *qui fournit une valeur de la série.*

Nous nous intéressons ici à la prédiction univariée : les modèles prédictifs basés sur la prédiction de vecteurs comme VARMAX (Vector Autoregressive Moving Average with eXogenous regressors model) ne seront donc pas abordés.

Un modèle prédictif est généralement utilisé pour fournir une valeur de la série à un temps situé dans le futur. Ce modèle peut avoir une expression mathématique – par exemple, une fonction constante égale à la moyenne de la série –; cependant, cette expression est rarement disponible mais est le résultat d'un algorithme d'apprentissage. Par exemple, un réseau de neurones est entraîné sur les données, il constitue un modèle prédictif dont l'expression mathématique n'est pas disponible mais fournit une valeur pour un instant du futur.

Dans le cadre de l'analyse prédictive d'une série temporelle, on considère que l'ordre dans lequel apparaissent les valeurs numériques contient également de l'information que le modèle prédictif se doit de valoriser.

Afin d'obtenir des prédictions en accord avec les variations de la réalité, on considérera fréquemment des variables supplémentaires dans les modèles prédictifs. Ces variables sont dites explicatives ou *exogènes*, à l'inverse de la valeur à expliquer qui est, elle, *endogène*. Lors de l'ajout de telles variables, une *corrélation* positive (l'une monte quand l'autre monte) ou négative (l'une monte quand l'autre descend) est espérée entre la variable endogène et les variables exogènes.

Ces variables exogènes peuvent être :

— *quantitatives*, comme la température en degré ;
— *qualitatives*, comme une discrétisation de la température en Froid/Doux/Chaud ;
— *binaires* pour noter l'existence d'un évènement, par exemple la présence de neige.

Il est important de noter qu'un modèle entraîné à l'aide de variables exogènes devient dépendant de celles-ci pour pouvoir proposer une prédiction. Par exemple, si l'on souhaite utiliser la météo, une prédiction des ventes à deux mois nécessitera de disposer d'une prédiction à deux mois de la météo.

Une prévision d'ensemble est une méthode consistant à combiner plusieurs prédictions, qui peuvent être basées sur plusieurs modèles prédictifs ou plusieurs paramétrages d'un modèle prédictif afin de proposer une unique prédiction. C'est notamment de cette manière que fonctionne le modèle de la forêt aléatoire [10].

Notions supplémentaires

Les modèles de prédiction de série temporelle sont souvent *auto-regressifs*. Cela signifie que les valeurs précédentes de la variable que l'on cherche à prédire influencent celle-ci, le plus souvent pondérées par la distance temporelle entre la valeur à prédire et la valeur d'historique concernée.

2.2 Travaux existants

La première partie de cet état de l'art s'inspire de l'excellente rétrospective de l'évolution des modèles de prédiction jusqu'en 2005 proposée par De Gooijer et al [23], en l'adaptant au contexte du négoce et en la prolongeant jusqu'à nos jours. Une réflexion actualisée est proposée sur les évolutions récentes dans le domaine de la prédiction et l'influence des sociétés sur son développement.

Un monde déterministe

Avant le XX^{e} siècle, le monde scientifique est fortement influencé par le déterminisme qui régit le monde, symbolisé par le *démon de Laplace* qui, connaissant la position et le mouvement de chaque particule dans l'univers à un instant précis, peut déduire un futur exact. C'est pour ces raisons que les premiers modèles prédictifs sont empiriques et/ou déterministes. C'est notamment le cas de la régression linéaire (Adrien-Marie Legendre en 1805) et de la moyenne mobile au début du XX^{e} siècle.

C'est surtout le *lissage exponentiel* qui sera utilisé dès le XVIII^{e} siècle pour effectuer des prévisions, bien que les fondations statistiques ne seront réellement établies que dans les années 50 [36, 46]. Ce lissage donne aux observations passées un poids décroissant exponentiellement avec leur ancienneté. Muth montrera finalement en 1960 que la fonction de lissage exponentiel simple (SES) est optimale pour prédire une série temporelle aléatoire [77].

Bien sûr, les notions de saisonnalité et de tendance existaient déjà, de même qu'une réflexion sur la pertinence des prédictions d'un modèle en fonction des caractéristiques de la série temporelle ciblée. Pegels proposera une classification d'une série temporelle en fonction de la nature additive ou multiplicative de sa tendance et ses motifs saisonniers [79]. Ce travail sera étendu par l'ajout d'une tendance amortie dans la classification [34].

Jusqu'en 2000, les travaux se sont particulièrement concentrés sur l'amélioration de méthodes d'initialisation du modèle, et l'étude des paramètres empiriques utilisés par le lissage exponentiel. Ces travaux mènent alors à la possibilité de proposer une méthode (et ses paramètres) en fonction d'une classification parmi 15 possibilités : 5 types de tendances, (aucune, additive, additive amortie, multiplicative, et multiplicative amortie) et 3 types de saisonnalités (aucune, additive et multiplicative). Il en ressortira notamment 4 configurations et modèles particulièrement efficaces [51] : SSE (pas de tendance, pas de saisonnalité), la méthode linéaire de Holt (tendance additive, pas de saisonnalité), méthode additive de Holt-Winters (tendance additive, saisonnalité additive) et la méthode multiplicative de Holt-Winters (tendance additive, saisonnalité multiplicative). Il est intéressant de noter qu'il existe très peu de travaux sur une prédiction multivariée du lissage exponentiel.

En parallèle et ce depuis les années 80, les modèles déterministes de lissage exponentiels sont critiqués car il est impossible de proposer un intervale de prédiction, le modèle étant déterministe et donc par essence non soumis à une quelconque stochasticité. Il souffre d'une comparaison avec les modèles de régression, qui eux le permettent puisqu'ils intègrent un terme ε correspondant à l'erreur. L'aléatoire apparaît dans les modélisations. Malgré cela, de nombreux auteurs montreront au début du XXI^{e} siècle que le lissage exponentiel simple est une solution fiable [88, 18], notamment parce qu'il n'est pas dépendant d'une pré-sélection de modèle comme le sont les modèles ARIMA, qui requièrent une étude de la série temporelle ciblée et une compréhension avancée de la modélisation mathématique pour proposer des prédictions fiables [49].

La naissance de la pensée stochastique

C'est avec les balbutiements de la physique quantique qu'émerge la pensée de l'indéterminisme dans la science au début du XX^{e} siècle. Des modèles stochastiques sont alors proposés, ces modèles cherchent notamment à approximer le hasard. Un tournant dans le domaine de la prédiction est le travail de Yule, affirmant que les séries temporelles peuvent être considérées comme la réalisation d'un processus stochastique [120].

C'est à partir de cette idée que les modèles *auto-régressifs* et en *moyenne mobile* ont commencé à être expérimenté. Bien qu'efficaces, ces modèles impliquent de définir plusieurs paramètres spécifiques à la série temporelle ciblée. Les connaissances seront alors résumées par Box et Jenkins, qui proposeront une méthode d'estimation des paramètres encore populaire aujourd'hui : la méthode Box-Jenkins [9]. Cette méthode et l'utilisation grandissante des ordinateurs a un impact déterminant sur la prédiction de série temporelle et la popularité des modèles ARIMA. Les recherches se concentrent alors sur la recherche du meilleur paramétrage d'un modèle ARIMA en fonction de la nature de la série temporelle, notamment au travers d'une ré-édition du livre de la méthode Box-Jenkins, *Time series analysis : forecasting and control*, en 1990.

Il est particulièrement important de noter que les améliorations et les modèles plus récents tendent à trouver un compromis entre le fantasme du démon de Laplace et l'aléatoire, ou le « non maitrisé », en ajoutant autant que possible du contexte. C'est notamment le cas des modèles comme ARMAX, qui permettent d'ajouter des régresseurs supplémentaires au modèle, basés sur une série temporelle extérieure comme la météo par exemple, tout en conservant la modélisation de l'erreur.

Sortir de la linéarité

Jusqu'alors, le modèle ARMA et ses variations sont des modèles spécifiques de régression linéaire basés sur des observations de la variable ciblée et du contexte passé. En réalité, on considère que l'approximation de série temporelle par des fonctions non linéaires remonte aux années 30, avec la représentation en série de Voltera [112]. La notion de mémoire est également présente puisque la sortie d'un instant t dépend de l'entrée du système à tous ses instants précédents. Volterra démontrera que toute fonction continue non linéaire peut être approximée par des séries de Volterra. Cette première réflexion sera reprise par Wiener dans un ouvrage clé de l'étude des problèmes non-linéaires : *Nonlinear problems in random theory* [115]. Bien que l'intérêt pour l'approximation de modèle non linéaire devienne importante dans les années 70 où les limites des modèles linéaires se confrontent de plus en plus à la réalité pratique, le paramétrage et l'entraînement de tels modèles théoriques restera impossible avant les années 80 à cause la complexité très élevée qu'ils impliquent.

Dans un récapitulatif des modèles non linéaires disponibles, De Gooijer et al. concluent que la supériorité prédictive des modèles non linéaires n'est pas clairement établie [24].

Basée sur les modèles auto-régressifs, la classe de modèle SETAR [105] en est l'extension non linéaire, celle-ci consistant à utiliser un ensemble de coefficient correspondant à un régime spécifique. Lorsque les valeurs passées dépassent certains critères (seuils), on considère que le régime change. Bien sûr, déterminer les critères et les multiples coefficients a donné lieu à de nombreuses publications et à la classe des modèles prédictifs à « changement de régime », dont le « Markov-switching » [42], basant le changement de régime uniquement sur l'état actuel du système. Le but étant d'estimer la probabilité du passage d'un régime à l'autre d'après l'historique.

Enfin, dans l'idée similaire de faire varier les coefficients mais cette fois de manière individuelle, des modèles à coefficients fonctionnels (FCAR) sont proposés [31]. L'idée derrière les modèles FCAR est de faire varier les coefficients à l'aide de fonctions basées sur une autre variable, endogène ou exogène.

Il est impossible de parler de modèles d'approximation non linéaires sans aborder les progrès de ces 20 dernières années sur les réseaux de neurones [102], bien que des questionnements concernant l'utilisation de ces réseaux comme solution miracle à tous les

types de prédiction étaient déjà levées en 1993 [17]. La comparaison entre les modèles traditionnels dérivés de la régression et d'ARMA est fortement étudiée au début des années 2000, les problèmes inhérents aux réseaux de neurones sont rapidement constatés : le sur-apprentissage et l'explicabilité inexistante du résultat notamment.

De nombreuses publications mettent en avant la supériorité d'un réseau de neurones simple (une couche cachée) sur les modèles linéaires [98], puis que les modèles linéaires produisent des meilleures prédictions que les ANN [43]. Pour les prédictions économiques, le réseau de neurones à une seule couche cachée est par ailleurs un modèle très populaire sur cette période.

Popularité du réseau de neurones et ses variantes

L'évolution sur les dix dernières années dénote clairement d'un intérêt tout particulier pour les réseaux de neurones, propulsés par l'explosion du big data [66]. C'est notamment le cas pour les réseaux de neurones récurrents (RNN) [91], qui sont très utilisés pour traiter la donnée séquentielle, et plus spécifiquement encore les LSTM (long short-term memory) [45] qui permettent de disposer d'une « mémoire » au sein même des cellules du réseau.

Des modèles hybrides [104] ou d'ensemble [101] apparaissent également. Il est impossible à l'heure actuelle de choisir un modèle qui s'appliquerait efficacement à toutes les prédictions, la nécessité d'expérimenter pour comparer reste inhérente aux prédictions et au contexte associé à celles-ci. Une piste d'amélioration récente est la génération de données, notamment utilisée en NLP et en imagerie, aux séries temporelles [99].

Prédiction et volume de donnée faible

La prédiction dans un contexte small data était déjà explorée au début des années 2000, lorsque les données disponibles étaient limitées. Les modèles de réseau de neurones à une seule couche étaient courants, et les problèmes de surapprentissage déjà bien connus. Cela a conduit à des méthodes telles que l'arrêt précoce de l'entraînement si aucune amélioration n'est constatée après un nombre défini d'itérations sur l'ensemble de test, dès 2002. À moins d'une granularité temporelle très fine permettant d'obtenir une quantité de données importante, les données sont généralement disponibles à une fréquence quotidienne, voire horaire dans les cas les plus favorables. Le risque de surapprentissage est

donc considérable, en particulier si la taille du réseau de neurones n'est pas adaptée. De plus, les modèles de prédiction de série temporelle auto-régressifs se basant sur les valeurs précédentes pour prédire la suivante, il est risqué de prédire de nombreuses valeurs, car un effet de cascade sur l'erreur de prédiction risque de prendre des proportions importantes [89]. Dans ces conditions, disposer d'une granularité plus fine ne résout pas le problème : il faudra prédire $7 * 24$ valeurs au lieu de 7 pour passer d'une prédiction horaire à journalière, remplaçant le manque de données par des problématiques de prédiction long terme.

2.3 Synthèse et limites

L'adoption de modèles de prédiction sur le terrain a souvent été tributaire des progrès réalisés en matière de puissance de calcul. L'évolution dans le domaine de la prédiction a toujours suivi un intérêt pratique croissant, en éliminant progressivement les contraintes (telles que la linéarité) ou en combinant diverses méthodes, comme la régression et l'auto-régression.

De nouvelles possibilités et méthodes ont été accompagnées de nouvelles métriques d'évaluation des résultats, telles que MAE, RMSE et AICC, ou de modèles prédictifs spécifiquement adaptés grâce à des décompositions, notamment de la saisonnalité et de la tendance, en utilisant des estimateurs de plus en plus complexes avec de nombreux coefficients. Le réseau de neurones semble représenter la conclusion logique de cette évolution, mais il ne doit pas être considéré comme une solution miracle ni comme une incarnation du démon de Laplace. Entraîner un réseau de neurones avec peu de données reste une tâche ardue. Bien que le concept de small data commence à émerger, la recherche dans ce domaine est nettement moins populaire que celle portant sur le big data [16].

Bien que les réseaux de neurones excellent dans les domaines de l'imagerie et du traitement naturel du langage, leur suprématie en matière de prédiction n'est pas établie. Il est essentiel de les comparer à d'autres modèles, souvent plus simples et moins gourmands en ressources, surtout en présence d'un faible volume de données. En outre, l'explicabilité des réseaux de neurones par rapport aux modèles traditionnels dérivés d'ARMA pose problème. Si dans d'autres domaines comme le traitement de la langue et l'imagerie, il est la seule option disponible ou obtient des résultats qui justifient de se passer de cette explicabilité, ce n'est pas le cas dans le domaine de la prédiction de série temporelle. Nous

verrons dans le chapitre 4 que les réseaux de neurones sont un outil additionnel intéressant pour la prédiction des ventes, tout comme les modèles ARMA, les forêts aléatoires et les estimateurs naïfs. La proposition d'un système prédictif pertinent implique de savoir quand utiliser quel modèle.

Chapitre 3

État de l'art : Optimisation du réapprovisionnement

Après la prédiction de la demande, notre attention se porte naturellement sur la mise en place d'un réapprovisionnement économe et d'une gestion de stock maîtrisée. Afin de maximiser les bénéfices engendrés par une vente, réduire les coûts qui l'accompagnent est indispensable. Pour ce faire, une maîtrise des différents maillons qui composent la chaîne logistique est requise. La compréhension de ces concepts, apparus avec l'informatisation des systèmes dans les années 60 [32] doit nous permettre de proposer un système d'aide à la décision applicable au négoce de matériaux. Sur la même structure que l'état de l'art de la prédiction des ventes proposée chapitre 3, plusieurs définitions des concepts clés du réapprovisionnement logistique sont proposées, puis une revue de la littérature orientée sur les modèles permettant l'optimisation du réapprovisionnement est réalisée. Finalement, une réflexion sur l'applicabilité des méthodes actuelles dans le contexte du négoce de matériaux justifie les expérimentations menées chapitre 5, cherchant à tirer le meilleur parti des modèles existants.

3.1 Optimisation du réapprovisionnement

3.1.1 Définitions

Le réapprovisionnement joint

Le problème du réapprovisionnement joint (joint replenishment problem ou JRP [37]) consiste à réapprovisionner chaque produit afin d'éviter les ruptures de stock pour un coût global de commande et de stockage minimal, en bénéficiant au maximum des promotions et remises disponibles sur les quantités et les panachages produits.

Il y a notamment deux points de vue sur ce sujet, l'un portant sur une production

interne, où le but est de minimiser les coûts de production (ajustement des machines). L'autre, celui qui nous occupera ici, concerne les frais de commande à un fournisseur, à savoir les coûts de préparation, de réception et de transport.

La littérature découpe en deux le coût total :

— le coût de commande *majeur*, qui n'est pas dépendant des produit commandés : il est dépendant de la quantité totale commandée. À condition d'atteindre un certain seuil total sur la commande, notamment de prix, les frais de commande et de livraison sont réduits à zéro. On appelle ce mécanisme *le franco (franc de port)*.

— le coût de commande *mineur*, qui dépend des produits commandés : il comprend le coût réel des produits, qui peut être vu à la baisse en fonction de remises quantitatives sur un produit spécifique ou un ensemble de produits. Il comprend aussi les frais de manutention et de stockage.

Ceci est à première vue contre-intuitif, parce que l'on aurait tendance à considérer que le coût d'acquisition des produits est le coût le plus élevé, c'est avant tout un coût indissociable de la vente qui permet de générer des bénéfices. Ce n'est pas le cas de la livraison qui est une perte sèche de bénéfice, tandis qu'il est possible de commander plus de produits pour atteindre le seuil de transport gratuit et ne pas payer la livraison. Dans le cadre de la production, c'est le coût de la mise en route des machines si elles sont arrêtées qui est le plus souvent considéré comme le coût majeur.

Valeur d'un produit

Dans le cas où tous les besoins sont considérés connus comme c'est le cas ici, la demande est entièrement adressée, ce qui signifie que la marge sur la vente d'un produit n'entre pas en compte dans une estimation de valeur. L'estimation de valeur concerne différents critères comme le besoin à long terme, le classement d'intérêt du produit et la difficulté à atteindre le seuil de transport gratuit chez un fournisseur où l'on commande ce produit de manière ponctuelle.

Stock maximal

Le stock maximal est la quantité théorique (ou physique) maximale que peut stocker un entrepôt spécifique. Si l'on choisit de se baser sur un concept physique, il est alors question de volume du produit et de place disponible. En revanche, on peut s'intéresser au concept théorique lorsque le stock maximal est calculé afin d'éviter un surstockage, en bloquant un ré-approvisionnement trop important de ce produit. Par exemple, bien que je

puisse physiquement stocker 1000 unités d'un produit, on constate une consommation de ce produit de seulement 60 unités par an, je n'ai alors aucune bonne raison de me baser sur le stock maximal physique pour gérer mon stockage.

Stock minimal ou de sécurité

À l'inverse, le stock minimal correspond au seuil en-dessous duquel l'on considérera que la quantité disponible est critique. Ce stock peut servir à lever des alertes, ou garder une marge de manœuvre pour de la prédiction à long terme du besoin.

Rang de Pareto

Il s'agit d'un classement attribuant une lettre ; A,B ou C à un produit. Ce classement est établi à partir du constat suivant : 20 % des produits d'une entreprise impliquent 80% de ses revenus. On peut alors déduire qu'une efficacité sur ces 20 % garantira une très bonne rentabilité générale. Puis 30% des articles suivants représentent environ 15 % de la valeur totale du stock, et finalement les 50 % des articles qui restent représentent environ 5 % de la valeur totale du stock.

Taux de rotation

Indicateur fréquemment utilisée dans le milieu, le taux de rotation permet d'indiquer si un produit est facilement écoulé, ou si, à l'inverse, son écoulement est lent. Un stock « mort », a un score dont le taux de rotation est 0. Le taux de rotation est calculé en divisant le nombre total des ventes sur une période donnée, par exemple un an, par le stock moyen du produit. Ce stock moyen est estimé comme étant la moyenne entre le stock au début et à la fin de la période.

3.2 Travaux existants

L'optimisation du réapprovisionnement chez un fournisseur est à l'étude depuis les débuts de l'industrialisation. En réalité, avant de prendre en compte l'aspect multi-produit d'un réapprovisionnement comme cela sera le cas notamment à partir des années 60 [107], l'optimisation mono-produit est assez peu étudiée, pour des raisons assez simples : sans l'aspect multi-critère, l'optimisation d'une commande d'un produit est assez simple. Il s'agit en effet de trouver un équilibre minimisant la somme des coûts de stockage et

de livraison/production. La modélisation de l'optimisation multi-produit en découlera spontanément. Cet état de l'art s'appuie notamment sur les vues d'ensemble proposées par [37, 63, 80].

3.2.1 Industrie 3.0

C'est dans les années 60/70 que les entreprises vont fortement investir dans les ordinateurs afin d'automatiser des opérations de comptabilité, de gestion de stocks et de traitement de la donnée, ce qui explique la concentration de travaux sur l'optimisation de réapprovisionnement sur cette même période. La puissance de calcul permet déjà à l'époque des méthodes d'optimisation multi-critères comme celle du réapprovisionnement de plusieurs produits. Cette minimisation de coût prend en compte la fréquence des commandes et la possibilité de commander plusieurs produits à la fois trouvera rapidement son nom : le Joint Replenishment Problem (problème de réapprovisionnement joint) avec des premiers articles utilisant cette dénomination [69].

3.2.2 Le problème de réapprovisionnement joint (JRP)

La modélisation du problème de base est simple, et a le mérite d'unifier les contextes pouvant varier d'une entreprise à une autre. On retrouve la base de la modélisation dans [37]. L'idée est tout simplement de considérer le temps comme une succession de périodes pendant lesquelles il est possible de commander. Cependant, chaque commande, peu importe le nombre de produits qu'elle contient, a un coût fixe (livraison, manutention...), et implique des coûts à long terme (coûts de stockage). Il s'agit donc de minimiser le coût total, en commandant des produits en groupe afin de limiter le coût fixe de commande, tout en limitant les coûts de stocks et ce, en n'acceptant aucun défaut de commande. Dans la mesure où cette demande est constante, déterministe et continue, il est possible de déterminer une stratégie à appliquer à l'horizon. Dans ce cas, on pourra parler d'une politique de réapprovisionnement puisque celle-ci n'est sujette à changement qu'en cas de modification du coût de commande, ou du stockage.

Les cycles de commande

La notion clé du problème de réapprovisionnement constant déterministe en temps continu est de déterminer le cycle de commande d'un produit. Il correspond à la politique de commande qui sera déterminée pour réapprovisionner chaque produit. Par exemple,

un produit qui doit être commandé à chaque fois que c'est possible aura un cycle de 1. Un produit devant être commandé 1 fois sur 3, aura un cycle de 3. On commandera alors ensemble les produits qui disposent des mêmes cycles, permettant, pour n produits d'un cycle, de diviser par n le coût fixe de commande. Plus le coût fixe de commande est élevé, plus le gain est important. À l'inverse, si le coût de commande est nul ou très faible, l'optimisation est inutile.

Regroupement des cycles

Sur la base de ces cycles, il existe deux façons de regrouper les articles. La méthode directe, qui consiste à commander ensemble les produits dont le cycle de commande est identique : tous les produits consommés en 3 semaines sont commandés toutes les 3 semaines. Dans la méthode indirecte, les produits, dont les cycles sont identiques ou multiples les uns des autres, sont commandés ensemble : les produits consommés en 6 semaines sont commandés avec les produits consommés en 3, mais le double de ces derniers doit être commandé pour tenir 6 semaines. La principale différence se situe au niveau du surcoût, au niveau de la commande elle-même pour le groupement direct (plus de coûts de commande). Le coût supplémentaire du modèle indirect proviendra du coût de stockage. Après étude, le regroupement indirect est légèrement moins coûteux et moins intensif en termes de calcul [27]. Les recherches et les comparaisons sur ce sujet sont cependant toujours en cours [65].

Complexité et optimalité

Le JRP est prouvé NP-HARD par Arkin et al [2], par une réduction au problème 3-SAT. Une spécialisation du problème initial implique l'existence de cycles stricts appelés Joint Replenishment Problem Strict Cycle (JRPSC) : au moins un produit est commandé à chaque cycle. Un algorithme a été proposé par Goyal si cette condition est remplie [38].

La recherche de l'optimum reste impossible sur des problèmes impliquant de nombreuses références de produits. S'appuyant sur les travaux antérieurs de Goyal, Silver propose une heuristique décrite comme plus facile à mettre en œuvre [93]. Il conclut également en soutenant que l'optimalité n'est pas nécessaire pour ce problème : considérer que la demande est constante, déterministe et continue permet de prendre des décisions stratégiques à long terme, mais n'est pas utile pour l'aspect opérationnel à court terme. Une réponse raisonnable est aussi pertinente dans ce contexte qu'une réponse optimale. Nous

pensons que cette hypothèse est toujours d'actualité. Silver ajoute également plusieurs notions au modèle de base, telles que la capacité maximale des entrepôts ou l'existence de niveaux de sous-groupes dans les groupes de commandes initiaux pour les produits similaires tels que les variantes de couleur. L'objectif est d'indiquer des coûts de fabrication différents en fonction des objets commandés. On devine au travers de cette spécification une volonté de se diriger vers des décisions tactiques plus utiles à court terme pour les entreprises.

Poursuivre l'optimalité : réduire le temps de calcul

Bien sûr, l'affirmation de Silver qu'une réponse raisonnable est acceptable n'a pas fait cesser les recherches visant l'optimalité. Mosh et Kaspi proposeront notamment un algorithme afin de déterminer le cycle de commande optimal pour chaque produit sur la base d'un minorant et d'un majorant de ce cycle [61]. En réduisant la plage de valeur possible pour ce cycle, la complexité initiale du problème est grandement diminuée [58]. Par la suite, de nombreuses propositions apparaissent dans le but de réduire les bornes mathématiques minimales et maximales afin d'améliorer le temps d'exécution de l'algorithme [26, 109, 111]. Il n'en demeure pas moins qu'il s'agit d'un problème NP-HARD, dont le coût de résolution devient rapidement prohibitif lorsque le nombre de produits concernés augmente. Les différentes propositions de réduction des bornes sont le plus souvent efficaces sur un certain nombre de produits, mais déterminer ces bornes devient rapidement plus coûteux que résoudre le problème lui-même.

Une réponse raisonnable : les méthodes heuristiques

En parallèle des recherches sur l'optimalité, plusieurs heuristiques sont proposées afin d'approximer une bonne solution dans des durées raisonnables. [64] propose une solution heuristique basée sur un algorithme génétique qui, bien que globalement moins efficace que la méthode RAND [60] (version améliorée de l'heuristique de Silver décrite précédemment), s'avère pertinente lorsque le nombre de produits augmente. Les méthodes heuristiques sont plus faciles à mettre en œuvre et à adapter aux besoins, en particulier pour ajouter des contraintes supplémentaires telles qu'un budget maximum ou un poids maximum à transporter. Les propositions précédentes ne prenaient pas en compte de telles contraintes, et leur représentation mathématique n'est pas triviale, alors qu'il est assez facile pour un algorithme meta-heuristique de pénaliser les mauvaises solutions, au prix cependant de l'optimalité.

3.2.3 Une demande dynamique : le DJRP

Lorsque l'on cherche à s'approcher de l'aspect tactique permettant de prendre des décisions concrètes, il est rapidement évident qu'une demande continue est extrêmement rare. Cette hypothèse était jusqu'alors nécessaire afin de proposer une politique de réapprovisionnement. Poussé par les progrès informatiques en terme de puissance de calcul, le besoin de propositions concrètes pour les entreprises est rapidement apparu. C'est notamment le cas pour les entreprises de nourriture qui impliquent un système de distribution au plus proche de la demande pour gérer les produits périssable [113]. On note dans le même temps des améliorations déterminantes dans le domaine de la prédiction, et l'apparition d'historiques de données plus fournis. Le problème du réapprovisionnement joint à demande dynamique (DJRP) est alors abordé au début des années 2000 [7].

Le DJRP est basé sur une demande déterministe mais variable. Il fournit donc des solutions plus concrètes et à court terme pour les entreprises. Notre étude s'inscrit dans cette catégorie puisqu'il ne parait pas pertinent de gérer la demande comme un flux continu constant. De la même manière que le JRP, le DJRP est NP-HARD [70]. Plutôt que de déterminer le cycle optimal dans un horizon temporel infini, on cherche à optimiser le coût total après un certain nombre de périodes. De plus, un indice temporel supplémentaire est ajouté afin de connaître le besoin ainsi que le prix d'un produit au temps t.

Optimalité du DJRP

En 2004, une nouvelle formulation est proposée par Boctor et al. pour résoudre le DJRP par le biais de la programmation linéaire [6]. Un concept de perturbation est utilisé pour extraire l'algorithme des optimums locaux, cette perturbation est proche de celle proposée par les mutations dans les algorithmes génétiques. Quatre catégories d'algorithmes optimaux sont proposées : Programmation dynamique [94, 57, 121], branch-and-bound [28], branch-and-cut [84], et Dantzig-Wolfe decomposition (column generation) [84]. Globalement, toutes ces implémentations proposent de résoudre des problèmes maintenus endessous de 100 produits, et un nombre de périodes temporelles limité à 20. Dans la mesure où ces solutions proposent le résultat optimal, c'est le temps moyen de résolution du problème qui est évalué par les auteurs. Comme le JRP standard, en fonction du nombre de produits, l'espace de solution à parcourir est immense, raison pour laquelle les instances évaluées sont le plus souvent d'une complexité maîtrisée.

Méthodes heuristiques du DJRP

Des heuristiques sont proposées pour améliorer la gestion de problèmes plus complexes : six sont présentées par Boctor et al. [6], Fogarty-Barringer [30] étant la meilleure des six d'après cette même étude. En général, la comparaison des résultats entre les algorithmes se fait en les exécutant sur des problèmes générés de manière aléatoire, dans la mesure où c'est avant tout une efficacité face à la complexité qui est évaluée, pas une pertinence sur le terrain.

La méthode basée sur la perturbation de Boctor et al [6] est plus performante que les recherches précédentes dans ce domaine. Cette méthode se divise en trois étapes. On commence par déterminer une solution réalisable en utilisant l'une des six heuristiques présentées. Après étude, les auteurs déterminent que l'utilisation de Fogarty et Barringer puis Silver-Kelle mène à la meilleure solution. Une fois cette solution optimisée obtenue, une perturbation de la solution est effectuée. Une période de temps est choisie au hasard ; si une commande existe déjà, elle est supprimée et toutes les quantités sont transférées à la commande précédente. S'il n'y a pas de commande dans cette période de temps T, on crée une commande et on y transfère toutes les quantités de la commande précédente. Enfin, Greedy Drop et Silver-Kelle sont appliqués. Si la solution optimale s'est améliorée, elle est conservée et le processus recommence en considérant cette solution comme notre nouvelle solution optimisée. Si la solution n'a pas été améliorée après n fois, le processus s'arrête.

Une heuristique clé : Fogarty et Barringer

La méthode heuristique proposée par Fogarty et Barringer est déterminante afin de réduire l'espace des solutions tout en limitant la dégradation induite. Elle simplifie le problème en affirmant que lorsqu'un réapprovisionnement est effectué, il doit couvrir exactement toutes les demandes jusqu'au prochain réapprovisionnement. Ainsi, une solution peut être caractérisée par ses périodes de réapprovisionnement. C'est également le cas pour notre problème, puisque le détaillant cherchera à couvrir une période spécifique jusqu'à laquelle il passera une nouvelle commande.

De Fogarty et Barringer au problème du sac à dos

Sur la base de ces hypothèses, on peut simplifier le problème en affirmant que la quantité commandée doit couvrir l'ensemble de la période visée (puisqu'aucun autre réapprovisionnement ne sera effectué pendant cette période), ce qui va dans le sens de la proposition heuristique de Barringer : commander, en fonction de la pénurie totale pendant la période, une quantité suffisante de chaque produit, et choisir les meilleurs fournisseurs auprès desquels les commander. Les commandes seront ensuite complétées avec les produits d'intérêt de manière à ce qu'elles atteignent les conditions et n'impliquent pas de frais de livraison. À première vue, il s'agit d'un problème de sac à dos non borné, où la limite de poids devient un objectif de prix qui peut être dépassé (mais qui reste un problème de minimisation), tout en maximisant la valeur à l'intérieur du sac. Bien qu'elle soit relativement spécifique dans notre cas, dans le cas plus général du JRP et du problème du sac à dos, la proximité entre les deux problèmes a déjà été abordée de manière explicite par [110].

Des études de (D)JRP plus spécifiques

On trouve dans la littérature plus récente des spécifications du JRP répondant à des problématiques que l'on devine plus orientées vers un certain besoin industriel. Peng et al. en proposent notamment une sélection récente [80], dans laquelle on peut retrouver des JRPs spécifiques, notamment un JRP prenant en compte les remises quantitatives des produits, et une remise plus générale calculée sur l'ensemble [21]. Une autre étude propose la prise en compte des limites de stockage (stock maximal) d'un DJRP [41]. Bien que la problématique multi-fournisseur soit centrale, on trouve assez peu d'études du DJRP adressant le problème, certainement parce que la complexité explose rapidement pour ce type de modélisation. Trois sont cependant relevées : [76], [108], et [122], proposant également des contraintes supplémentaires liées à l'empreinte carbone.

3.2.4 Trois méthodes de résolution heuristiques considerées par la suite

L'algorithme génétique (GA [75])

Basé sur le processus évolutionnaire, l'algorithme génétique vise à générer une bonne solution en créant une population de solutions possibles aléatoires appelées individus ou

spécimens. Chaque individu est classé en fonction de son aptitude : l'évaluation de sa fonction de score sur l'instance considérée. Les individus ayant un score d'aptitude faible (dans le cas d'une minimisation) sont plus susceptibles d'être sélectionnés comme parents pour la reproduction. Au cours de la phase de reproduction, des paires de parents sont sélectionnées en fonction de leur aptitude et sont croisées pour produire une progéniture. Cette opération de croisement combine la « génétique » des parents pour créer de nouveaux individus présentant une combinaison de leurs caractéristiques. Les nouveaux descendants forment la génération suivante de la population. Ce processus est répété pendant plusieurs générations, ce qui permet à la population d'évoluer au fil du temps et de converger potentiellement vers de meilleures solutions.

En appliquant de manière itérative les opérateurs de sélection, de croisement et de mutation, les algorithmes génétiques explorent l'espace de recherche et exploitent les régions prometteuses pour trouver une solution qui optimise une fonction d'aptitude donnée. Le processus se poursuit jusqu'à ce qu'un critère d'arrêt soit satisfait, par exemple en atteignant un nombre maximal de générations ou un niveau d'aptitude souhaité.

Il existe de nombreuses façons de croiser les parents, telles que le croisement en un point, le croisement en k points, le croisement uniforme... Pour maintenir la diversité génétique, la mutation est également utilisée lors du croisement des caractéristiques des parents : au lieu de provenir de l'un ou l'autre parent, la quantité commandée d'un produit à un fournisseur est modifiée aléatoirement. Enfin, afin de conserver les meilleurs résultats, l'élitisme est utilisé : les meilleurs x % des enfants seront transmis à la génération suivante de spécimens, intacts (mais ils demeurent des parents potentiels pour générer les $(100 - x)$ % restants de la population).

Le recuit simulé

Nommé d'après la technique métallurgique, le recuit simulé [5] est conçu pour échapper à l'optimum local en passant par les voisins de la solution lorsque ce voisin est meilleur, ou, dans certains cas, même si la solution se dégrade.

À partir d'une solution pré-existante, une solution voisine est générée au hasard. Cette solution générée peut remplacer la solution de départ, dans deux cas :

- le score d'aptitude s'est amélioré (nouvelle meilleure solution) ;
- la solution est acceptée avec une probabilité de $e^{-\frac{\Delta}{T}}$. Avec $\Delta = \hat{E} - E$ la différence entre le score d'évaluation de la nouvelle solution, et le score de la dernière solution acceptée et T une température qui diminue avec le temps, à chaque nouvelle

génération de solution : $T_t = T_{t-1} * 0.99992$. 'A chaque itération, la température va baisser et la probabilité d'accepter un voisin dégradant drastiquement la solution baisse avec elle.

Lorsqu'une nouvelle meilleure solution est rencontrée, elle est conservée en mémoire et à la fin, la meilleure solution rencontrée sera utilisée comme résultat.

Un problème du sac à dos très spécifique

Le knapsack [35] a fait l'objet d'études approfondies. Bien que la forme traditionnelle O/1 soit assez différente de nos contraintes, une variante de ce problème central, le *sac à dos non borné* [47], nous permet de remplir le sac avec un nombre illimité d'instances de n'importe quel élément.

Le problème de réapprovisionnement spécifique avec lequel nous travaillons semble s'apparenter à un sac à dos multidimensionnel non borné impliquant plusieurs sacs de différentes capacités. Chaque sac doit au moins atteindre la limite (seuil de livraison gratuite) et ne peut contenir que des produits spécifiques au catalogue du fournisseur. Le problème du sac à dos ne traite pas de la minimisation du retard, en particulier lorsque certains fournisseurs autorisent les envois fragmentés et d'autres non. Si l'on omet la dimension temporelle, notamment au travers de l'heuristique de Fogarty et Barringer, on s'approche d'un sac à dos multidimensionnel non limité, où chaque fournisseur représente un sac. La limite est un objectif et certaines quantités de produits sont obligatoires (même si elles peuvent être réparties dans plusieurs sacs). Le rapport typique valeur/poids deviendrait un rapport valeur/prix. Enfin, les produits nécessaires doivent arriver dans un délai minimal (en fonction du moment où la pénurie est prévue). Le délai pour un produit peut augmenter lorsque l'on commande d'autres produits chez le même fournisseur (sac) si celui-ci n'effectue pas de livraison fractionnée. Tout en conservant leur spécificité, les heuristiques du problème du sac à dos explorent l'espace des solutions de la même manière.

3.3 Synthèse et limites

Il est difficile de juger les recherches comme celles du DJRP comme ci celles-ci correspondaient exactement à notre problème, et de la même manière les recherches sur le problème du sac à dos. En effet, les recherches sur l'optimalité menées jusqu'ici impliquent le plus souvent un problème épuré, alors qu'il est le plus souvent question de variantes

plus compliquées, surtout à notre époque. [80] propose un résumé récent des avancées sur le DJRP, notamment le travail [83] étendant le problème à l'aide de contraintes et de coûts supplémentaires.

C'est en effet sur cet aspect que le bât blesse : il n'est pas pertinent pour les auteurs de proposer des modèles spécifiques adaptés à une entreprise, c'est pour cette raison que les modèles mathématiques du JRP et du DJRP sont généraux. Il n'y a pas d'intérêt pour la recherche à complexifier la description et la compréhension du modèle en le spécifiant à une problématique d'entreprise. Pourtant, cela pose des problèmes évidents d'ajout de contraintes comme nous le verrons dans la partie consacrée aux spécificités de notre contexte. Si les problèmes comme le JRP et le DJRP étaient déjà des problèmes simplifiés, exécutés sur des instances limitées, on imagine assez bien que l'ajout de contraintes non linéaires dans un contexte multi-fournisseur multi-produit va grandement impacter les performances, notamment des modèles optimaux. Si une prédiction de vente peut s'appliquer de la même manière à toutes les entreprises, ce n'est bien sûr pas le cas de la représentation mathématique d'un problème d'optimisation pour une entreprise.

Ceci étant dit, les heuristiques utilisées et les notions de cycles de commande proposent des solutions de simplification très intéressantes pour espérer proposer des solutions suffisantes dans un contexte de durée d'exécution limitée à quelques minutes, car il n'est pas exclu que parmi les nombreux produits disponibles, certains puissent se simplifier via ces cycles. Pour des instances importantes et des problèmes complexes, l'optimalité est utile pour évaluer les modèles heuristiques avec précision, mais elle n'est pas atteignable dans des durées acceptables pour une entreprise dont le métier se base sur les résultats journaliers de ce problème. Car si la puissance de calcul ne cesse d'augmenter, c'est également le cas des besoins et des contraintes que l'on souhaite prendre en compte. Par ailleurs, les études multi-produit actuelles se concentrent le plus souvent sur des instances d'une dizaine de produits [74, 71] pour en évaluer l'efficacité, un nombre de produit trop limité pour s'appliquer à notre contexte où les références sont bien plus nombreuses. La plupart des méthodes fonctionnant sur une dizaine de produit ne passeront pas à l'échelle sur un millier de ceux-ci. Des études plus récentes cherchent à combler ce vide car de nombreux milieux se heurtent à ces problèmes de complexité [73, 53].

Parmi les propositions déjà novatrices de modélisation prenant en compte plus de contexte industriel [80], l'aspect strictement pratique de leur utilisation est laissé de côté. Dans le contexte d'un négociant disposant de plusieurs milliers de produits à réappro-

visionner, avec plusieurs fournisseurs et une demande variable, il ne parait pas raisonnable d'espérer obtenir une réponse rapidement. C'est encore plus vrai si le nombre de contraintes augmente. C'est pourtant un pré-requis nécessaire à l'incorporation de tels modèles d'aide à la décision dans le processus. La littérature ne propose pas d'étude de modèles « coupés » après un certain temps d'exécution, par exemple une vingtaine de minute. Lorsqu'un employé a besoin de ce résultat pour poursuivre sa journée, il est nécessaire d'être capable de proposer rapidement une solution satisfaisante. Les modèles sont souvent comparés en fonction de la qualité de leurs résultats s'ils sont heuristiques, ou de la vitesse de résolution s'ils sont optimaux. En réalité, la vitesse importe aussi pour les algorithmes heuristiques, le but étant dans notre cas d'obtenir le meilleur résultat possible dans un temps limité.

Une étude approfondie du comportement (résultat, passage à l'échelle) des différentes façons de résoudre le problème doit donc être menée pour déterminer, en fonction du problème et sa complexité, la meilleure solution disponible. Cette étude sera l'objet du chapitre 5 dans laquelle les résultats et les temps d'exécution de plusieurs algorithmes de résolution seront évalués sur des problèmes de complexité variables, dans le but de déterminer, en fonction du contexte et du temps disponible, l'algorithme menant à un résultat fiable.

Chapitre 4

Prédiction efficace et économe des volumes de vente

Le contexte prédictif du négoce de matériaux est rendu difficile par un nombre de produits élevé, des volumes de vente très variables, et une granularité temporelle hebdomadaire, menant à des ventes chaotiques. Ce chapitre présente notre contribution dans le domaine de la prédiction des volumes de vente. Après une introduction qui aboutit à l'énoncé de notre positionnement en la matière, nous présentons les données métier disponibles et les modèles prédictifs considérés dans nos expérimentations. Cette contribution s'articule sur trois axes principaux : la compréhension, l'extraction et la généralisation de la saisonnalité d'une série temporelle, la comparaison entre quatre modèles prédictifs de nature et de complexité différente, et l'intérêt d'une approche meta-learning pour diriger les ressources informatiques disponibles là où elles sont le plus pertinentes pour rendre utilisable en pratique un tel ensemble de prédiction.

4.1 Introduction : le contexte prédictif

Contexte technique et évaluation

Dans tous les types d'industrie, la *demande* est le point clé sensible qu'il convient de maîtriser au mieux pour maximiser ses profits en minimisant ses stocks. En effet, tous les coûts sont impliqués par la demande : main d'œuvre, stock de sécurité, livraison rapide pour pallier une rupture de stock, coût d'achat, assurance du stock. Dans l'hypothèse d'une demande parfaitement prédite, on pourrait se passer ou minimiser une grande partie de ces coûts : pas de besoin de stock de sécurité, de stock maximal, stock minimal à assurer, flux tendu sur les commandes fournisseurs. De manière générale, des prédictions à moyen et court terme parfaites mèneraient à une réduction de l'importance de l'aspect tactique des entreprises, moins de gaspillage d'efforts humains et de moyens.

Granularité fine, grosses complications

Pourtant, comme discuté section 2.2, le domaine de l'analyse prédictive n'est pas le plus populaire ou celui qui se développe le plus parmi les différents domaines liés au machine learning, notamment parce que le volume de donnée séquentielles pour prédire une certaine demande est dépendant de la granularité temporelle ciblée. Cependant, réduire cette granularité pour augmenter le volume de vente requiert de disposer de la donnée dans le détail. Les négociants de matériaux n'ont, par exemple, pas la possibilité de réduire la granularité en-dessous de la journée, puisqu'ils ne disposent pas des heures de vente. Et même si cela était possible, la plupart des modèles prédictifs séquentiels utilisent la prédiction précédente pour effectuer la suivante, il n'est donc pas raisonnable d'espérer prédire une semaine avec une granularité à l'heure, représentant 50 (10 heures d'ouverture, 5 jours par semaine) valeurs à prédire. Finalement, les descripteurs utilisés doivent également s'adapter à la granularité, impliquant une météo plus précise à l'heure, mais aussi la possibilité de disposer de prédictions météo aussi précises dans le futur. Résultat : il est très difficile d'augmenter le volume de donnée séquentielle disponible pour l'entraînement.

Granularité et évaluation

Il est important de réfléchir à la représentation des erreurs pour évaluer des prédictions. En effet, les erreurs en pourcentage, bien qu'utiles pour comparer différents modèles et produits, ou estimer une erreur moyenne, cachent la réalité de ces erreurs. Si l'on s'intéresse aux erreurs en pourcentage, il sera bien plus facile de prédire la consommation électrique à l'échelle d'une ville entière sur une journée que celle d'une habitation. Effectivement, avec une granularité aussi faible, la consommation d'une habitation est difficile à prédire : elle dépend du comportement de ses habitants et variera drastiquement en fonction de leurs activités, de leurs absences, menant à des pourcentages d'erreur très élevés : 200 %, 300 %... Des niveaux d'erreurs qui ne seront jamais rencontrés au niveau de la prédiction électrique pour une ville entière, moins chaotique puisque les comportements auront tendance à se compenser. On dépassera rarement des erreurs de 1 % ou 2 %. Pourtant, avec une moyenne de 50 kWh par jour pour une habitation, l'erreur de 300 % ne représentera jamais que 150 kWh, quand l'erreur de 1% répartie sur 50 000 foyers d'une ville représentera 25 000 kWh.

Prédire au XXI^{e} siècle

On se rend alors vite compte que les modèles adaptés au *big data* très en vogue sont très difficiles à utiliser sans tomber dans le sur-apprentissage dû au manque de donnée. Même avec 10 ans de données de vente sur un produit, cela ne représente que 3 500 jours de données journalières, un échantillon faible pour entraîner un réseau de neurones. De plus, les comportements consommateurs ne sont pas intemporels ; entraîner des modèles sur des comportements très différents puisqu'ils datent d'il y a 10 ans n'apportera aucune information sur les habitudes actuelles des clients. Même s'il est difficile de proposer un modèle fonctionnel pour un produit, un expert avec du temps et des connaissances parviendra à composer un modèle pertinent. S'il y a maintenant 5 000 produits étalés sur différents fournisseurs, la tâche est humainement impossible. Il devient alors obligatoire de trouver un système de prédiction automatique efficace basé sur des descripteurs généraux.

Essayer de trouver des clés pour remplacer l'expert en prédiction, c'est l'un des objectifs du meta-learning. Par exemple, si l'entraînement de plusieurs modèles prédictifs permet d'en déterminer le plus efficace pour prédire une série temporelle spécifique (sur l'ensemble de validation), il implique cependant d'entraîner tous les modèles comparés. Cette consommation de ressource est inabordable dans notre contexte. En revanche, si un modèle de classification permet d'orienter le choix de ce modèle prédictif, nous serons alors en mesure de déterminer en amont le modèle prédictif à utiliser, économisant un temps précieux et menant à une prédiction fiable.

Négoce de matériaux

Bien que ce problème soit critique dans le secteur du négoce de matériaux où les marges de vente sont faibles, aucune des méthodes de prédiction populaires comme le réseau neuronal ou les modèles ARMA ne sont actuellement utilisées par les acteurs du secteur. Le plus souvent, ceux-ci s'appuient sur des méthodes empiriques liées à leurs compétences commerciales, proches de l'analyse chartiste [8], de méthodes plus anciennes telles que le modèle de Wilson [96, 56] ou l'analyse de l'oscillateur de Chaikin [1]. Si ces méthodes et les connaissances commerciales des spécialistes du domaine peuvent garantir une certaine rentabilité, une aide à la décision basée sur une prédiction pertinente des ventes pourrait aider les acteurs et potentiellement augmenter leurs marges bénéficiaires. Il convient donc de se demander pourquoi les modèles prédictifs traditionnels tels qu'ARMA, les forêts aléatoires ou les réseaux neuronaux ne sont pas déjà intégrés dans les modules ERP.

Cette absence de modèles prédictifs complexes peut s'expliquer au travers de plusieurs facteurs.

Puisqu'ici tout est négociable

D'un point de vue strictement logistique, le négoce de matériaux n'est pas facile à automatiser : il y a une volonté de maintenir l'expertise et le contact humain, les prix sont souvent négociables, les primes de fin d'année des fournisseurs sont difficiles à évaluer. Un produit a généralement plusieurs alternatives, les taux de transport sont complexes à prévoir. Il est habituel de constituer un stock pour satisfaire le client, même si celui-ci n'est pas rentable d'un point de vue coût-bénéfice. L'activité de négociant est également le résultat d'arrangements complexes entre les acteurs, rendant difficile une automatisation stricte du domaine.

Prédiction dans le négoce

Un système de prédiction du volume des ventes est donc difficile à construire, mais il constitue la pierre angulaire d'un système intelligent dans le négoce. Plusieurs raisons expliquent cette difficulté. La première est le nombre d'articles différents : avec une moyenne de 20 000 articles par commerçant, proposer une solution prédictive pour tous ces produits, qui peut être entraînée et mise à jour en un temps raisonnable, est un défi informatique. Par exemple, il n'est pas approprié d'entraîner un réseau neuronal pour chacun des produits. De même, la sélection de descripteurs pertinents est une tâche difficile. Certains produits sont sensibles aux conditions météorologiques, d'autres dépendent d'une ou plusieurs saisonnalités spécifiques (mensuelles, trimestrielles), et il est impossible de tester tous les descripteurs possibles pour déterminer une sélection optimale. Enfin, le volume des ventes au niveau du produit est trop fin pour prédire les ventes d'un entrepôt entier. En pratique, il est nécessaire d'adopter une vision stratégique et de considérer chaque produit comme un élément de l'entreprise auquel une portion restreinte des ressources prédictives peut être allouée.

Pour pouvoir faire une prédiction, une approche de meta-apprentissage comme la modélisation d'ensemble [11] nécessite d'entraîner chaque modèle sur chaque produit. D'autres approches [14] sont axées sur la prévision des pertes futures réalisées par les modèles étudiés, puis sur le choix spécifique du modèle approprié en fonction des conditions. Ces deux méthodes nécessitent l'apprentissage de nombreux modèles pour prédire

les volumes de vente ou les erreurs connexes, ce qui entraîne une charge de calcul très lourde, souvent inabordable.

Influence de la granularité sur le problème prédictif

Si les risques sont limités lorsque l'on prévoit toutes les ventes de tous les entrepôts car une moyenne générale stabilise le système, ce n'est pas le cas lorsque la granularité est la prévision des ventes d'un entrepôt particulier, car ces ventes peuvent fluctuer fortement en fonction de paramètres difficiles à déterminer ou à anticiper. Il existe différentes granularités spatiales, de l'entrepôt unique à l'ensemble de l'entreprise, en passant par les entrepôts d'une région ou d'un sous-ensemble variable. D'un autre point de vue, la granularité temporelle est également très importante et révèle une grande diversité de problèmes : il est plus facile de prévoir des ventes mensuelles que des ventes hebdomadaires. Cependant, une précision hebdomadaire peut être nécessaire, en fonction du délai d'approvisionnement et du taux de rotation des produits. Notre travail consiste donc à **proposer un système prédictif optimisant le temps nécessaire à l'apprentissage de l'intégralité des produits, en adaptant les granularités spatiales (entrepôt, région) et temporelles (prédictions hebdomadaires, journalières)**.

Positionnement

Le contexte du négoce de matériaux rencontre les problématiques classiques de prédiction du besoin, exacerbées par la nature même des négociants : le contact humain. Cette volonté de service implique notamment un nombre de références produit très élevé, des granularités temporelles et spatiales fines, des ventes commercialement désavantageuses. Ainsi, le but de cette étude est de proposer des prédictions sur un grand nombre de produits, entièrement automatisées, en réduisant au minimum le nombre de descripteurs spécifiques afin de généraliser au maximum le processus de prédiction.

Nous décidons d'aborder ces problématiques en démontrant l'efficacité de générer des descripteurs saisonniers en fonction des ventes connues, le processus étant peu coûteux et généralisable à l'ensemble des séries temporelles. Une comparaison de quatre modèles prédictifs justifiera l'utilisation d'un processus de meta-learning pour optimiser l'allocation de ressource disponible en fonction des produits. Dans l'ensemble, toute l'expérimentation mise en œuvre vise à réaliser les meilleures prédictions possibles, sans supposition sur les différents produits. Ces produits étant trop nombreux pour permettre une recherche

exhaustive de la meilleure solution prédictive et son hyper-paramétrage associé, une orientation des ressources informatique doit être réalisée pour contenir l'explosion du temps de calcul tout en minimisant le contrecoup sur la prédiction des articles.

4.2 Matériel et méthodes de prédiction

Données

Les données présentées résultent de la collecte et de l'addition des volumes de vente quotidiens de 1 000 produits d'un négociant de matériaux. À terme, notre objectif est de traiter 20 000 produits. On se concentrera d'abord sur l'analyse des résultats obtenus sur les dix produits générant le plus de volume de vente. Les périodes d'apprentissage et de test sont spécifiées dans le tableau 4.1. Les samedis et dimanches n'ont pas généré de ventes et sont donc exclus de l'étude. Dans le cadre des séries temporelles, la validation croisée est difficile à mettre en œuvre car les observations de l'ensemble d'apprentissage doivent précéder chronologiquement les observations de l'ensemble de test [50, chapitre 5.10]. En outre, les données disponibles étant peu nombreuses, l'arbitrage entre les tailles de l'ensemble d'entraînement, de l'ensemble de validation et de l'ensemble de test est complexe.

TABLE 4.1 – Les données utilisées et leur répartition entre les échantillons d'apprentissage, de test et de validation.

Données	Periodes	Taille (semaines)
apprentissage	du 2017-12-18 au 2020-08-30	141
validation	du 2020-08-31 au 2020-09-27	5
test	du 2020-09-28 au 2020-10-30	5

Dans le cas présent, nous avons choisi de consacrer 93 % des données à l'entraînement, 3 % à la validation et 3 % au test (voir tableau 4.1).

Pour des raisons de confidentialité, les données ne sont pas détaillées, pas plus que la nature des produits traités. On notera les produits en fonction de la catégorie qu'ils représentent, suivi d'un numéro identifiant le produit dans cette catégorie. Par exemple,

le produit C2-2 est le second produit rencontré dans la deuxième catégorie de produit. Le problème de prédiction est un problème de régression : à chaque période de vente d'un produit est associée une quantité, cette dernière étant l'information visée par la prédiction.

Un biais fréquent dans la prédiction du besoin est de considérer que les prédictions basées sur les ventes historiques ou la demande sont égales. Dans la mesure où il très rare de disposer des ventes ratées, la prédiction de besoin est en réalité souvent différente du besoin réel, que l'on tend alors à sous-évaluer. Cependant, dans le négoce de matériaux, un détaillant est en mesure de fournir immédiatement les références les plus demandées quitte à sur-stocker. On peut donc considérer exceptionnellement que pour ces références, la demande correspond aux ventes.

Les descripteurs utilisés pour représenter une journée peuvent être classés en trois catégories. Tout d'abord, trois informations météorologiques sont prises en compte : la force du vent, le volume des précipitations et la température de la zone de l'entrepôt. Tous ces descripteurs sont des variables quantitatives. Ensuite, la date est décomposée en huit variables discrètes : le numéro du jour dans l'année, le numéro de la semaine, le numéro du mois, mais aussi le numéro de la semaine dans le mois, *etc.* Enfin, quatre événements calendaires sont considérés sous forme de descripteurs binaires : les vacances, le confinement lié au COVID-19, la période COVID-19 de manière générale, et la présence d'une activité extraordinaire au travers de la détection de valeurs aberrantes. Au total, 15 descripteurs sont utilisés.

La variable ciblée par la prédiction est la somme des ventes réalisées en une semaine, sur une référence spécifique de l'entrepôt d'un commerçant. La nature exacte des produits est anonymisée mais ceux-ci sont regroupés par catégorie (C1, C2, C3) et numérotés (-1, -2, *etc.*). Le tableau 4.2 détaille la moyenne, le volume de vente maximum, le minimum et l'écart-type relatif pour chacun des produits étudiés. On constate qu'il n'y a pas de corrélation évidente entre les ventes moyennes d'un produit et l'écart-type associé, notamment pour les produits C1-2 et C2-3. Au vu des différences entre chaque valeur maximale et la moyenne associée, on peut s'attendre à la présence de pics d'activité extraordinaires.

4.3 Indicateurs de qualité de la prédiction

Différents indicateurs de la qualité de la prédiction, détaillés ci-après, sont utilisés : MAE, AIC et MAPE [50, chapitre 3.4 et 5.5]. Ces indicateurs sont utilisés pour le développement de modèles (voir section 4.4), dans un cadre d'optimisation (MAE pour le réseau

TABLE 4.2 – Détails sur les ventes des 10 produits étudiés.

Produit	Moyenne	Volume max	Volume min	écart-type relatif $\frac{\sigma}{\mu}$
C2-2	1333	4452	24	0.60
C2-1	370	1176	0	0.67
C4-1	1112	4460	0	0.69
C1-3	121	508	2	0.70
C3-1	64	231	1	0.71
C1-4	273	1010	3	0.71
C1-1	445	1872	11	0.73
C1-2	133	546	1	0.74
C1-5	103	458	0	0.77
C1-6	177	1125	0	0.86

neuronal et la forêt aléatoire) ou de choix de paramétrage (AIC pour le modèle DHR). Enfin, ils sont utiles pour évaluer la qualité prédictive d'un modèle, et ainsi maîtriser le risque encouru.

Sur la base d'une granularité hebdomadaire des volumes de ventes à prédire, on note :

— y volume des ventes sur la semaine (vérité) ;

— $\widehat{y}$ prédiction des ventes sur la semaine.

Mean Absolute Error (MAE)

La MAE est la moyenne de l'écart entre la prédiction et la vérité. Elle est utilisée à l'entraînement des modèles prédictifs (réseau de neurones et forêt aléatoire, voir section 4.4) comme étant la fonction à minimiser :

$$MAE = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n}$$

.

Critères d'information d'Akaike (AIC)

L'AIC fournit une estimation de la perte d'information lors de l'entraînement d'un modèle sur un paramétrage spécifique pour évaluer l'efficacité de celui-ci sur les données d'entraînement. L'AIC permet de choisir, parmi plusieurs paramétrages d'un modèle (ARMA ou DHR, voir section 4.4), celui représentant le mieux les données d'apprentissage.

Cependant, il ne donne aucune information sur la qualité intrinsèque du modèle par rapport aux autres et ne peut pas être un critère de sélection du modèle. $AIC = 2k - 2\ln(L)$ où k est le nombre de paramètres à estimer et L le maximum de vraisemblance.

Mean Absolute Percent Error (MAPE)

La pertinence de nos prédictions en terme de risque est évaluée à l'aide d'une version normalisée de l'erreur absolue moyenne (MAE) : l'erreur absolue moyenne en pourcentage (MAPE). La MAPE est la moyenne empirique des écarts de prédiction normalisés par la valeur réelle : $\mathcal{E} = \frac{1}{n} \sum \frac{|\hat{y}-y|}{y}$.

Puisque c'est la granularité hebdomadaire et donc l'erreur qui lui est associée qui nous intéresse, nous avons considéré pour une semaine w que y_w représente les ventes de cette semaine. L'erreur de prédiction associée est alors : $\mathcal{E}_{week} = \frac{1}{n_{week}} \sum_w \frac{|\hat{y}_w - y_w|}{y_w}$.

4.4 Modèles prédictifs

Dans les expériences relatées dans la section 4.5, nous comparons quatre modèles prédictifs.

Moyenne annuelle glissante (year mean - YM)

YM est un modèle de prédiction dit «naïf », où la valeur $\widehat{y_w}$ est telle que (avec 52 le nombre de semaines dans une année) :

$$\hat{y}_w = \frac{1}{52} \sum_{j=1}^{52} y_{w-i}$$

Le principe de ce modèle naïf est de proposer une prédiction basée sur les ventes moyennes d'un produit au cours des 52 dernières semaines. Il est actuellement largement utilisé dans le secteur du négoce.

Réseau neuronal récurrent avec mémoire à long terme (RNN LSTM)

Le réseau neuronal est un modèle datant des années 1950 [85]. D'abord négligé en raison du manque de puissance de calcul, ce modèle est devenu populaire depuis les années 1990. Le réseau neuronal récurrent est une variante du réseau neuronal permettant de prendre en compte la séquentialité des données d'entrée, en liant la sortie d'une cellule

du réseau à son entrée suivante. Plus précisément, le LSTM [45] pour « long short-term memory » est un réseau neuronal récurrent dans lequel chaque cellule gère trois signaux supplémentaires : entrée, sortie et oubli. Cette configuration permet de modifier à chaque itération un état lié aux couches cachées du modèle, influençant les prochains signaux passant par le neurone. Dans notre cas, l'optimisation de ce modèle se fait en minimisant la MAE.

Forêts aléatoires (RF)

Les forêts aléatoires [10], bien que principalement utilisées pour résoudre les problèmes de classification, peuvent également être utilisées dans le cadre de la régression. Chaque arbre est généré en tirant n observations de l'ensemble d'apprentissage et en sélectionnant un échantillon du nombre total de descripteurs liés à ces observations. Une fois que ces N tirages et cette sélection de b descripteurs ont été effectués, un arbre de décision est formé pour chaque tirage, dont la croissance est limitée par la définition d'une profondeur maximale. Lors de la prédiction, chaque arbre fournira sa prédiction et la valeur finale sera une moyenne de celles-ci. L'optimisation de ce modèle se fait en minimisant la MAE.

Régression harmonique dynamique basée sur un modèle ARIMAX (DHR)

Les modèles ARMA [9, chapitre 3], dont ARIMAX est dérivé sont basés sur deux processus principaux : l'autorégression, qui signifie qu'une variable X_t peut être expliquée par ses valeurs passées, et une moyenne mobile sur le processus d'erreur de prévision, signifiant que la variable peut être expliquée par l'accumulation d'un bruit blanc : les erreurs des prédictions précédentes. Un modèle ARMA considère donc que X peut être expliqué sur la base des valeurs antérieures de X et des erreurs antérieures. L'optimisation de ce modèle se base sur une minimisation de l'AIC.

Le I de l'acronyme ARIMAX correspond à la nécessité de différencier la série temporelle pour la rendre stationnaire, et le X ajoute au modèle la possibilité d'utiliser les descripteurs présentés précédemment comme termes de la régression. En théorie, un modèle SARIMAX pourrait être utilisé, prenant en compte la saisonnalité d'une série temporelle. Mais en réalité, ce modèle est limitant sur plusieurs aspects : il ne permet pas de prendre en compte plusieurs saisonnalités (hebdomadaire, mensuelle, annuelle). Par exemple : plus de pelles à neige sont vendues en hiver pour des raisons évidentes, notamment en début de semaine car les gens sont surpris par la neige en fin de semaine alors que le magasin est fermé.

Concernant l'implémentation, bien qu'en pratique il soit théoriquement possible d'utiliser une saisonnalité allant jusqu'à 350, le temps d'apprentissage est trop pénalisant et la consommation de mémoire élevée puisque ce paramètre implique une recherche de corrélation de 350 jours entre eux. Dans le contexte d'une prédiction basée sur des saisonnalités multiples, dont certaines sont longues (52 pour une année de semaines de travail par exemple), il est possible de passer d'un modèle SARIMAX à un modèle ARIMAX où les différentes saisonnalités sont extraites et considérées comme des descripteurs du modèle [50, chapitre 9.5]. Nous utiliserons par la suite la dénomination *DHR* pour Dynamic Harmonic Regression plutôt que ARIMAX.

Optimisation paramétrique des modèles prédictifs

Un paramétrage spécifique est nécessaire pour chaque modèle prédictif. Lorsque le nombre de références est faible, ce paramétrage optimal est recherché pour chaque produit. Cependant, il est impossible de dédier un modèle spécifique à chacun des 20 000 produits car le temps de calcul serait prohibitif. Par exemple, un LSTM a besoin de 36 mn pour tester les 162 combinaisons de valeurs, conduisant à 500 $jours$ de temps de calcul pour les 20 000 produits.

Outre la sélection des paramètres du modèle, nous avons effectué une sélection des descripteurs pertinents. Cette sélection est très coûteux car elle implique le test de toutes les combinaisons possibles de paramétrage, menant à 2^n entraînements pour n descripteurs à tester, à effectuer sur les 162 combinaisons de valeurs de paramétrage existant. Nous avons utilisé une sélection de descripteurs basée sur la capacité d'un descripteur à réduire l'entropie dans un modèle de forêt aléatoire. La DHR est l'unique modèle à être amélioré par cette sélection de descripteurs, c'est donc le seul modèle sur lequel cette sélection de descripteurs est appliquée. Les résultats de cette sélection sont détaillés dans l'annexe B.1.

Extraction de descripteurs saisonniers

Pour qu'un modèle prédictif puisse intégrer des variations saisonnières corrélées aux ventes d'un produit, il est nécessaire de réintroduire les causes de ces variations sous forme de descripteurs. Une fois déterminées, ces saisonnalités deviendront donc de nouveaux descripteurs. Le calcul de la saisonnalité commence par l'utilisation d'une fonction d'autocorrélation (ACF) [9, Chapter 2] pour déterminer les décalages temporels avec lesquels la variable cible semble être corrélée. Par exemple, un pic sur l'ACF répété avec un

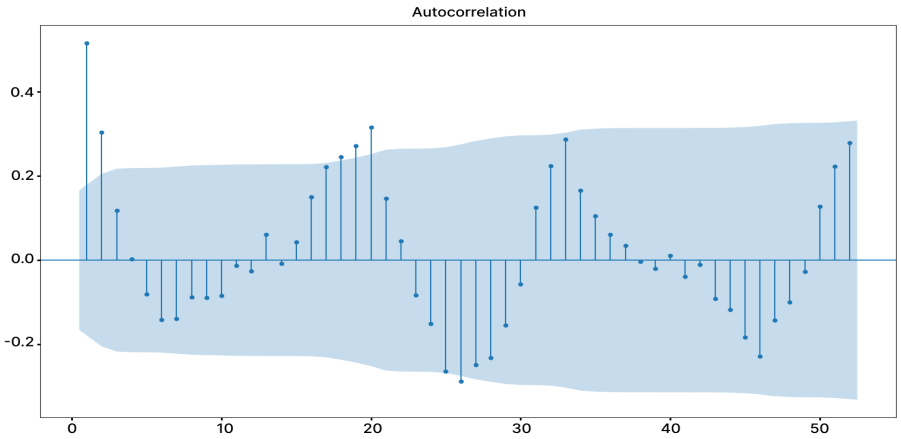

FIGURE 4.1 – ACF du C2-2 avec une périodicité de 52.

décalage de 4 semaines (voir figure 4.1) suggère une saisonnalité mensuelle, un décalage de 12 ou 13 semaines une saisonnalité trimestrielle, et ainsi de suite.

Une fois la saisonnalité prédominante déterminée, une décomposition Seasonality-Trend Loess (STL) [20] de la variable cible est effectuée. La STL sépare la série temporelle en trois composantes additives : une tendance, une saisonnalité et un résidu qui a les propriétés du bruit blanc. La Figure 4.2 montre sur sa partie supérieure la série à décomposer et sur les trois autres graphiques les composantes dont l'addition donne la série originale. La composante saisonnalité est alors extraite, et simplifiée à l'aide d'une transformation de Fourier rapide, d'une réduction du nombre d'harmoniques (filtre passe-bas) et finalement d'une transformation inverse. Une simplification limitant à 5 harmoniques est détaillée Figure 4.3. Cette saisonnalité simplifiée devient finalement une nouvelle série temporelle utilisée comme un descripteur d'entrée pour alimenter le modèle. Cette opération de décomposition en fonction de la saisonnalité prédominante est répétée jusqu'à ce que tous les pics significatifs d'auto-corrélation (dépassant la zone bleutée sur la Figure 4.1) soient pris en compte.

Mise en œuvre

Afin d'expérimenter sur nos données, nous avons réalisé une implémentation en Python utilisant certaines bibliothèques populaires : pandas, pour manipuler l'ensemble de données des produits, sklearn pour les prévisions de la forêt aléatoire, le filtrage d'importance et la mise à l'échelle des données, pmdarima pour les prévisions DHR, tensorflow pour les

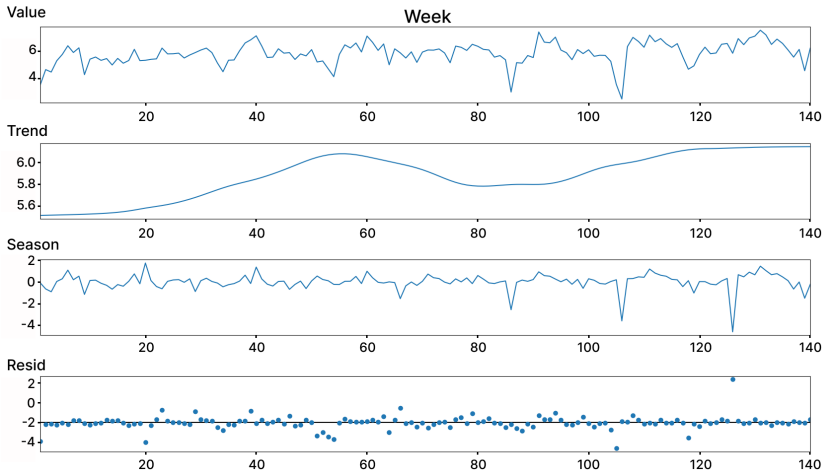

FIGURE 4.2 – Décomposition STL du produit C2-2 avec une périodicité saisonnière de 20.

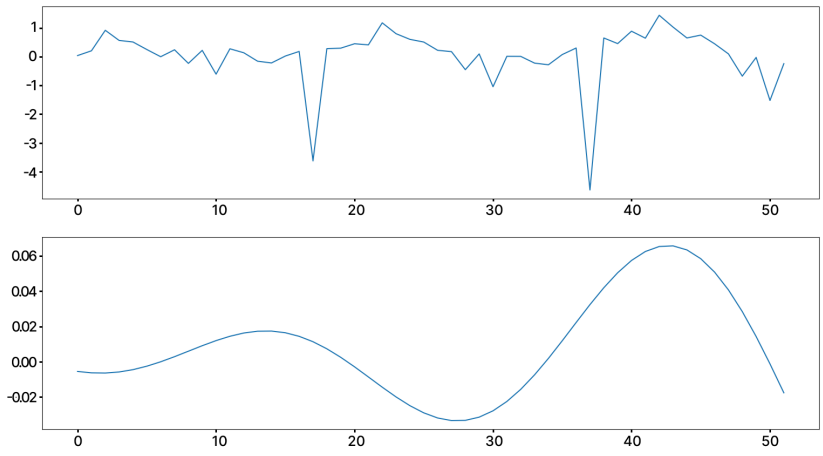

FIGURE 4.3 – Simplification par décomposition STL et réduction de l'ordre de Fourier.

prévisions LSTM, numpy et statsmodels ont été utilisés pour isoler et transformer les descripteurs saisonniers.

4.5 Résultats et discussion

Cette section présente les résultats de nos expériences en matière de prédiction du volume des ventes pour les dix produits les plus vendus. L'intérêt d'ajouter des descripteurs saisonniers est discuté, puis les différents modèles sont comparés. Nous concluons en proposant des scénarios prédictifs combinant les meilleurs modèles.

Intérêt de la décomposition saisonnière

L'intérêt de considérer des descripteurs saisonniers peut être constaté dans les résultats du Tableau 4.3. Celui-ci montre les taux d'erreur hebdomadaires des trois méthodes, avec et sans la décomposition saisonnière. La décomposition saisonnière améliore toujours les résultats. Pour le reste de l'analyse, les descripteurs saisonniers seront inclus dans les modèles, car ils améliorent la prédiction et sont peu coûteux à calculer. Il convient également de noter que le modèle DHR obtient de meilleurs résultats (29.41 % MAPE) que le modèle saisonnier traditionnel SARIMAX (32.65 %). De plus, l'entraînement est plus rapide (15 s pour DHR, 4.6 mn pour SARIMAX) notamment parce que SARIMAX doit déterminer 4 paramètres supplémentaires.

Table 4.3 – Intérêt de la décomposition saisonnière sur l'erreur MAPE (en %).

modèle	sans saisonnalité	avec saisonnalité
DHR	31.45	**29.41**
LSTM	42.78	**34.00**
RF	35.81	**34.45**

Comparaison des modèles

Les modèles DHR, LSTM, RF et YF sont entrainés sur les dix produits les plus vendus. Pour chaque modèle, le tableau 4.4 donne les résultats en fournissant le pourcentage d'erreur absolue moyenne (MAPE) sur ces produits, ainsi que l'erreur d'apprentissage associée. Il est facile de désigner le modèle le mieux adapté aux contraintes imposées :

TABLE 4.4 – MAPE hebdomadaire des différents modèles sur les 10 produits les plus vendus.

product	YM	DHR	LSTM	RF
C2-2	36.23	**26.91**	32.06	31.41
C2-1	30.10	19.43	18.94	**17.26**
C4-1	29.65	25.86	**20.83**	22.63
C1-3	31.83	**25.84**	28.67	30.66
C3-1	26.73	26.27	34.45	**23,44**
C1-4	46.63	**39.65**	50.79	53.64
C1-1	35.40	**17.71**	32.96	26.77
C1-2	56.59	**44.56**	59.14	63.47
C1-5	32.91	38.83	**25.18**	37.07
C1-6	32.74	**29.07**	36.96	38.18
mean	35.88	**29.41**	34.00	34.45
training time	7.3 s	15 s	36 mn	3.4 mn

la DHR. Ce modèle est le plus adapté en raison de la qualité de ses résultats et de son faible temps d'apprentissage. Les résultats du LSTM ne justifient pas son coût d'apprentissage, et le RF, bien que plus rapide, offre les moins bons résultats des trois modèles. La Figure 4.4 détaille ces mêmes résultats et permet de constater que, bien que les valeurs d'erreur fluctuent, le classement relatif reste le même pour chaque modèle (pics et creux).

La bonne performance de la DHR n'est pas surprenante : l'algorithme comprend une recherche exhaustive de la meilleure paramétrisation pour un temps d'apprentissage faible. Il en résulte un modèle entrainé rapidement, bénéficiant d'un paramétrage spécifique à chaque produit. Il est important de garder à l'esprit que, malgré l'inclusion des confinements et de la COVID19 dans les descripteurs, la période d'octobre 2020 fait partie d'une année difficile à prévoir en raison de son caractère exceptionnel. De plus, des problématiques terrain supplémentaires, non prises en compte dans les descripteurs, génèrent de l'incertitude : fiabilité des vendeurs à enregistrer immédiatement leurs ventes, rabais sur les produits, hausse des prix due à la COVID. Finalement, les prévisions hebdomadaires s'avèrent honorables, sept produits sur dix se situent en dessous du seuil d'erreur de 30 %.

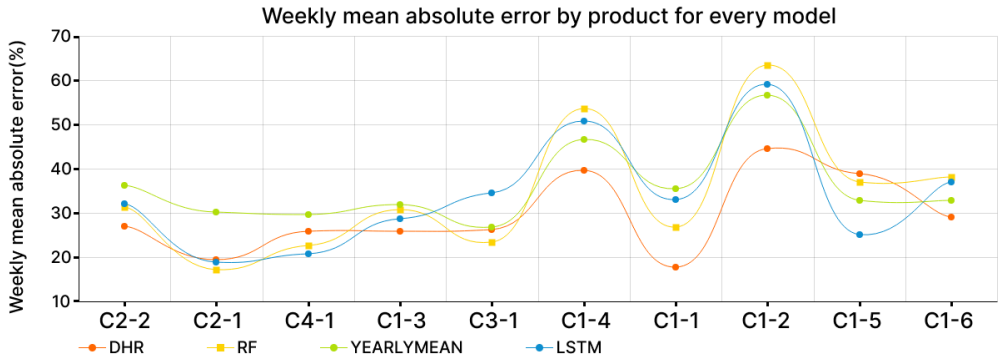

FIGURE 4.4 – Erreur hebdomadaire absolue.

4.6 Scénarios prédictifs

Cette étude nous a permis de déterminer le modèle le plus pertinent pour proposer une prédiction des ventes des dix produits les plus vendus. Nous avons généralisé ces expérimentations pour les 100 produits les plus vendus : les résultats moyens sont consignés dans le tableau 4.5, à côté du classement constaté pour chaque scénario. Par exemple, on peut lire que le scénario qui consiste à n'utiliser que des YM arrive premier 312 fois sur 1000 et dernier 492 fois. Pour chaque évaluation et classement des modèles, ce sont les moyennes des modèles sur 10 produits tirés au hasard qui sont évaluées, la moyenne consignée est la moyenne de ces multiples évaluations. Même si certains produits tirent vers le haut l'erreur hebdomadaire (un produit a notamment atteint une erreur de 1 000 %), le modèle de prédiction YearlyMean devient le meilleur modèle lorsque les volumes de vente diminuent. Cela signifie que le choix du meilleur modèle de prévision pour un produit pourrait être lié au nombre de ventes, ainsi qu'à d'autres descripteurs de la série temporelle.

TABLE 4.5 – Erreur hebdomadaire moyenne et nombre de fois où chaque scénario a été classé premier, second... Pour les 100 essais des 100 produits

modèle	erreur hebdomadaire	1er	2e	3e	4e
YM	61.91 %	312	346	311	31
DHR	69.83 %	115	122	358	405
LSTM	65.81 %	127	178	171	524
scenario	56.90 %	492	318	152	38

Afin de confronter cette théorie, nous proposons de construire deux arbres de décision.

Comme la YM est le modèle le plus rapide et le plus efficace, on commence par construire un arbre (YM-TREE) capable de déterminer si la YM est un modèle fiable pour un produit spécifique. On entraine ensuite un second arbre (LSTMDHR-TREE) pour déterminer un modèle viable entre LSTM et DHR, ou si les deux processus proposent une solution acceptable. La Figure 4.5 résume le processus menant au modèle scénario.

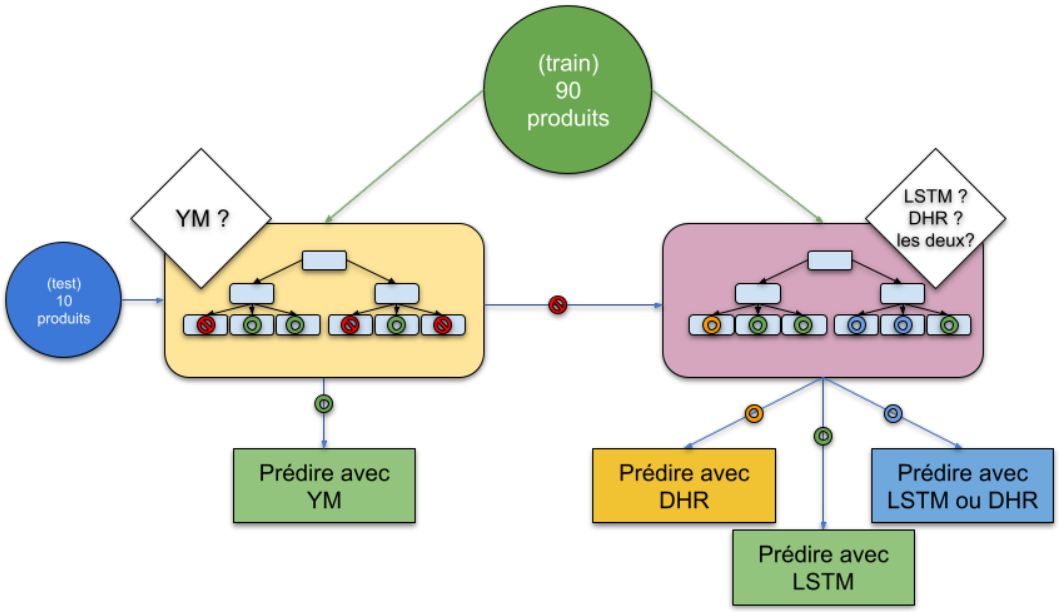

FIGURE 4.5 – Schéma du scénario prédictif basé sur des arbres de décision.

Pour construire ces deux arbres, on utilise la prédiction sur 100 produits de YM, DHR et LSTM de la manière suivante : un modèle est considéré comme viable pour prédire un produit spécifique si son erreur se situe dans une fourchette de 5 % avec la meilleure erreur de prédiction pour ce produit. Par exemple, notre produit C1-3 obtient un score de 31.83 % (YM), 25.84 % (DHR) et 28.67 % (LSTM), ce qui signifie que les modèles acceptables pour ce produit sont la DHR et le LSTM. On construit ensuite le YM-TREE, classifieur ciblant la présence de YM dans les modèles acceptables, en créant une variable binaire pour prédire l'utilisation pertinente de YM. Pour construire ces arbres, on utilise des descripteurs peu nombreux et faciles à calculer : écart-type relatif, quantité totale de ventes, nombre de ventes et les résultats de DHR et YM sur l'ensemble d'évaluation. Nous avons suivi le même processus pour le LSTMDHR-TREE, sauf que la tâche de classification contient 3 classes : LSTM viable, DHR viable, LSTM et DHR viables. Une fois ces deux arbres construits, on vérifie pour chaque produit si la

YM peut fonctionner. Dans l'affirmative, la YM est utilisée pour prédire les ventes. Si ce n'est pas le cas, LSTMDHR-TREE est utilisé pour déterminer quel modèle conviendrait le mieux entre LSTM et DHR (si les deux sont acceptables, DHR est préféré car il est plus rapide). Afin d'évaluer ce processus, 10 produits sont choisis au hasard, retirés du pool de prédiction et prédits à l'aide des 90 autres. L'ensemble du processus d'entraînement des deux arbres est résumé figure 4.5.

Ce processus itère sur les 10 produits suivants de l'ensemble des produits, et prédit en utilisant les 90 autres, jusqu'à ce que tous les produits soient prédits une fois. Ce processus, appelé validation par sous-échantillonnage aléatoire [13], est répété 100 fois. Pour simplifier, le modèle YMTREE+LSTMDHR-TREE sera appelé *modèle de scénario* dans la suite des explications. Les résultats du modèle scénario sont présentés dans le tableau 4.5 ainsi que son classement vis-à-vis de toutes les politiques de prédiction utilisant un seul modèle.

L'utilisation du modèle de scénario entraîne une amélioration significative des prédictions, d'au moins 5 % par rapport à n'importe quel modèle, tout en montrant que près de la moitié du temps, le scénario propose la meilleure prédiction moyenne sur les 10 produits aléatoires choisis à chaque itération. Au cours des 10 000 prédictions (prédiction sur 100 produits, 100 fois), nous avons observé qu'en moyenne 70 % des modèles utilisés sont des YM, 20 % des DHR et 10 % des LSTM. Le temps de calcul sur 10 produits est donc estimé à 39.4 *mn* pour le scénario, plus court que les 360 *mn* pour entraîner 10 LSTM, bien plus long cependant que l'exécution de 10 YM.

4.7 Conclusion

Ce chapitre a montré une comparaison pratique de quatre modèles classiques de prévision, dans le contexte spécifique de la prédiction des ventes de matériaux. Au-delà des performances de prédiction, nous avons porté une attention particulière aux temps d'apprentissage ainsi qu'à la sélection des descripteurs. Nous concluons que l'utilisation de la saisonnalité comme descripteur est efficace : bien qu'elle augmente légèrement le temps de calcul, elle prend en compte les particularités de chaque produit.

Notre contribution à cette problématique consiste en la proposition d'un modèle de scénario adapté choisissant automatiquement entre YM, DHR et LSTM. Cette approche pourrait être qualifiée de *pseudo meta-learning*. La prédiction moyenne de ce système de prédiction entièrement automatisé sur de nombreux produits est améliorée et le temps

d'apprentissage raccourci. Ces résultats ont été publiés à la conférence internationale SMARTCOMP2022, tenue à Helsinki [97].

L'amélioration de ce travail est toujours d'actualité. Bien que l'expérimentation courante décrive les résultats obtenus sur 100 produits, un problème de prévision doit en pratique traiter jusqu'à 20 000 articles par commerçant. Il existe potentiellement de fortes corrélations entre les volumes de vente des différents articles, qui ne sont pas encore exploitées dans le processus actuel. L'ajout de descripteurs avancés pour décrire la série temporelle comme le kurtosis, la skewness [3] ou la silhouette [59] pourraient améliorer la classification, de même que l'utilisation pour le scénario prédictif de classifieurs plus avancés qu'un simple arbre de décision, par exemple des forêts aléatoires. L'ajout de méthodes simples pour remplacer la YearlyMean par des méthodes plus efficaces pour un coût similaire est également à l'étude, notamment les méthodes de lissage exponentiel simple, double et triple [50, chapitre 7] qui permettent respectivement d'effectuer la prédiction d'un niveau, puis d'y ajouter une tendance, et finalement une saisonnalité.

Chapitre 5

Un réapprovisionnement adapté aux négoces

Si la prédiction du besoin est un point clé, c'est parce qu'elle permet de réduire les coûts grâce à une meilleure maîtrise du réapprovisionnement, du stockage et de la distribution. Dans ce chapitre, on s'intéressera désormais à la fois aux notions classiques qui gravitent autour du réapprovisionnement dans la chaîne de distribution, mais aussi aux variations de ces notions qui caractérisent le négoce de matériaux. Une fois le problème posé, nous proposons la résolution de multiples instances de réapprovisionnements à l'aide de différents algorithmes. La comparaison de ces approches permettra de proposer une résolution du problème selon le temps disponible, une ressource critique et particulièrement difficile à maîtriser dans le cadre d'une optimisation ciblant un nombre très important de produits.

5.1 Introduction

À cause de références produit variées, de multiples dépôts et nombreux fournisseurs, l'espace de recherche pour l'optimisation du problème du réapprovisionnement est extrêmement vaste. Les techniques actuelles ne permettent pas de garantir l'optimalité dans un temps limité pour des problèmes de cette complexité.

5.1.1 Digest

Cette section propose une vue d'ensemble de la contribution proposée tout au long de ce chapitre. Elle démarre avec le contexte et la formalisation du problème qui en découle, suivi d'un aperçu général des méthodes de résolution mises en place avant de conclure par une synthèse mêlant le résultat de l'expérimentation et la réflexion qu'elle entraîne.

Contexte

Le contexte du négoce de matériaux est proche des pratiques classiques de la chaîne de distribution, c'est un fonctionnement que la plupart des commerçants et industriels connaissent bien. Dans la chaîne de distribution générale, un équilibre doit être trouvé entre le coût de stockage, le coût de livraison et la demande à fournir en produit. Pour que l'optimisation soit pertinente, la prédiction de la demande doit être au plus proche de la réalité. On ajoutera le plus souvent un stock de sécurité, le plus faible possible, permettant de s'assurer qu'en cas d'erreur dans la prédiction, il sera toujours possible de fournir la demande le temps de se réapprovisionner.

Dans le négoce de matériaux, cet équilibre entre coûts de stockage et coût de livraison n'est pas aussi capital que dans d'autres industries, car les responsables n'estiment pas les coûts de stockage : ils détiennent le plus souvent les entrepôts. L'équilibre est même rompu puisque les négociants, ne souhaitant pas payer de frais de transports, sont prêts à commander des produits non nécessaires pour atteindre la condition de livraison gratuite. Si l'on considère que les coûts de stockage sont nuls, cette logique se tient : il vaut toujours mieux commander des produits de construction, qui finiront par être vendus, plutôt que payer des frais de transport. Bien sûr, commander si possible des produits qui seront nécessaires à moyen terme serait plus optimal pour compléter cette commande. C'est sur cet aspect que notre modélisation se détache de la rationalisation proposée par des résolutions plus classiques du DJRP.

Modèle et implémentation

Pour mettre en place notre expérimentation, nous avons utilisé quatre algorithmes de résolution classiques présentés plus en détail section 5.3.2 : un solveur, basé sur la formulation mathématique du problème, est notamment utilisé comme résultat de référence. La modélisation du problème vise à prendre en compte les particularités du négoce, comme vérifier que le transport gratuit est atteint, mais aussi des contraintes implicites comme l'impossibilité de commander une quantité négative d'un produit. La modélisation tient compte des périodes de rupture liées aux durées de livraison et du coût de commande total permettant de combler tous les besoins à commander. Dans le même temps, la condition de livraison gratuite doit être atteinte en complétant la commande avec des produits qui sont considérés comme intéressants pour le futur. Cette modélisation est approfondie section 5.2.

Le solveur basé sur la formulation mathématique du problème atteint l'optimalité, à condition qu'il termine en un temps raisonnable. Pour les problèmes très complexes, ce temps n'est pas raisonnable dans le cadre d'une utilisation professionnelle. Une réponse aux alentours de la vingtaine de minutes et au maximum de l'heure pour les problèmes complexes est attendue.

Trois algorithmes heuristiques sont alors implémentés pour résoudre le problème : l'algorithme génétique, le recuit simulé et un algorithme glouton adapté de l'algorithme glouton du problème de sac à dos. Ces trois algorithmes sont adaptés pour tenir compte des spécificités du négoce, et une fonction d'évaluation générale permet de les comparer entre eux. Pour tenir compte de la contrainte de temps, ces algorithmes sont arrêtés au bout de 20 minutes d'exécution, et le meilleur résultat rencontré lors de l'exécution est retourné lors de cet arrêt.

En plus de ces quatre algorithmes mis en compétition, des chaînes de résolution sont comparées. Elles enchaînent de deux à quatre des algorithmes de résolution abordés : un premier algorithme calcule une solution qui est affinée par l'algorithme suivant. Le fonctionnement de ces chaînes est détaillé section 5.3.3.

Des instances synthétiques de difficultés variables sont générées pour comparer ces algorithmes, cette difficulté est notamment influencée par trois paramètres : le nombre de produits et de fournisseurs concernés, et la possibilité ou non pour ces derniers de fragmenter l'envoi des produits de la commande. La génération de ces instances et la classification de celles-ci en instance facile ou difficile est approfondie section 5.3.1.

Résultats

Entre les quatre algorithmes de résolution présentés et les différentes chaînes de résolution qui en découlent, 25 approches de résolutions sont comparées. La comparaison s'axe autant sur le résultat que sur le temps d'exécution requis pour l'atteindre. La comparaison est divisée en deux parties selon que l'instance est facile ou difficile. Trois points clé ressortent de cette étude :

— dans une chaîne de résolution, améliorer la solution de l'algorithme glouton par une autre méthode bonifie toujours le résultat, pour un coût d'exécution très faible. Ces résultats sont consignés dans les tableaux 5.9 et 5.11 respectivement en section 5.4.1 et 5.4.2 ;

— l'utilisation d'une chaîne de résolution permet de développer des stratégies d'arrêt précoce lorsque l'amélioration du résultat par un algorithme stagne, pour passer

plus rapidement au suivant en limitant la dégradation du résultat final. Ainsi, l'exécution d'une chaîne de résolution composée d'un l'algorithme glouton suivi par le recuit simulé puis de l'algorithme génétique est plus rapide que l'exécution individuelle de ces trois modèles. Les tableaux complets de résultats proposés en Annexe C et les tableaux 5.8 et 5.10 des sections 5.4.1 et 5.4.2 montrent que cet intérêt varie en fonction de la complexité du problème et du nombre de méthodes de résolution enchaînées ;

— les algorithmes les plus intéressants diffèrent peu entre les instances faciles et difficiles. On y retrouve principalement les deux chaînes de résolution suivantes : glouton suivi de recuit simulé, et génétique suivi de glouton et de recuit simulé. Le tableau 5.9 de la section récapitule les meilleurs modèles de résolution en fonction de la complexité du problème et de la contrainte de temps imposée, avec et sans l'utilisation d'un solveur visant l'optimalité.

La contribution proposée est donc composée d'une modélisation adaptée aux spécificités du négoce de matériaux, de la démonstration de l'intérêt des chaînes de résolution et des méthodes d'arrêts précoces lorsque le contexte d'utilisation limite temporellement l'exécution, et de conseils concrets résultant de nos expérimentations sur les chaînes de résolution les plus adaptées.

5.1.2 L'approvisionnement : un problème multi-objectif

Cette section présente une liste des principaux critères devant être pris en compte dans l'optique de proposer une optimisation du réapprovisionnement.

Prix d'achat unitaire d'un produit Le prix unitaire des produits étant prépondérant dans le coût de commande à un fournisseur, il oriente le revendeur lors de son choix de commerçant pour passer commande [67]. Cependant, bien que le coût d'achat représente la dépense la plus élevée, il est considéré comme un coût *mineur* puisque c'est l'achat du produit au fournisseur qui permet de le vendre au client, générant le bénéfice de l'entreprise. Dans la mesure où l'achat est considéré comme inévitable, c'est la variabilité de prix, souvent faible, entre les fournisseurs qui est prise en compte plutôt que le coût total.

Délais de livraison fournisseur Le délai de livraison est un autre critère qui orientera le choix du revendeur [48], en fonction de la tension sur son stock au moment de passer

commande. Effectivement, l'importance du délai fournisseur pourra varier en fonction de la gravité d'une rupture produit. Si certains commerces comme le luxe maintiennent une demande et des prix élevés grâce à ces ruptures, ce n'est généralement pas le cas pour les commerces plus classiques comme l'alimentaire ou le textile, ce manque à gagner influant directement sur les bénéfices réalisés, mais aussi sur les habitudes consommateurs qui pourraient se tourner vers la concurrence si celle-ci a encore du stock [78].

Coûts de stockage Le coût de stockage correspond au prix estimé pour maintenir le produit à disposition d'un acheteur tant qu'il n'est pas vendu. Ce coût va varier en fonction de la nature du produit stocké : taille, poids et vitesse de détérioration [87]. Outre le coût direct lié à la location d'un entrepôt, ou le fonctionnement d'une chambre froide, on y associe des coûts supplémentaires comme l'assurance sur le stock et l'immobilisation du capital de l'entreprise, qui l'empêche d'investir son argent autrement [117]. Ce coût est considéré comme *majeur* car il réduit chaque jour les bénéfices réalisés suite à la vente du produit stocké. Dans certains cas, il peut même mener à la destruction pure et simple du produit concerné pour limiter les pertes parce qu'il n'est plus utilisable ou consommable, comme c'est parfois le cas pour les produits frais en grande surface [103].

Frais de transport Le frais de transport représente l'autre coût majeur de la commande passée à un fournisseur. Contrairement au prix d'achat, qui est inévitable, les frais de livraisons peuvent souvent être réduits ou annulés lorsque la commande atteint un certain seuil de prix ou de volume. Cette réduction motive les revendeurs à commander des volumes importants, ou divers produits en une seule fois pour ne pas payer plusieurs fois ces frais. Le prix de livraison et les conditions pour ne pas le payer peuvent varier entre les industries concernées. Par exemple, la livraison alimentaire a un coût élevé, car elle implique un conditionnement particulier des denrées, tandis que le transport de textile est moins contraint par ces problématiques [12].

Remises supplémentaires Des remises sur le prix d'achat des produits peuvent être pratiquées par certains fournisseurs [68]. Le prix unitaire d'un produit peut varier en fonction du nombre d'unités du produit acheté. Des bonus de fin d'année peuvent également s'appliquer : il s'agit de contrats spécifiques garantissant des remises sur les produits l'année suivante si un seuil de coût de commande est atteint l'année précédente [106]. Les remises peuvent concerner des produits différents formant un lot, comme une remise importante sur la colle adéquate lors de la commande de carrelage chez un fournisseur.

À l'équilibre entre flux tendu et sur-stockage : optimisation Le plus souvent dans l'optimisation du réapprovisionnement, c'est l'équilibre entre stockage, prix d'achat et prix de vente qui est recherché. Comme nous venons de le détailler, les réductions offertes par le fournisseur sont le plus souvent conditionnées par des volumes d'achat importants. Pourtant, commander des volumes importants implique des risques liés au stockage, et la bonne affaire faite grâce à l'achat en volume peut rapidement se transformer en perte sèche. Les deux comportements extrêmes consistent à acheter des volumes très élevés pour bénéficier de remises maximales, quitte à payer des frais de ports, ou à l'inverse, se réapprovisionner en flux tendu en fonction de la demande estimée pour réduire au minimum le stockage nécessaire, impliquant des coûts à l'achat plus importants et un risque de rupture plus élevé [72]. Bien sûr, c'est pour trouver le juste milieu entre ces coûts que des méthodes d'optimisation comme *Material Requirements Planning* [15] et *Manufacturing Ressources Planning* [52] existent.

5.1.3 Particularités des coûts dans le négoce de matériaux

Cette section détaille les spécificités des coûts présentés précédemment vis-à-vis du négoce de matériaux. Ces spécificités seront prises en compte dans la modélisation des problèmes de réapprovisionnement joint rencontrés ultérieurement.

Le délai produit dépend de la commande totale Si un délai de livraison est indiqué par produit, ce délai peut également varier en fonction des produits commandés. Certains fournisseurs pratiquent l'envoi séparé tandis que d'autres enverront la commande une fois tous les produits disponibles. En fonction du fournisseur, le produit sera livré soit selon son délai indiqué, soit selon le délai du produit le plus lent de la commande.

Éviter la rupture de stock à tout prix La rupture de stock survient après un retard de livraison ou l'impossibilité de se réapprovisionner dans les temps à cause d'un problème logistique, d'une demande trop forte ou d'un oubli. Cette rupture génère un premier coût correspondant à la marge de la vente perdue. Puis un second, plus difficile à évaluer : la perte de confiance ou d'habitude du client, qui devra se rendre chez la concurrence pour obtenir son produit. Le négoce ciblant avant tout des professionnels, la permutabilité entre des produits similaires n'est pas aussi évidente que pour des produits de la grande consommation, comme choisir une autre bouteille de lait. Si l'artisan a l'habitude de travailler avec certains produits, il n'en changera qu'en cas d'absolue nécessité. La maîtrise

de ces ruptures de stock est donc déterminante dans le rapport avec la concurrence, raison pour laquelle le négociant, en cas de besoin urgent, aura tendance à préférer un coût de livraison élevé à une rupture temporaire. Car si le coût de cette livraison rapide est quantifiable, ce n'est pas le cas du coût d'une rupture produit.

Coûts de stockage Pour éviter les coûts liés à la rupture de stock, les négociants de matériaux stockent plus que de raison. Dans le négoce de matériaux, les coûts de stocks ne sont pas estimés : le plus souvent, les entrepôts ne sont pas loués, mais appartiennent aux négociants, ils n'ont donc pas de coûts de stockage au sens premier du terme. Malgré cela, ces stocks impliquent des coûts d'assurance, de mobilisation de trésorerie, de manutention et il est critique de veiller à limiter le sur-stock pour éviter qu'un entrepôt ne soit restreint dans son stockage par des produits qui ne se vendent pas ou peu. Ces produits empiètent alors sur l'espace disponible pour les produits clé, qui nécessitent souvent un stockage important, mais disposent d'un bon taux de rotation.

Prix de la livraison Les négociants se refusant à payer des coûts de livraison, la condition de livraison gratuite doit donc toujours être atteinte, quitte à commander des produits qui ne seront utiles qu'à très long terme. Ce choix est justifié par l'absence d'évaluation des coûts de stockage, mais aussi par la longue conservation des produits commandés, qui ne se détériorent pas dans le temps tant qu'ils sont stockés au sec. Effectivement, si l'on considère que les coûts de stockage sont nuls, mieux vaut commander des produits qui ne servent à rien que de payer l'équivalent de ces produits en frais de port, menant à une perte sèche. Dans la pratique cependant, le sur-stockage complique la gestion des entrepôts.

Remises supplémentaires Bien que tous les types de remises présentés à la section précédente existent dans le négoce, c'est la remise quantitative sur les produits qui est prépondérante, c'est-à-dire que le prix unitaire évolue en fonction de la quantité commandée, le plus souvent par une fonction en escalier décroissante. Ces remises incitent les négociants à commander des volumes importants pour pouvoir proposer des tarifs faibles sur certains produits d'appel qui attirent les artisans spécialistes. Ces artisans sont particulièrement attentifs aux tarifs de certains produits, qui représentent une partie élevée de leurs coûts de fonctionnement sur les chantiers.

L'équilibre du coût, du bénéfice et du volume

Pour augmenter les bénéfices, deux méthodes prévalent : réduire les coûts ou faire varier le prix pour vendre plus cher ou plus de volume, le choix pouvant varier en fonction de plusieurs facteurs comme la concurrence et le produit concerné. Il est en effet rationnel de considérer que la réduction des prix s'accompagnera d'une meilleure compétitivité des négociants sur leurs marchés concurrentiels, et donc que le volume des ventes augmentera, tout comme les coûts de gestion. À l'inverse, une augmentation du prix pourrait mener à plus de bénéfice à condition que la concurrence soit faible, et que le produit soit indispensable. Dans la mesure où il s'agit avant tout d'un milieu professionnel d'artisans qui ont besoin des matériaux eux-mêmes pour travailler, on peut considérer les produits comme indispensables, cela signifie que c'est surtout la concurrence qui limite la baisse de prix. C'est le tarif proposé qui influence le plus l'achat du client [119, 33, 116], puisque ce prix joue sur ses bénéfices personnels ou sa compétitivité dans les appels d'offres. Il serait alors nécessaire d'effectuer pour chaque dépôt une étude de marché pour évaluer la concurrence et calculer des élasticités tarifaires pour déterminer les prix les plus rentables pour chaque produit. C'est donc plutôt l'autre aspect qui nous intéressera ici : la baisse des coûts. On considère l'augmentation des volumes de ventes comme subordonnée à la baisse du prix, elle-même subordonnée à la réduction des coûts. De plus, même sans modification de la stratégie tarifaire, la baisse des coûts augmente directement le bénéfice. La prédiction des ventes présentée au chapitre 4 va dans le sens de la baisse des coûts, puisqu'elle permet, en maîtrisant la demande, de réduire les coûts de stockage et de manutention.

5.1.4 Synthèse

Dans les grandes lignes, le négoce de matériaux est soumis aux problématiques classiques de la chaîne d'approvisionnement et de distribution. Cependant, certaines nuances clé doivent être prises en compte pour élaborer une solution de réapprovisionnement adaptée. Les deux principales différences concernent les coûts de stock qui ne sont pas évalués, la livraison gratuite devant être atteinte à tout prix. Un autre point important, mais moins spécifique aux négoces est le refus catégorique de la rupture de stock, quitte à se réapprovisionner pour beaucoup plus cher si cela permet de la limiter dans le temps ou de l'éviter. Normalement, ces trois facteurs sont soumis à des évaluations de coûts qui permettent de déterminer le choix le plus rentable entre stockage, livraison fréquente et rupture de stock à un instant précis. Le négoce de matériaux s'appuie sur des postulats

empiriques avec lesquels il faut composer pour proposer une solution adéquate.

5.1.5 Le fonctionnement actuel et ses limites

Des indicateurs limités Les négociants s'appuient principalement sur des indicateurs empiriques pour effectuer leurs réapprovisionnements : diversification du portefeuille (60 % de produits principaux bien connus, 40 % de produits plus saisonniers, spécifiques ou plus risqués) [82], un classement ABC pour identifier les différentes catégories de produits [86] ou une analyse de la rotation des stocks [4]. Ces indicateurs sont principalement utilisés pour fournir un contexte stratégique sur les produits à prioriser et surveiller plutôt que des informations opérationnelles sur les produits à commander.

Humain et machine, expertise ciblée et volume Même avec des experts, il n'est pas réaliste de traiter manuellement jusqu'à 20 000 produits provenant de 500 fournisseurs, chacun proposant en moyenne 5 000 produits dans son catalogue. Les utilisateurs finissent par commander le produit nécessaire auprès du même fournisseur et par remplir la commande avec quelques produits qu'ils sont certains de vendre, même à long terme.

Il existe certainement de meilleures alternatives, comme un produit moins cher disponible auprès d'un autre fournisseur ou des conditions d'expédition gratuites plus faciles à atteindre.

Estimation de l'intérêt d'un produit pour compléter la commande Pour compléter une commande et atteindre le transport gratuit, le comportement des négociants est simple : il suffit de compléter la commande avec des produits clés, dont la stabilité des ventes est certaine dans le temps, c'est-à-dire le plus souvent les produits dont le rang de Pareto est A. Si à première vue cette décision semble rationnelle, elle est en réalité peu optimale. En effet, les commandes devraient au contraire être complétées avec les produits dont les ventes sont faibles, mais qui tomberont en rupture à moyen terme, impliquant des commandes pratiquement vides, tandis que le réapprovisionnement des produits de rang A atteint facilement le franco de port. Bien sûr, cette notion n'est pas étrangère aux négociants, mais il est impossible d'avoir en tête les consommations approximatives de toutes les références disponibles. Les négociants connaissent le plus souvent mieux leurs produits principaux, les rangs A.

5.1.6 Intuition sur l'intérêt d'un produit dans la complétion d'une commande

Actuellement, les négociants de matériaux n'effectuent une commande que si le seuil de prix minimal pour ne pas payer de transport est atteint. Ainsi, un questionnement clé apparaît rapidement pour effectuer des réapprovisionnements intelligents : avec quels produits compléter la commande ?

À l'aide d'un système prédictif, il devient possible de se baser sur ces prévisions de vente pour déduire le besoin de réapprovisionnement, même des plus petits produits. Ainsi, on peut pondérer l'intérêt de compléter la commande avec un produit en fonction de la proximité de sa rupture de stock : un produit qui ne tombe pas en rupture sur un horizon important est un produit qui n'a aucune valeur, puisqu'il ne sera pas vendu. On s'assure alors que la gratuité de la livraison sera atteinte à l'aide d'une complétion de la commande basée sur des produits qui connaîtront des ruptures à moyen-long terme, quand le réapprovisionnement, lui, est déclenché par des ruptures à court terme. Ce fonctionnement glissant permet de limiter les commandes, et donc le stock puisque le coût de commande est transféré en coût de stock chez les négociants de matériaux. Bien sûr, d'autres critères viennent s'ajouter pour choisir les produits complémentaires, comme la possible augmentation du délai de livraison de la commande si des produits sont plus longs à fournir, ainsi que le prix du produit en lui-même.

5.1.7 Une optimisation multi-critère contrainte par le temps

Dans la pratique, un négociant planifie régulièrement son réapprovisionnement. Il est donc impossible de s'appuyer sur des processus de calcul longs et il faut alors envisager des méthodes qui fonctionnent avec un nombre limité de ressources temporelles. Les optimisations sont donc **nécessaires en temps réel pour un utilisateur qui a besoin de réapprovisionner des produits au quotidien.** De plus, chaque détaillant possède plusieurs entrepôts, chacun nécessitant sa propre optimisation du réapprovisionnement pour suggérer des commandes. Pour qu'un tel processus soit utilisable au quotidien comme aide à la décision, la méthodologie doit produire une solution suffisamment fiable, dans un temps de calcul acceptable. La durée d'exécution «acceptable» est fixée à 20 minutes dans notre étude expérimentale.

5.1.8 Positionnement

Il ressort de ce préambule sur l'optimisation de la distribution dans le négoce que la complexité de l'espace à explorer pour trouver la solution optimale est excessivement élevée : la difficulté principale réside dans le nombre de produits et de fournisseurs. Les optimisations de délai, de seuil de frais de port et autres remises condamnent définitivement la recherche de l'optimum, justifiant l'intérêt pour les méthodes heuristiques. Nous montrerons dans la section suivante que l'utilisation de chaînes de résolution basées sur plusieurs algorithmes, couplées à l'heuristique de Fogarty et Barringer [30] (voir Section 3.2.3 page 28) pour réduire la complexité temporelle, permet de répondre aux contraintes inhérentes au métier en garantissant une bonne solution dans une durée d'exécution maîtrisée. Cette bonne solution n'a pas vocation à être validée automatiquement, mais plutôt soumise à un expert du réapprovisionnement qui pourra l'adapter avant de la valider.

5.2 Résolution d'instances de réapprovisionnement

Cette section détaille la modélisation que nous avons choisie pour résoudre notre problème de DJRP, qui inclut des remises sur le volume et des livraisons gratuites. La relaxation du problème et les fonctions d'évaluation utilisées pour comparer les solutions des différents algorithmes de résolution sont également abordées.

5.2.1 Formalisation du problème

Étant donné un horizon temporel, des besoins produit et les délais de livraison des fournisseurs, une première méthode consiste à transposer les contraintes dans un modèle d'optimisation linéaire pour déterminer quel produit commander auprès de quel fournisseur ; une relaxation Lagrangienne [29] peut être ajoutée pour pénaliser tout choix qui conduit à une pénurie de produit.

Dans ce modèle, les coûts dépendent principalement de trois facteurs : les prix d'achat, les frais d'expédition et les pénalités résultant des pénuries de produits. Notre travail suppose que les prix d'achat et les pénalités de pénurie sont connus et se concentre sur l'optimisation des coûts d'expédition tout en tenant compte des contraintes qui favorisent la livraison gratuite[13]. Le modèle doit également prendre en compte les remises quantitatives : le prix unitaire est dégressif en fonction de seuils de quantité spécifiques à chaque produit.

5.2.2 Modélisation du problème

Les notations, les constantes, les variables et les contraintes

Trois ensembles d'entiers sont utilisés : I pour les produits, J pour les fournisseurs, K pour les niveaux de remise. Pour indicer par les variables, nous utiliserons $i \in I$, $j \in J$ et $k \in K$. Par exemple, si l'on note V le tableau de constante qui contient les prix des produits, pour accéder au prix du produit i, commandé chez le fournisseur j au niveau de tarif k on notera : $V_{i,j,k}$. Les constantes $|I|, |J|, |K|$ correspondent respectivement au nombre de produits dans I, au nombre de fournisseurs dans J et finalement au nombre de paliers disponibles dans K.

Les *constantes* correspondent à des valeurs contextuelles du négoce : prix de vente, délais fournisseurs, présence ou non d'un produit dans un catalogue fournisseur... Ces valeurs sont immuables car le but de l'optimisation est justement de proposer des valeurs aux variables décidables qui prennent en compte le contexte fourni par les constantes. Un tableau exhaustif des constantes est donné tableau 5.1.

L'ensemble des *variables décidables* après optimisation correspond à la solution proposée par le modèle. Dans notre cas, c'est la variable $x_{i,j}$ qui joue ce rôle. Elle indique la quantité de produit i à commander chez le fournisseur j.

Des variables décidables dépendantes de $x_{i,j}$ sont utilisées. Ces variables sont utilisées pour impacter la modélisation du problème en fonction des modifications de la variable décidable initiale. Par exemple, θ_j est utilisé pour décrire l'implication ou non d'un fournisseur dans une solution, si au moins un produit est commandé chez le fournisseur j, $\theta_j = 1$, sinon, $\theta_j = 0$. Cette valeur de θ_j est ensuite utilisée pour déterminer le coût de la livraison : même si le seuil de livraison gratuit n'est pas atteint, puisque rien n'a été commandé chez le fournisseur j, aucun frais de port n'est à ajouter au coût final. La liste exhaustive des variables décidables est donnée tableau 5.2.

Pour décrire à un solveur une problématique, il faut *contraindre* le problème étudié pour s'assurer que le solveur propose des solutions de réapprovisionnement compatibles avec la réalité. Par exemple, une solution qui mènerait à dépasser le stockage maximal d'un produit est interdite, de même que commander une quantité négative de produit est impossible.

TABLE 5.1 – Constantes utilisées dans le modèle.

notation	*description*
C	constante plus grande que n'importe quelle quantité de produit considérée
constantes associées au produit i	
B_i	quantité nécessaire
M_i	quantité maximale que le négociant peut détenir
R_i	temps disponible avant pénurie
So_i	inventaire initial
constantes associées au fournisseur j	
W_j	seuil de livraison gratuite
F_j	possibilité d'envoyer la commande partiellement
constantes associées au produit i et au fournisseur j	
$D_{i,j}$	délai de livraison du produit
$O_{i,j}$	disponibilité du catalogue
constantes associées au produit i, au fournisseur j et au niveau de remise k	
$V_{i,j,k}$	prix unitaire
$T_{i,j,k}$	quantité minimale à commander pour bénéficier du niveau de remise k
constantes associées à la valeur du produit i	
Pr_i	Valeur de classement ABC
$CTMi$	proximité avec la quantité maximale de i que le négociant détient et commande

TABLE 5.2 – Variables du modèle.

notation	*description*
$x_{i,j}$	quantité de produit i commandée au fournisseur j
$x_{i,\cdot} = (x_{i,0}, \ldots, x_{i,J-1})$	
$lt_{i,j}$	délai de livraison d'un produit i dans une commande spécifique d'un fournisseur j
variables binaires	
θ_j	implication du fournisseur j dans la solution
ω_j	condition de livraison gratuite pour le fournisseur j atteinte
$\sigma_{i,j}$	le produit i est commandé au fournisseur j
$\rho_{i,j}$	le produit i sera livré à temps s'il est commandé au fournisseur j
$\delta_{i,j,k}$	activation du niveau de prix k pour le produit i auprès du fournisseur j

5.2.3 Fonction à optimiser et relaxation du problème

Le principe du solveur est de minimiser ou maximiser le résultat d'une fonction d'évaluation notée Z. Cette fonction regroupe les différents coûts et gains engendrés par le choix des variables décidables pour générer un score. Dans notre approche, en prenant en compte les coûts évoqués dans la section 5.1.3, on déduit une première fonction de coûts à minimiser qui regroupe et pondère l'ensemble de ceux-ci en fonction de leur importance (voir Équation 5.13).

La relaxation d'une contrainte est fréquemment utilisée pour garantir qu'une solution soit proposée. Par exemple, si lors de ma commande, l'un des produits dont j'ai besoin tombera en rupture de stock demain, mais qu'aucun fournisseur ne peut réapprovisionner ce produit dans un délai aussi court, la contrainte demandant à ce que tous les produits commandés arrivent avant leur rupture estimée est insatisfaisable : le modèle conclura à l'insolvabilité du problème. La relaxation de contrainte de l'équation 5.14 autorise les retards, au détriment d'un coût qui s'ajoutera à la fonction Z.

Valeur heuristique d'un produit

Les fonctions heuristiques présentées ne prennent en compte que les coûts que le solveur doit réduire au minimum en validant les contraintes de délais et de gratuité de transport. En l'état, l'Équation 5.14 ne permet pas de favoriser un produit par rapport à un autre. Deux produits sont donc parfaitement interchangeables tant que la préférence de l'un n'implique pas de grandement dépasser le seuil de gratuité de livraison. Par exemple, s'il est nécessaire de compléter de 100 euros la commande pour atteindre le franco de port, choisir une unité d'un produit à 100 euros ou 20 unités d'un produit à 5 euros mène exactement à la même solution finale.

Comme évoqué section 5.1.6, nous estimons que l'intérêt de compléter la commande avec un produit qui n'est pas en rupture sur la période couverte par l'optimisation se mesure au travers de trois indicateurs : le classement de Pareto du produit, la proximité avec le stock maximal du stock actuel sommé avec les unités déjà dans la commande actuelle, et la date de rupture estimée de ce produit. Cette formalisation de l'intérêt d'un produit de complétion mène à la nouvelle fonction de coût modélisée par l'équation 5.18 : nous ajoutons une estimation de la valeur des produits commandés nommée $E_{i,j}$. Puisque le problème est une minimisation, les coûts sont ajoutés et la valeur estimée soustraite. Enfin, une variation du calcul de $E_{i,j}$ plus proche du contexte terrain est proposée : la

valeur du produit diminue à mesure que la quantité commandée augmente.

5.2.4 Notations

Le tableau 5.1 énumère toutes les constantes nécessaires pour détailler la modélisation, la plupart d'entre elles sont facilement compréhensibles grâce à leur nom, tandis que certaines nécessitent une brève explication. C est une grande constante dont la valeur est supérieure à toute quantité de produit commandée. B_i indique le besoin du produit i dans un horizon de planification donné, M_i représente le seuil de quantité à ne pas dépasser pour le produit i. R_i indique le temps avant la pénurie du produit i. Lorsqu'aucune pénurie n'est prévue dans l'horizon de planification considéré, la constante R_i est fixée à C. Compte tenu d'un fournisseur j, le seuil de gratuité des frais d'expédition W_j quantifie le montant minimal pour bénéficier de la gratuité des frais de port. F_j est une information binaire indiquant si un fournisseur j accepte de fragmenter l'envoi d'une commande ou non. $T_{i,j,k}$ est le seuil quantitatif du produit i à commander au fournisseur j pour bénéficier du k^e niveau de remise sur le prix unitaire. $V_{i,j,k}$ est le prix du produit i commandé au fournisseur j au k^e niveau de remise quantitative.

Le tableau 5.2 énumère les variables utilisées dans le modèle. L'unique ensemble de variables décidables $\{x_{i,j}\}$ correspond à la quantité de produit i commandée au fournisseur j ; il s'agit d'une valeur entière. Pour simplifier, $x_{i,\cdot}$ désigne le regroupement quantitatif des commandes du produit i, la notation sera utilisée à partir de l'Équation 5.15. Dans notre modèle, le délai, $lt_{i,j}$ peut varier en fonction de deux paramètres de la commande en cours : les autres produits commandés au fournisseur j et la capacité de ce dernier à fragmenter l'envoi d'une commande.

Les variables restantes sont toutes des variables binaires. La variable θ_j indique si la solution actuelle implique le fournisseur j ou non. La variable ω_j indique si la commande à un fournisseur j est suffisamment élevée pour atteindre sa condition de livraison gratuite. La variable $\sigma_{i,j}$ indique si au moins une unité du produit i est commandée au fournisseur j dans la solution actuelle. La variable $\rho_{i,j}$ vérifie si la commande passée chez le fournisseur j mène à une livraison de i avant qu'il n'y ait pénurie. La dernière variable $\delta_{i,j,k}$ renseigne sur l'accès à la remises de niveau k, et donc du prix unitaire $V_{i,j,k}$.

5.2.5 Les contraintes en détail

Le système de tarification

Aucune quantité négative d'un produit ne peut être commandée :

$$x_{i,j} \geq 0 \qquad \forall i, j \tag{5.1}$$

La quantité commandée du produit i ne peut dépasser un maximum donné :

$$\sum_{j=0}^{|J|-1} x_{i,j} \leq M_i - So_i \qquad \forall i \tag{5.2}$$

Le produit i ne peut être commandé si le fournisseur j ne le propose pas dans son catalogue :

$$x_{i,j} \leq O_{i,j} * C \qquad \forall i, j \tag{5.3}$$

Le choix du prix unitaire activé pour un produit est défini par les trois contraintes suivantes :

$$\delta_{i,j,k} \leq \frac{x_{i,j}}{T_{i,j,k}} \qquad \forall i, j, k \tag{5.4}$$

$$\sum_{k=0}^{|K|-1} \delta_{i,j,k} = 1 \qquad \forall i, j \tag{5.5}$$

La contrainte 5.4 conditionne l'activation d'une remise à la commande de la quantité suffisante chez un fournisseur j. La contrainte 5.5 garantit qu'un seul niveau de prix de produit peut être activé à la fois.

La réponse aux besoins

Dans une commande spécifique, le délai de livraison d'un produit dépend de la capacité d'un fournisseur à dissocier la commande :

$$\sigma_{i,j} \geq \frac{x_{i,j}}{C} \qquad \forall i, j \tag{5.6}$$

$$lt_{i,j} = max(\{F_j * D_{i,j} * \sigma_{i,j}\}, \{D_{i,j} * \sigma_{i,j}\}) \qquad \forall i, j \qquad (5.7)$$

Avec la contrainte 5.6, $\sigma_{i,j}$ indique si dans la solution courante, le produit i est commandé au fournisseur j. Lorsqu'un fournisseur accepte de dissocier une commande, chaque produit est envoyé seul, le délai de livraison correspondant est donc $D_{i,j}$. Dans le cas contraire, le produit étant envoyé avec d'autres produits, on considère que le délai de livraison du produit i devient le délai de livraison le plus élevé d'un produit commandé au même fournisseur : $max(\{F_j * D_{i,j} * \sigma_{i,j}\})$.

Directement à partir du délai effectif, on peut maintenant calculer si le fournisseur j est en mesure de fournir le produit i à temps, avant pénurie :

$$\rho_{i,j} \leq \frac{R_i}{lt_{i,j}} \qquad \forall i, j \qquad (5.8)$$

Pour être valable, une solution doit fournir une quantité de produit i suffisante pour répondre au besoin :

$$\sum_{j=0}^{|J|-1} x_{i,j} * \rho_{i,j} \geq B_i \qquad \forall i \qquad (5.9)$$

La livraison gratuite

Dès que la solution commande un produit au fournisseur j, ce dernier est impliqué :

$$\theta_j \geq \frac{\sum_{i=0}^{|I|-1} x_{i,j}}{C} \qquad \forall j \qquad (5.10)$$

La livraison gratuite chez le fournisseur j est possible dès que le montant de la commande est supérieur à un seuil donné W_j :

$$\frac{\sum_{i=0,k=0}^{|I|-1,|K|-1} V_{i,j,k} * \delta_{i,j,k} * x_{i,j}}{W_j} \geq \omega_j \qquad \forall j \qquad (5.11)$$

Lorsque deux solutions ne diffèrent que par la gratuité des frais de port, la solution avec gratuité des frais de port sera toujours plus compétitive. Ainsi, l'optimisation favorise toujours $\omega_j = 1$ par rapport à $\omega_j = 0$.

Dans notre modèle, si le fournisseur j est impliqué, son seuil de livraison gratuite doit

être dépassé :

$$\theta_j = \omega_j \qquad \forall j \tag{5.12}$$

Avec la contrainte 5.12, une solution valide n'implique aucun coût de transport, même si cela implique de commander un peu plus que les besoins effectifs.

Fonction de score

La fonction de score associée correspond au montant d'une commande :

$$Z = \sum_{j=0}^{|J|-1} \sum_{i=0}^{|I|-1} \left(\sum_{k=0}^{|K|-1} V_{i,j,k} * \delta_{i,j,k} \right) * x_{i,j} \tag{5.13}$$

L'objectif du modèle est de trouver les valeurs pour les variables $\{x_{i,j}\}$ qui satisfont toutes les contraintes tout en minimisant Z.

5.2.6 La valeur d'un produit

Rappelons que nous introduisons dans le calcul du coût une constante E_i qui indique l'intérêt d'utiliser le produit i pour compléter la commande. Cette valeur permet à un négociant d'interagir avec le modèle par le biais de la fonction de notation suivante :

$$Z' = \sum_{j=0}^{|J|-1} \sum_{i=0}^{|I|-1} \left(\sum_{k=0}^{|K|-1} V_{i,j,k} * \delta_{i,j,k} - \mathbf{E_i} \right) * x_{i,j} \tag{5.14}$$

Même si la définition de la valeur d'un produit est laissée à l'appréciation de l'utilisateur, nous proposons une version basée sur les deux indicateurs listés sur les deux dernières lignes du tableau 5.1 : le « classement ABC » d'un produit et sa proximité à la quantité maximale commandée (CTM ou *closeness to max* ci-dessous).

Le *ABC ranking* du produit i, Pr_i, est directement déterminé par son taux de rotation selon les principes de l'analyse ABC [86]. Les produits classés A représentent les 20 % des produits les plus lucratifs ; à eux seuls, ils génèrent généralement 80 % des bénéfices de l'entreprise. Comme un produit de la catégorie B ou C est souvent associé à un petit besoin, une solution peut indirectement éviter une commande supplémentaire en utilisant les produits B ou C pour atteindre les seuils de livraison gratuite. Par conséquent, via la constante E_i, le modèle favorisera la commande d'un produit de catégorie B ou C par

rapport à un produit de catégorie A. Dans notre modèle, le classement ABC est traduit en une valeur qui indique l'intérêt de compléter une commande avec des produits des catégories A, B ou C. Nous adoptons une estimation simple de Pr_i : un produit B ou C vaut 1 tandis qu'un produit A vaut $0,5$.

Pour un produit i, la *closeness to max*, CTM_i, correspond au rapport entre la quantité commandée et la quantité maximale qui va l'être :

$$CTM_i(x_{i,.}) = \frac{\sum_{j=0}^{|J|-1} x_{i,j}}{M_i - So_i} \qquad \forall i \tag{5.15}$$

Par définition, $0 \leq CTM_i(x_{i,.}) \leq 1$, la valeur d'un produit E_i est alors définie telle que (R_i est le temps disponible avant pénurie) :

$$E_i(x_{i,.}) = \frac{Pr_i}{R_i} * (2 - CTMi(x_{i,.})) \qquad \forall i \tag{5.16}$$

5.2.7 Relaxation du temps avant pénurie (Contrainte 5.8)

Une relaxation lagrangienne est autorisée sur le nombre de jours avant la pénurie, noté R_i. Cette relaxation est introduite pour s'assurer que la contrainte 5.9 autorise un certain degré de retard dans la satisfaction de la demande, évitant ainsi que le problème ne devienne insatisfaisable. L'objectif est de minimiser cette relaxation tout en satisfaisant les autres contraintes.

Une famille de variables quantitatives r_i représentant le nombre de jours de pénurie pour le produit i est introduite en remplaçant l'équation 5.8 par :

$$\rho_{i,j} \leq \frac{R_i + r_i}{lt_{i,j}} \qquad \forall i, j \tag{5.17}$$

Afin de limiter les pénuries dans le système, celles-ci sont pénalisées par une constante C' dans la fonction de score :

$$Z'' = Z' + C' * \sum_{i=0}^{|I|-1} r_i \tag{5.18}$$

Par la suite, C' est fixé comme étant $1/10^e$ de C.

Pour cette relaxation, aucune hypothèse n'est formulée quant à la gravité d'un retard de trois jours sur un seul produit par rapport à un retard d'un jour sur trois produits

différents. Par conséquent, chaque jour de retard pour n'importe quel produit est considéré comme ayant le même poids dans la fonction de coût. Cette approche traite tous les retards de la même manière, en se concentrant sur la minimisation globale des ruptures de stock.

5.3 Matériel et méthodes sur l'optimisation

Cette section commence par expliquer comment nous générons les instances synthétiques sur lesquelles les méthodes de résolution seront comparées. Nous terminons par la présentation des indicateurs de qualité utilisés pour évaluer les performances des méthodes de résolution. Le générateur d'instances synthétiques peut être mis à disposition, mais pour des raisons de confidentialité, l'implémentation des algorithmes de résolution ne peut être fournie.

5.3.1 Génération d'instances synthétiques

Bien que les problèmes générés ne soient pas directement issus du milieu du négoce de matériaux, les instances générées doivent être de difficultés variables et réalistes. Cette difficulté dépend de l'ampleur des catalogues fournisseurs, du nombre de fournisseurs et de produits.

La génération d'une instance implique elle-même la génération d'un ensemble de valeurs, une pour chaque constante répertoriée dans le tableau 5.1. Des nombres aléatoires sont utilisés sur la base des données observées sur le terrain. Deux fonctions, $R_{\text{int}}(x, y)$ et $R_{\text{float}}(x, y)$, sont utilisées pour générer des nombres aléatoires dans l'intervalle spécifié. La fonction $R_{\text{int}}(x, y)$ renvoie un nombre aléatoire uniformément échantillonné dans l'ensemble des entiers $\{x, \ldots, y\}$, tandis que $R_{\text{float}}(x, y)$ renvoie un nombre aléatoire uniformément échantillonné dans l'intervalle des nombres réels $[x, y]$.

Cette génération est découpée en quatre sous-parties. La première détaille les trois paramètres qui régissent la difficulté de l'instance générée. Ensuite, les variables aléatoires intermédiaires nécessaires à la génération des constantes sont présentées. La génération des variables intermédiaires est présentée à la suite, et pour conclure, les contraintes qui encadrent la génération pour proposer une modélisation plausible sont détaillées.

Les choix dans la synthèse d'une instance

La synthèse d'une instance commence avec le choix de trois paramètres qui gouverneront fortement la difficulté du problème d'approvisionnement associé. Le nombre de produits, le nombre de fournisseurs ainsi que la possibilité de fragmenter une commande sont des paramètres de l'instance, décidés par l'expérimentateur et non générés aléatoirement.

Le tableau 5.3 présente les différentes valeurs que nous avons utilisées pour générer les instances. La troisième ligne du tableau 5.3 énumère trois valeurs. *False* correspond à l'option où aucun fournisseur ne propose de dissocier une livraison. À l'inverse, *True* signifie que tous les fournisseurs proposent de dissocier une commande. *Alternate* correspond à un compromis où seuls les fournisseurs de rang pair acceptent de fragmenter la commande. Ces différentes options conduisent à 54 combinaisons. Cependant, l'option *Alternate* n'a pas de sens lorsqu'un seul fournisseur est impliqué, le rapport expérimental de la section 5.4 considère donc seulement 48 combinaisons différentes pour générer les instances.

TABLE 5.3 – Valeurs possibles pour une instance.

notations	*facteurs*	*valeurs possibles*
$\|I\|$	nombre de produits	{10, 30, 50, 100, 500, 1000}
$\|J\|$	nombre de fournisseurs	{1, 2, 3}
F_j	possibilité d'envoyer la commande partiellement	{False, True, Alternate}

Constantes aléatoires intermédiaires

Afin de simuler l'ensemble des constantes présentes dans le tableau 5.1, le tableau 5.4 introduit quatre constantes aléatoires intermédiaires.

BP_i représente le prix de référence pour le produit i, il est utilisé comme base pour générer des écarts de prix de l'ordre de 5 % entre les fournisseurs pour un même produit. PB_i correspond à la quantité du produit i qu'il faudra acheter, si celui-ci s'avère être nécessaire. N_i est une constante binaire, qui vaut 1 avec une probabilité de 20 %, utilisée pour caractériser la rupture ou non d'un produit, et donc la nécessité d'en recommander. Enfin, $N'_{i,j}$ est également une constante binaire, passant à 1 avec une probabilité de 50 %, régissant la disponibilité d'un produit i chez un fournisseur j.

TABLE 5.4 – Constantes intermédiaires utilisées pour la génération d'instances.

notation	*description*	*génération*
BP_i	produit i prix de référence	$R_{\text{float}}(0.1, 1000)$
PB_i	besoins hypothétiques du Produit i	$R_{\text{int}}(1, 1000)$
N_i	le produit i est-il nécessaire (prob. 20 %)	$R_{\text{int}}(0, 1)$
$N'_{i,j}$	Le produit i est-il disponible chez le fournisseur j (prob. 50 %)	$R_{\text{int}}(0, 1)$

La génération des constantes d'une instance

La synthèse d'une instance se fait via ses constantes et les formules indiquées dans le tableau 5.5. Cette génération s'appuie uniquement sur les quatre constantes aléatoires du paragraphe précédent.

Contraintes qui encadrent la génération

Comme $O_{i,j}$ représente la disponibilité d'un produit i chez un fournisseur j, il faut s'assurer qu'au moins un fournisseur peut fournir i. Formellement, cela implique :

$$\sum_{j=0}^{|J|-1} N'_{i,j} \geq 1 \qquad \forall i, j \tag{5.19}$$

Chaque produit est associé à trois niveaux de quantité de remise. $T_{i,j,k}$ représente la quantité minimale du produit i qui doit être commandée au fournisseur j pour bénéficier du niveau de remise k. La génération de cette constante est effectuée de telle sorte que chaque $T_{i,j,k}$ soit supérieur à $T_{i,j,k-1}$. Bien sûr, $T_{i,j,0}$ doit être initialisé, sans condition puisqu'il s'agit du niveau de départ :

$$T_{i,j,0} = 0 \qquad \forall i, j \tag{5.20}$$

$V_{i,j,k}$ représente le prix du produit i chez le fournisseur j lorsque le niveau k de remise est actif. Ainsi, $V_{i,j,k}$ est généré comme une séquence où une remise aléatoire est appliquée, c'est pourquoi seul $V_{i,j,0}$ doit être spécifié, en tant que variation du prix de référence BP_i :

$$V_{i,j,0} = BP_{i,j} * R_{\text{float}}(0.95, 1.05) \qquad \forall i, j \tag{5.21}$$

Alors que M_i représente la quantité maximale qui peut être détenue par le détaillant, il

TABLE 5.5 – Génération des constantes.

notation	*génération*
constantes associées au produit i	
B_i	$PB_i * N_i$
M_i	$PB_i * (1 + R_{\text{float}}(0.1, 1))$
R_i	$R_{\text{int}}(5, 20)$
So_i	$PB_i * R_{\text{float}}(0, 0.5)$
constantes associées au fournisseur j	
W_j	$\left(\sum_{i=0}^{I-1}\sum_{k=0}^{K-1} V_{i,j,k} * B_i * O_{i,j} * \delta_{i,j,k}\right) * 1.2$
constantes associées au produit i et au fournisseur j	
$D_{i,j}$	$R_{\text{int}}(1, 14)$
$O_{i,j}$	$N'_{i,j}$
constantes associées au produit i, au fournisseur j et au niveau de remise k	
$V_{i,j,k}$	$V_{i,j,k-1} * R_{\text{float}}(0.95, 0.99)$
$T_{i,j,k}$	$R_{\text{int}}(10^k, 10^{k+1})\ k$
constantes associées à la valeur du produit i	
Pr_i	$R_{\text{float}}(0, 1)$

faut éviter le cas où il s'avère impossible de commander la quantité nécessaire d'un produit. Dans le cas d'une telle solution irréalisable, M_i est augmenté de manière aléatoire jusqu'à ce que la solution soit réalisable :

$$\text{tant que } M_i < PB_i + So_i \text{ faire } M_i = M_i + R_{\text{int}}(1, 1000) \qquad \forall i \tag{5.22}$$

La classification ABC du produit i (Pr_i) est tirée aléatoirement avec une probabilité d'un tiers pour chaque rang.

Une solution triviale à notre problème consisterait à commander tous les produits nécessaires chez le même fournisseur, dépassant directement le seuil de livraison gratuite s'il est trop bas. Pour éviter que l'algorithme focalise sur cette solution triviale, la contrainte de livraison gratuite est rendue difficile à atteindre, en requérant que le montant total des achats dépasse de 20 % celui de la solution triviale.

Classification des instances selon leur difficulté

Trois facteurs influencent la difficulté de résolution d'une instance : le nombre de produits, le nombre de fournisseurs et leur capacité à fragmenter l'envoi d'une commande. Selon nous, une instance devrait être classée comme facile lorsque toutes les méthodes de résolution conduisent au même délai de livraison, tandis qu'une instance devrait être classée comme difficile lorsque des délais de livraison différents sont proposés. Pour les problèmes faciles, la valeur des produits commandés et le prix d'achat sont primordiaux, tandis que pour les problèmes difficiles, c'est le délai de livraison qui compte le plus.

Notre objectif est ici de classer *a priori* une instance comme facile ou difficile, sans connaître les délais de livraison qui sont un résultat *a posteriori* des méthodes de résolution.

D'après nos observations des temps de résolution du problème, une instance est considérée comme facile si elle répond à l'une des règles suivantes :

— moins de 100 produits sont impliqués ;
— un seul fournisseur est pris en considération ;
— 100 produits sont commandés auprès de 2 ou 3 fournisseurs qui ne permettent pas de dissocier l'envoi des produits ;
— 500 produits ou moins sont commandés auprès de 2 fournisseurs qui ne permettent pas de dissocier l'envoi des produits.

Dans le cas contraire, l'instance est classée comme difficile.

Après avoir exécuté les méthodes de résolution, nous avons évalué la pertinence de cette classification *a priori* et les résultats sont indiqués par la matrice de confusion du tableau 5.6. Ce tableau valide la pertinence de nos critères de difficulté d'une instance.

TABLE 5.6 – Matrice de confusion de la classification des instances.

	prédit difficile	*prédit facile*
délais de livraison différents	11	5
délais de livraison similaires	2	28

5.3.2 Algorithmes de résolution

Cette étude est basée sur différents algorithmes d'optimisation choisis pour leurs différentes manières d'explorer l'espace des solutions possibles. Il ne s'agit pas d'une liste

exhaustive des solveurs qui pourraient être utilisés.

Solveur Quadratique (*QUAD*)

Nous dénommons *QUAD* le solveur quadratique qui résout le DJRP de la section 5.2.1, avec le score Z défini par l'Équation 5.13. Pour la suite de ce chapitre, le terme *solveur* fait spécifiquement référence à QUAD. L'implémentation a été effectuée à l'aide de Gurobi 9.0 en Python [40].

L'utilisation d'un solveur est une méthode classique pour résoudre un DJRP. Toutefois, la limitation du temps de calcul risque de restreindre le passage à l'échelle lors de l'ajout de nouvelles contraintes et variables, ces dernières étant liées au nombre de produits et de fournisseurs.

Algorithme génétique (*GA*)

Si les principes des algorithmes génétiques détaillés section 3.2.4 sont généraux, ils impliquent également des réflexions spécifiques au problème. La fonction d'aptitude dépend du problème, de même que la plage de valeur à utiliser pour la construction aléatoire de la première génération de spécimens. Le processus de mutation dépend également du problème : lors du croisement des parents, le spécimen résultant peut dépasser le stock maximal autorisé ou ne pas commander une quantité suffisante d'un produit obligatoire. Ces caractéristiques spécifiques sont traitées comme suit : lors du croisement de deux spécimens, l'enfant bénéficiera de la stratégie de commande complète pour un produit. Si le parent commandait un produit à deux fournisseurs, les enfants feront de même avec les mêmes quantités : le processus de croisement ne porte pas sur les quantités commandées à un fournisseur de chaque produit, mais plutôt sur la stratégie de commande de chaque produit.

La mutation en revanche, peut se produire sur n'importe quelle quantité commandée de n'importe quel produit. La probabilité qu'une caractéristique subisse une mutation après un croisement est faible. Lorsque la mutation se produit, l'une des trois actions suivantes est aléatoirement sélectionnée :

— *commander plus* : Ajouter à la quantité commandée de ce produit chez ce fournisseur, un nombre aléatoire entre 1 et la quantité maximale qui peut encore être commandée sans dépasser la limite de sur-stock ;

— *commander moins* : Soustraire à la quantité commandée de ce produit chez ce fournisseur un nombre aléatoire compris entre 1 et la quantité maximale qui peut

être retirée sans enfreindre l'exigence de besoins ;

— *transférer une partie de la commande* : Choisir un autre fournisseur chez qui ce produit peut être commandé et lui transférer une fraction aléatoire de la commande.

Des itérations sur trente générations sont effectuées, les spécimens enfants deviennent de nouveaux spécimens parents à croiser ; finalement, le meilleur spécimen de la dernière génération est conservé comme étant la meilleure solution. Après fine-tuning, la population initiale a été fixée à 6 000 specimens, la probabilité de mutation fixée 1.5 % par produit et par fournisseur, l'élitisme pratiqué fait passer 20 % des meilleurs spécimens directement à la génération suivante. Dans la mesure où il a été nécessaire d'implémenter un croisement personnalisé, cet algorithme génétique a été spécifiquement réalisé en Python.

Recuit simulé (*SA*)

Pour faire suite à la présentation générale du recuit simulé proposée section 3.2.4, des adaptations sont nécessaires pour le négoce. À partir d'une solution existante, une solution voisine doit être générée. Pour ce faire, un couple (produit, fournisseur) est ciblé et cinq actions peuvent être activées, à condition qu'elles ne créent pas une solution impossible au regard du besoin et du sur-stockage :

— ajouter une quantité aléatoire du produit ciblé au fournisseur ;
— retirer une quantité aléatoire du produit ciblé au fournisseur ;
— transférer une quantité aléatoire de ce produit à un autre fournisseur ;
— ajouter une unité du produit ciblé au fournisseur ;
— retirer une unité du produit ciblé au fournisseur ;
— fixer à 0 la quantité du produit ciblé commandée au fournisseur.

Après fine-tuning, la chaleur initiale sélectionnée est de 300 000. À chaque étape, on lui retire huit millièmes. Dans la mesure où l'optimal n'est pas connu, cette chaleur initiale mène à un nombre d'itérations maximal d'environ 160 000.

L'ensemble du processus fait l'objet d'une implémentation spécifique en Python.

Algorithme glouton (*Greed*)

En partant de l'intuition développée section 3.2.4, et en se basant sur l'algorithme de Dantzig [22] pour le sac à dos à une seule dimension, un algorithme glouton reposant sur un ratio valeur/prix est proposé. Le problème étant un problème de sac à dos multidimensionnel [62], quelques adaptations sont nécessaires pour que cet algorithme fonctionne. Tout d'abord, le catalogue des fournisseurs donne la possibilité ou non de commander un

produit chez un fournisseur. Si l'on démarre avec une solution antérieure, le catalogue sera limité, rendant impossible la commande de produits auprès d'un fournisseur si aucune quantité n'a été commandée à ce fournisseur dans la solution précédente.

L'algorithme se déroule en trois étapes :

1. commander d'un bloc chaque produit en rupture chez le fournisseur le moins cher tout en réduisant la durée de rupture au minimum en fonction des délais de livraison ;
2. transférer les quantités des produits depuis les fournisseurs qui dépassent la condition de transport gratuit, vers ceux qui ne l'ont pas encore atteinte tout en limitant l'augmentation de la durée de rupture produit ;
3. compléter les commandes pour lesquelles le transport est payant avec une unité d'un produit non nécessaire, qui propose le meilleur ratio valeur/prix dans la solution actuelle ; le tout en limitant la durée de rupture produit, car si la livraison d'un fournisseur n'est pas fragmentable, c'est le produit commandé avec la livraison la plus lente qui devient le délai global de la commande. Répéter cette opération jusqu'à atteindre la gratuité de port.

5.3.3 Chaînes de résolution

Objectif général

Cette étude compare non seulement les algorithmes de résolution présentés, mais aussi avec les processus d'enchaînement (ou *pipeline*) de ces algorithmes, qui les exécutent les uns à la suite des autres pour améliorer les résultats obtenus en amont. Un algorithme de résolution seul sera appelé *algorithme de résolution*, tandis qu'un pipeline de ces algorithmes basé sur un warm-start sera appelé *chaîne de résolution*.

Le warm-start [55], ou « démarrage à chaud » consiste ici à initialiser un algorithme avec un résultat issu de l'exécution d'un autre algorithme, plutôt que de partir d'une commande vide ou aléatoire. Ainsi, seul le premier algorithme d'une chaîne de résolution démarre « à froid ». À titre d'exemple, il est possible d'initialiser un algorithme génétique en ajoutant dans sa première population une bonne solution obtenue par un autre algorithme de résolution tel qu'un recuit simulé. De même, une solution obtenue par un premier algorithme peut être utilisée pour limiter le catalogue considéré par l'algorithme glouton qui le suit.

Méthodes comparées

Dans notre étude expérimentale, les 48 instances générées ont été soumises aux 25 méthodes de résolution énumérées dans la première colonne du tableau **??** de l'annexe C. Les tableaux ci-dessous se concentrant sur les meilleurs d'entre eux. La notation « + » indique la succession de différents algorithmes dans une chaîne de résolution.

5.3.4 Indicateurs de qualité

L'objectif de cette étude est de comparer l'efficacité de plusieurs méthodes de résolution sur plusieurs instances de problèmes. Afin de fournir des informations intelligibles, nous proposons une évaluation absolue de la qualité. Puisque l'efficacité d'une méthode de résolution est liée au temps, une méthode d'early-stopping est également présentée.

Qualité d'une méthode de résolution

Bien que chaque algorithme utilise sa propre fonction d'aptitude pour guider sa recherche, nous avons également besoin d'une mesure pour évaluer la qualité d'une solution, qui soit indépendante de la méthode de résolution utilisée pour pouvoir les comparer entre elles. Les trois algorithmes heuristiques (*GA*, *SA* ou *Greed*) peuvent produire des solutions qui ne respectent pas la contrainte de livraison gratuite (Contrainte 5.12). Dans ce cas, la solution est pénalisée pour chaque fournisseur dont le seuil de livraison gratuite n'est pas atteint. En utilisant les notations de la section 5.2.1, le nombre de fois qu'une solution viole la contrainte 5.12 de livraison gratuite correspond à :

$$\overline{FS} = \sum_{j=0}^{|J|-1} \theta_j * (1 - \omega_j) \tag{5.23}$$

À la fin, la mesure de la qualité d'une solution provient directement de la fonction de coût définie par l'équation 5.18 :

$$Z^{(3)} = Z'' + \Omega * \overline{FS} \tag{5.24}$$

Ω est une constante suffisamment grande pour défavoriser sans équivoque le nombre de violations de la contrainte de livraison gratuite. Pour cela, Ω est fixée à 10^7 car Z'' ne dépasse jamais 10^6.

TABLE 5.7 – Pondération des facteurs dans la fonction de score.

facteur	poids avant pondération	pondération	poids après pondération
prix total	10^7	10^0	10^7
valeur totale	10^2	10^4	10^6
nombre de transport payés	$10^0 * \Omega$	10^7	$10^7 * \Omega$
Délai cumulé	$10^0 * \sum_{i=0}^{\|I\|-1} r_i$	10^6	$10^6 * \sum_{i=0}^{\|I\|-1} r_i$

La mesure $Z^{(3)}$ qualifie de façon absolue la qualité d'une solution. Cependant, sa valeur est difficile à interpréter et nous préférons, dans les résultats de la section suivante, indiquer la qualité d'une méthode par rapport à celle d'un algorithme de référence. $QUAD$ fournissant une solution exacte du problème, nous choisissons sa qualité comme référence. Nous appelons ce rapport la *détérioration de la qualité* fournie par une méthode de résolution.

Par exemple, sur la deuxième colonne du tableau 5.8, la deuxième ligne indique que SA produit une solution de qualité moyenne égale à 1.39 fois celle de $QUAD$: SA détériore le résultat de $QUAD$ de 39 %.

Délai cumulé d'une solution

La relaxation du délai avant pénurie introduit dans la section 5.2.7 permet à une solution de fournir des produits après rupture de stock. Ces délais sont pris en compte dans la qualité d'une solution sous le terme d'un *retard cumulé*, qui correspond au second terme de l'équation 5.18 : $\sum_{i=0}^{|I|-1} r_i$ où r_i correspond au nombre de jours de pénurie du produit i.

Ce retard cumulé est spécifiquement utilisé dans la discussion sur les instances classées difficiles, car c'est là que le retard est le plus discriminant pour le choix de la méthode de résolution.

Intuition sur la qualité d'une solution

Chaque facteur de qualité (prix, valeur, coût de transport, délai cumulé) est pondéré en fonction de son importance estimée. Parmi ces facteurs, la valeur est le moins important. L'objectif principal est en effet de limiter le coût total, d'éviter les retards et de

ne pas payer de frais de port. La magnitude utilisée pour ces facteurs est détaillée dans le tableau 5.7. Le but est de hiérarchiser l'importance des différents facteurs dans l'optimisation. Par exemple, la valeur ne doit jamais dépasser le prix pour un produit, sinon commander le maximum de chaque produit deviendrait la meilleure solution.

Pour les comparaisons à venir, effectuées en pourcentage, on peut raisonnablement considérer que les dégradations de qualité de plus de 5% impliquent une extension des délais de rupture produit. Tandis que les variations inférieures seront plutôt liées à une optimisation du prix et de la valeur obtenue. Le transport gratuit est lui toujours atteint. Cette constatation relève de la généralité. Une optimisation du prix peut très bien mener à une amélioration de plus de 5%.

Mesure du temps d'exécution

Le *temps de calcul* correspond au temps alloué à une méthode de résolution pour résoudre une instance donnée. Ce temps d'exécution étant limité à 20 minutes par algorithme, le temps total d'exécution n'excède jamais 20 minutes multipliées par le nombre d'algorithmes impliqués dans la méthode de résolution. Par exemple, le temps de calcul pour une chaîne de résolution *QUAD+Greed+GA* ne peut pas dépasser $3 * 20$min soit une heure.

Mesure du temps d'exécution lié à l'early-stopping

Pour tirer le meilleur parti du temps disponible, il faut dans les chaînes de résolution veiller à ce que qu'un algorithme soit arrêté rapidement dès que la qualité de sa solution actuelle ne s'améliore plus ou pas suffisamment. Ceci ne concerne que les algorithmes SA et GA, qui améliorent leur solution itérativement. Dans le tableau 5.8, nous reportons le *temps d'early-stopping*, défini comme la durée à partir de laquelle la résolution de 80 % des instances cesse de s'améliorer (26 instances sur 33 pour les instances faciles et 10 instances sur 13 pour les instances difficiles).

5.4 Résultats et discussion

Cette section résume et compare les résultats obtenus avec chaque méthode de résolution. Les principaux résultats sont présentés dans le tableau 5.8 pour 33 instances faciles, dans le tableau 5.10 pour 13 instances difficiles. L'intégralité des résultats est disponible

à l'annexe C.

Ces expérimentations montrent que certaines chaînes de traitement produisent une solution de qualité comparable à celle d'un algorithme seul, tout en réduisant considérablement le temps de calcul.

5.4.1 Résultats pour les 33 instances faciles

Comparaisons des algorithmes de résolution

La première ligne tableau 5.8 indique le temps d'exécution de *QUAD*, la référence de qualité. Parmi les trois algorithmes restants, *GA* fournit les meilleures solutions et ne détériore la qualité que de 9 %, cependant c'est le plus lent. *Greed* est la méthode la plus rapide, mais produit des solutions d'une qualité nettement inférieure (23 % de dégradation). Commencer par *Greed* pourrait donc être un moyen efficace de fournir une solution initiale qui servirait d'entrée à un second algorithme plus complexe.

TABLE 5.8 – Résultats moyens sur les 33 instances faciles.

méthode de résolution	*détérioration (%)*	*temps d'exécution*	*temps early stopping*
QUAD	0 (référence)	13.58 min	N.C. [a]
SA	39.26	58.98 s	51.58 s
Greed	22.98	8.74 s	N.C.
GA	9.02	16.34 min	16.34 min
QUAD+SA	-0.08	14.97 min	13.98 min
QUAD+SA+GA	-0.08	31.32 min	13.98 min
QUAD+Greed+SA	0.56	15.26 min	14.95 min
QUAD+Greed+SA+GA	0.32	31.56 min	26.68 min
Greed+SA	6.25	67.28 s	56.27 s
Greed+SA+GA	5.98	17.43 min	15.18 min
GA+Greed+SA	2.06	17.41 min	17.16 min

a. Non Concerné

Apport de *Greed* en warm-start

Le tableau 5.9 se concentre sur les chaînes de résolution impliquant *Greed* en début ou milieu d'exécution. En effet, lorsqu'il est utilisé en dernier, il ne fournit jamais de résultats compétitif. La dernière colonne indique l'amélioration de qualité relative induit par *Greed*.

Par exemple, la première ligne indique que *Greed* exécuté avant *SA* améliore la solution de 31 %.

L'intégration de *Greed* est d'autant plus pertinente qu'il est économe en temps de calcul (+5.62 *s* au maximum).

TABLE 5.9 – Apport de *Greed* en warm-start sur les instances faciles.

méthode de résolution	*amélioration*	*variation du temps d'exécution*
Greed+SA	31.0 %	+4.69 s
Greed+GA	1.7 %	+5.62 s
Greed+GA+SA	2.1 %	+0.22 s
Greed+SA+GA	3.6 %	-90 s
GA+Greed+SA	6.0 %	-1.85 s
SA+Greed+GA	4.36 %	+3.77 s

Rappelons que *QUAD* fait référence à l'utilisation d'un solveur, il ne s'agit pas d'un algorithme heuristique. Ainsi, comme la recherche de *QUAD* peut ne pas être terminée dans le temps imparti, son utilisation en temps restreint n'offre aucune garantie quant à la qualité de la solution qu'il fournit. Dans la pratique, il s'agit là d'un argument de poids pour exclure *QUAD*. Lorsque l'on exclut *QUAD* de la chaîne de traitement, trois méthodes de résolution se distinguent. *Greed+SA* est de loin la méthode de résolution la plus rapide à exécuter tout en conduisant à une dégradation acceptable de 6.25 % par rapport à QUAD. Pour trouver un traitement améliorant ces résultats, le temps de calcul augmente considérablement : *Greed+SA+GA* est meilleur de 0.27 % mais son exécution est en moyenne 16 fois plus longue. *GA+Greed+SA* ne dégrade la solution de *QUAD* que de 2.06 %, en moyenne, mais son temps de calcul est 18 fois plus long que *Greed+SA* pour une amélioration de 4.18 %.

Meilleures méthodes avec QUAD

Les résultats des chaînes de traitement utilisant *QUAD* produisent des solutions de qualité, au prix d'un temps de calcul considérable. Dans la plupart des cas, la meilleure méthode *QUAD+SA+GA* améliore *QUAD* de 0,08 %.

L'utilisation d'un warm-start à partir d'une solution *QUAD* peut se faire dans deux cas : la solution *QUAD* peut être donnée en entrée à un autre algorithme, ou le résultat de *QUAD* peut élaguer le catalogue ultérieurement utilisé par *Greed*. Dans le cas d'un

élagage pour *Greed*, l'utilisation de *QUAD* est toujours intéressante, c'est pourquoi nous préférons utiliser *QUAD* comme prétraitement de *Greed*.

Lorsque l'on se limite aux méthodes où *QUAD* précède *Greed*, deux méthodes fiables sont trouvées. *QUAD+Greed+SA+GA* conduit au meilleur résultat disponible, et à la durée la plus longue. En revanche, *Greed+SA+GA* permet d'obtenir un résultat proche (+0,32 % par rapport à *QUAD*), tout en divisant presque par deux le temps de calcul. Ces résultats montrent que le chaînage améliore les résultats pour un temps de calcul similaire ou réduit.

5.4.2 Résultats moyens pour les 13 instances difficiles

Les résultats obtenus pour les 13 [1] instances difficiles sont présentés dans le tableau 5.10. *SA* obtient clairement les meilleurs résultats, se situant en moyenne à 5.69 % de *QUAD*. *QUAD* n'est jamais battu par un autre algorithme seul.

TABLE 5.10 – Résultats moyens pour les 13 instances difficiles.

méthode de résolution	*détérioration (%)*	*retard cumulé (jour)*	*temps d'exécution*	*temps early stopping*
QUAD	référence (0)	350.15	24.71 min	N.C.
GA	18.14	376.85	20.72 min	20.72 min
Greed	16.58	383.08	60.08 s	N.C.
SA	5.69	355.31	5.29 min	5.24 min
QUAD+SA	-0.90	344.08	30.68 min	26.44 min
QUAD+GA+SA	-0.90	344.08	51.12 min	26.02 min
QUAD+Greed+SA	-0.59	343.77	31.17 min	31.13 min
QUAD+Greed+GA+SA	-0.86	343.08	52.14 min	47.68 min
Greed+SA	0.25	347.23	6.27 min	6.25 min
Greed+GA+SA	0.16	346.31	27.13 min	25.44 min
GA+Greed+SA	-0.20	345.69	26.91 min	26.91 min

Apport de *Greed* en warm-start

Comme le montre le tableau 5.11, à l'exception de *SA+Greed+GA*, l'utilisation de *Greed* avant le dernier algorithme de la chaîne se traduit toujours par une amélioration de la qualité. Encore une fois, *Greed* est un bon point de départ ou une transition efficace,

1. Ces résultats ne concernent que 13 des 15 instances initialement testées, la méthode *QUAD* n'ayant pas donné de résultats à la fin du temps imparti pour les deux instances exclues.

mais son traitement prend désormais jusqu'à 60 secondes supplémentaires : une plus grande difficulté des instances s'accompagne d'une augmentation du temps de calcul.

TABLE 5.11 – Apport de *Greed* en warm-start sur les instances difficiles.

méthode de résolution	*amélioration*	*variation du temps d'exécution*
Greed+SA	5.42 %	+60.05 s
Greed+GA	5.60 %	-31.47 s
Greed+GA+SA	3.81 %	-40.09 s
Greed+SA+GA	4.78 %	+61.43 s
GA+Greed+SA	4.19 %	+47.94 s
SA+Greed+GA	-1.45 %	+291.65 s

Pour les instances difficiles, la meilleure méthode résolution n'impliquant pas *QUAD* est à nouveau *GA+Greed+SA*, améliorant *QUAD* de 0.2 %. Il s'agit toutefois de l'un des processus les plus longs. Si l'on tient compte de la consommation de temps, les deux autres meilleures méthodes seraient *Greed+GA+SA* et *Greed+SA*, qui nécessitent respectivement 27.13 mn et 6.27 mn pour être exécutées avec un early-stopping, dégradant toutes deux le résultat de 0.16 % et 0.25 %.

Meilleures méthodes avec *QUAD*

La meilleure chaîne de résolution est *QUAD+SA* avec 99.10 % en 30.68 mn, une amélioration mineure mais pour une extension du temps de calcul assez faible. Une fois encore, si l'on considère uniquement l'optimisation du délai à partir de *QUAD+Greed*, les deux meilleurs chaînes sont *QUAD+Greed+SA* et *QUAD+Greed+GA+SA*. Toutes deux impliquent un temps d'exécution très long pour une amélioration faible. Pour les instances difficiles, les résultats montrent que les chaînes de résolution permettent d'améliorer les résultats par rapport à *QUAD* dans le cadre d'une contrainte de temps limitée.

5.4.3 Discussion sur les résultats globaux

Meilleures méthodes sans QUAD

Le tableau 5.12 résume les meilleures méthodes de résolution à sélectionner en fonction de la difficulté des instances et du temps disponible, selon utilisation ou non de *QUAD*. Bien qu'il existe quelques différences mineures, les conclusions sont les mêmes pour le

TABLE 5.12 – Méthodes de résolution les plus intéressantes.

méthode de résolution	*deter. qualité (%)*	*early stopping*	*caractéristiques essentielles*
sur des instances faciles sans QUAD			
Greed+SA	6.25	56 s	bons résultats, très rapide
Greed+SA+GA	5.98	15.18 min	amélioration légère, lent
GA+Greed+SA	2.06	17.17 min	bon résultat, très lent
sur des instances faciles avec QUAD			
QUAD+Greed+SA+GA	0.32	26.68 min	meilleur résultat, lent
Greed+SA+GA	0.56	14.95 min	excellent résultat, rapide
sur les instances difficiles sans QUAD			
Greed+SA	0.25	6.245 min	excellent résultat, rapide
Greed+GA+SA	0.16	25.43 min	amélioration légère, assez lent
GA+Greed+SA	-0.20	26.9 min	meilleur, lent
sur les instances difficiles avec QUAD			
QUAD+Greed+SA	-0.59	31.13 min	excellent resultat, lent
QUAD+Greed+GA+SA	-0.86	47.66 min	meilleur résultat, coût maximal

traitement des instances faciles ou difficiles : *Greed* s'avère pertinent pour améliorer les résultats et accélère la convergence de *SA* et *GA*. Et les chaînes de traitement peuvent trouver de meilleures solutions que *QUAD* seul.

Comme le temps nécessaire à *QUAD* est difficile à prévoir, son utilisation en temps restreint pourrait conduire à une solution de mauvaise qualité, voire aucune solution. En cas d'échec de *QUAD*, la recherche devrait se poursuivre sur *GA+Greed+SA*.

5.5 Conclusion

Cette étude a montré qu'il est difficile d'atteindre l'optimalité pour les instances difficiles. Or, ces instances difficiles sont monnaie courante, avec par exemple seulement trois fournisseurs et 1 000 produits. Même avec une modélisation plus simple, il demeure de nombreux cas où le solveur n'atteindra pas l'optimalité dans un temps acceptable pour une utilisation en temps réel. Bien que le solveur donne de bons résultats pour la plupart des instances, il conduit à des comportements instables en fonction de la difficulté rencontrée, menant à des difficultés pour arrêter le processus et conserver le meilleur nœud trouvé. Le solveur implique une phase de pré-résolution du problème, celle-ci peut durer plusieurs minutes et un arrêt de l'exécution dans cette phase ne proposera aucune solution

(car aucun noeud n'aura été exploré).

Nous avons montré que l'enchaînement des méthodes d'optimisation est efficace, en particulier, la solution de l'algorithme *Greed* permet de réaliser un warm-start menant à de meilleures solution pour quelques secondes de traitement supplémentaires.

Cette étude a également montré l'efficacité de l'enchaînement d'algorithmes de résolution lorsqu'un temps limite d'exécution est défini, tirant parti d'algorithmes tels que *GA* et *SA*, dont le passage à l'échelle est plus facile à anticiper. En outre, en fonction des contraintes futures qui pourraient devoir être ajoutées à la modélisation, telles que les coûts liés au stockage des produits, *QUAD* pourrait devenir impossible à utiliser ou difficile à maintenir, tandis que *GA* et *SA* ne nécessiteraient qu'un ajustement de la fonction d'aptitude. Même si le problème est ici modélisé de manière générale, l'ajout de spécificités argumentera en faveur du chaînage, qui facilite la maintenance et l'utilisation dans un contexte d'entreprise.

Alors que nous pensions que la séparation entre les cas difficiles et les cas faciles permettrait de prendre des décisions plus éclairées, il s'avère que les préconisations convergent.

Notre contribution consiste donc à avoir proposé une bonne solution pour un problème spécifique basé sur le DJRP avec un grand espace de recherche et un contexte d'exécution en temps réel. Les résultats de cette étude sont l'objet d'une soumission à l'*European Journal of Operational Research*.

Perspectives

L'utilisation d'une modélisation plus simple de *QUAD* pourrait permettre d'apporter une première solution à l'optimisation des retards. On pourrait alors utiliser cette optimisation des délais comme solution initiale pour restreindre les autres algorithmes de résolution. À l'extrême, on pourrait modifier le catalogue des fournisseurs pour que seuls les couples (produit, fournisseur) recommandés dans la solution de *QUAD* soient commandables par les algorithmes suivants.

Les algorithmes considérés par cette étude ne constituent pas une liste exhaustive, et d'autres algorithmes de résolution pourraient être envisagés. Pour garantir un meilleur compromis entre l'efficacité et la consommation de temps, des études supplémentaires sur les conditions d'arrêt prématuré et la détérioration des résultats sont nécessaires.

Pour finir, notons que la modélisation requiert que l'intégralité des produits doivent être livrée avant la date de pénurie, sous peine de générer une pénalité de retard. Il pourrait

cependant s'avérer nécessaire d'introduire un pas de temps supplémentaire, autorisant un réapprovisionnement *partiel* afin de retarder la pénurie de produits. Bien entendu, la difficulté augmentera considérablement avec un tel ajout. Dans cette optique, une réflexion est menée en annexe D pour rendre périodique une partie des réapprovisionnements à l'aide d'un système d'automatisation, réduisant la difficulté liée au nombre de produits. Cette approche permettrait notamment de déterminer un cycle optimal de réapprovisionnement et de coupler le réapprovisionnement des produits de cycle similaires, allégeant le processus d'optimisation des produits à faible valeur ajoutée.

Chapitre 6

Conclusion et Perspectives

Comme nous l'avons montré dans le chapitre 4, bien que les modèles prédictifs récents comme les réseaux de neurones fonctionnent, ils sont coûteux à entraîner sur autant de références et nécessitent un volume de donnée dont peu d'entreprises disposent. *A contrario*, les méthodes plus anciennes demandent avant tout une compréhension de leur fonctionnement (saisonnalité, tendance), mais offrent des résultats raisonnables à moindre coûts (de donnée, et de calcul).

Nous avons également proposé un modèle de scénario choisissant automatiquement entre les différents modèles YM, DHR et LSTM. Dans le cas de nombreux produits, ce système améliore la qualité de la prédiction et réduit le temps d'apprentissage.

L'expérimentation sur le choix de la meilleure méthode à utiliser en fonction des caractéristiques de la série temporelle s'est avérée plus efficace que l'utilisation d'un modèle unique pour toutes les prédictions.

Concernant les méthodes d'optimisations comparées dans le chapitre 5, des constations similaires sont observées : différents solveurs (mathématiques et heuristiques) mènent à différentes contraintes d'implémentations et résultats, et la réduction de la complexité (et donc du temps d'exécution) ne peut passer que par une limitation du problème permettant de réduire l'espace à explorer. Dans notre cas, nous avons choisi de sacrifier la temporalité du problème pour accélérer la convergence du modèle vers l'optimalité. Mais là encore, l'optimalité n'est atteinte que pour un nombre de produit et de fournisseurs limités. Dans notre contexte temps réel, nous avons privilégié une proposition rapide de réapprovisionnement, reposant sur un enchaînement de méthodes différentes et d'early-stopping plutôt que laisser un algorithme s'exécuter pendant toute la durée de calcul disponible.

Pour ces deux problèmes distincts, la conclusion est similaire : il est critique de garder à l'esprit qu'il n'existe pas de méthode miracle qui transformerait moins de descripteurs en de meilleurs résultats prédictifs ou une méthode de résolution applicable à tous les problèmes d'optimisation. Fort de cette constatation, il convient de déterminer des méthodes

annexes permettant de mettre à profit toutes ces méthodes, en les combinant si le temps le permet, ou au contraire en étant capable de sélectionner la meilleure. La maîtrise de tous les outils disponibles pour adapter un choix technologique en fonction du domaine concerne notamment le choix d'un modèle prédictif en fonction du volume de donnée disponible, mais aussi des caractéristiques de la série temporelle, comme son écart relatif. Pour l'optimisation, un chaînage d'algorithmes de résolution adaptés et un early-stopping bien calibré sont essentiels.

Ce travail de thèse nous a permis de découvrir le domaine industriel spécifique du négoce de matériaux et de l'analyser à la lumière de concepts informatiques récents. Il semble que le contexte des entreprises de négoce, tel qu'exposé dans le chapitre 1, reflète davantage l'ère de l'industrie 3.0 (période allant de 1970 à 2010) que celle de l'industrie 4.0. Dans ce contexte, les ressources disponibles et les méthodes de collecte de données sont plus proches de la 3.0. Les éléments essentiels sont collectés, mais l'idée de contrôler l'intégralité du processus de distribution au moyen de capteurs n'est pas d'actualité. Les expérimentations menées le confirment : dans le secteur du négoce, les techniques de prédiction et d'optimisation héritées de l'industrie 3.0 sont encore tout à fait pertinentes, obtenant des résultats satisfaisants en un temps relativement court.

Perspectives

Meta-learning et prédiction

Une amélioration classique mais chronophage consiste à déterminer des descripteurs supplémentaires pour la prédiction des ventes de produits. On pourrait par exemple considérer des événements locaux, comme des promotions chez les négociants, ou le cours des matières premières qui constituent le produit cible. Dans le même registre, la corrélation des ventes de certains produits pourrait permettre de les prédire plus facilement leurs ventes : si la vente de carrelage s'effondre, la vente de colle à carrelage suivra probablement la même pente, mais des corrélations moins évidentes existent peut-être. Une décomposition plus précise des clients pourrait également être envisagée.

Les méthodes de prédiction mises en œuvre n'adressent pas encore totalement le problème de prédire les ventes de plus de 10 000 références produit. Puisque moins de 5 % d'entre eux sont vendus fréquemment, une solution naïve serait de s'intéresser principalement au top 500 des produits les plus vendus, et se contenter d'effectuer des moyennes glissantes ou des lissages exponentiels sur le reste des produits. Pour rendre cette sélec-

tion moins arbitraire, il semble que le meta-learning permette de déterminer le meilleur modèle prédictif à utiliser en fonction des caractéristiques de la série temporelle. L'efficacité du scénario prédictif présenté chapitre 4 est encourageante, surtout si l'on considère qu'il n'est basé que sur des descripteurs très basiques. Une étude supplémentaire est en cours, basée sur des descripteurs avancés : *silhouette*, *kurtosis*, *skewness*, ainsi que des variantes exécutées sur des séries temporelles désaisonnalisées ou sans tendance. Dans le même temps, des algorithmes de prédiction supplémentaires comme le lissage exponentiel simple, double ou triple s'ajouteront aux modèles disponibles, le but étant d'attribuer le temps d'exécution disponible en fonction de la criticité des produits.

Il conviendra cependant de déterminer si le modèle de *meta-learning* entraîné est efficace pour d'autres clients, ou si au contraire il est nécessaire d'en entraîner un par négociant, menant à des problèmes de coût d'entraînement dans la mesure où celui-ci requiert une comparaison de tous les modèles sur un ensemble de produit sélectionné. Finalement, concernant le modèle de sélection d'algorithme, un arbre de décision est actuellement utilisé en production car son efficacité est bien connue en classification supervisée. Des expérimentations sur une classification par réseau de neurones ou des règles d'association en discrétisant les descripteurs doivent être menées. Mélanger les produits de plusieurs négociants, pour créer un modèle plus générique, est également une piste intéressante.

Réduire la complexité générale du réapprovisionnement

Si l'on considère que la prédiction des volumes de vente de tous les produits est disponible, la complexité impliquée par la résolution d'un problème de réapprovisionnement joint avec 10 000 produits et 20 fournisseurs est extrêmement élevée. Pourtant, comme pour les prédictions, seule une petite partie de ces produits nécessite une optimisation avancée. Pour autant, il ne faut pas négliger les produits peu vendus, ils peuvent s'avérer utiles pour compléter la commande et atteindre le transport gratuit, et doivent eux aussi être réapprovisionnés occasionnellement. Il convient donc d'adopter une approche permettant de réduire la complexité générale du problème, en prenant en compte les besoins en réapprovisionnement de ces produits subsidiaires.

Les différents types de produit

On distingue trois niveaux d'importance pour le réapprovisionnement : *nécessaire*, impliquant un besoin à court-moyen terme dans la période que l'optimisation cherche à

couvrir, directement concerné par l'optimisation puisque celui-ci tombe en rupture ; un *produit intéressant*, résultant d'un besoin moyen-long terme, clé dans le cadre de la complétion d'une commande, car ces produits seront en rupture dans la prédiction étendue ; finalement, les produits qui ne sont ni nécessaires ni intéressants : les produits dont aucune rupture n'est prévue, même en effectuant une prédiction qui dépasse la durée que l'on cherche à couvrir dans l'optimisation.

De nombreux produits à faible renouvellement peuvent être retirés de l'optimisation. Effectivement, un produit qui ne tombe pas en rupture de stock à long terme n'a aucune raison d'être réapprovisionné. On peut alors exclure facilement tout un ensemble de produits. En réalité, ces produits qui ne tombent pas en rupture, ou représentent un coût et un bénéfice très faible pour l'entreprise, sont la majorité des références. Sur l'ensemble de 9938 produits d'un négociant particulier, si l'on s'intéresse à filtrer les produits qui sont vendus au minimum au moins une fois par semaine en moyenne, c'est-à-dire qui représentent 150 occurrences de vente sur les 3 dernières années, seuls 422 produits demeurent. Cela signifie que 9 516 produits sont faiblement vendus, leur impact n'est globalement que peu significatif sur les revenus du négociant. Pour ces produits dont le volume est faible et les réapprovisionnements peu fréquents voire inexistants (produit en stock mais aucune vente), une autre approche doit être envisagée.

Réapprovisionner les produits intéressants

L'approche historique de résolution du JRP semble intéressante : elle permet de définir des cycles de commande optimaux «automatiques» et cycliques.

Un début d'expérimentation pour déterminer ces cycles de réapprovisionnement à l'aide de Simulink est proposé Annexe B.1. On cherchera à commander ensemble les produits dont les cycles coïncident, afin d'obtenir des habitudes de commandes régulières. On pourrait alors imaginer que, tous les 6 mois (26 semaines), une commande de tous les produits de cycle 26 semaines soit constituée. Les produits concernés pourraient alors être retirés de l'optimisation des produits clés. Cette optimisation des cycles, bien que conséquente, ne serait effectuée que rarement, car elle est basée sur des ventes annuelles moyennes qui varient peu.

Annexe A

Stationnarité et régression fallacieuse

On appelle *fallacieuse* une régression menant à la détermination de fausses corrélations entre les variables explicatives et celle expliquée. En économétrie, la régression fallacieuse est le plus souvent liée aux régressions effectuées sur des variables non-stationnaires ou des variables *confondantes*. Cependant, en dehors du contexte spécifique des séries temporelles et de l'économétrie, le terme « régression fallacieuse » est souvent utilisé de manière plus générale pour décrire toute situation où une corrélation entre deux variables est mal interprétée comme une relation de cause à effet.

Afin d'illustrer concrètement ce qu'est une régression fallacieuse, un exemple est proposé. Dans celui-ci, nous allons nous intéresser à deux séries temporelles : la vente de tuile de couleur rouge, et la vente de tuile de couleur ocre. La figure A.1 semble très clairement indiquer une corrélation négative : plus on vend de tuiles rouges, moins on vend de tuiles jaunes. En réalité, la tendance à la hausse des tuiles rouges, et la tendance à la baisse des tuiles jaunes cache la véritable relation entre ces deux modèles. Il est clair que ces deux séries temporelles ne sont pas stationnaires, elles évoluent fortement au cours du temps. Afin de rendre stationnaires nos deux séries temporelles, une *différenciation* d'ordre 1 est appliquée. Cette différenciation est définie avec X_t le volume de vente à un instant donné, tel que :

$$X'_t = X_t - X_{t-1}$$

La figure A.2 donne le résultat des deux séries temporelles après la différenciation. On s'aperçoit alors clairement – sur cet exemple volontairement simpliste – que les deux séries subissent exactement les mêmes variations au même moment. Si la tendance à l'augmentation des tuiles rouges et celle à la baisse des tuiles jaunes est indiscutable, il est faux de déduire que ces deux variables sont négativement corrélées. En réalité, ces deux séries temporelles sont positivement corrélées entre elles.

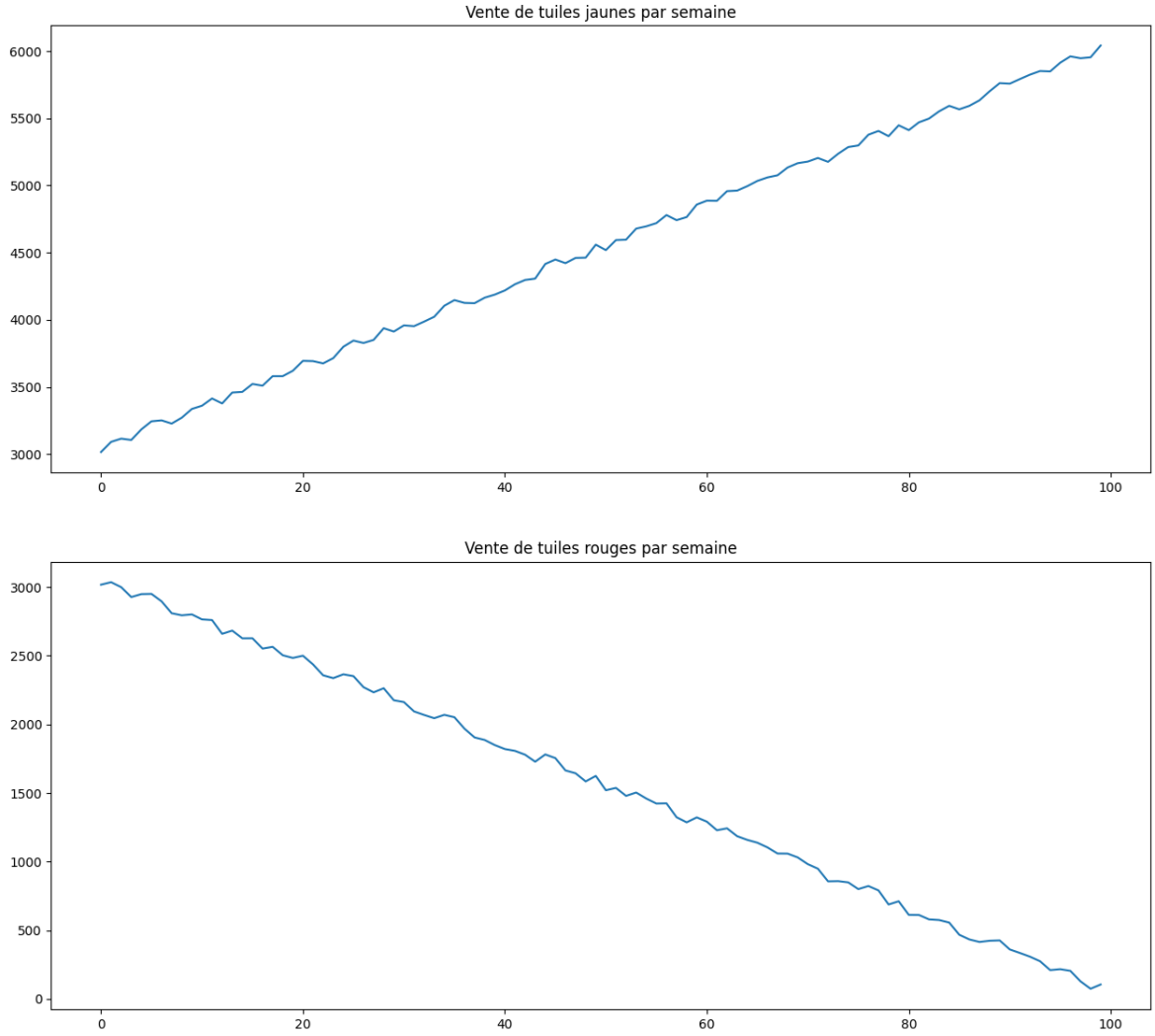

FIGURE A.1 – Vente hebdomadaire de tuiles en fonction de leurs couleurs.

Causalité directe, indirecte et corrélation

Pour prédire la vente de tuile, on peut considérer que la vente de tuile est en relation de *causalité directe* avec les conditions météorologique extrêmes comme la grêle ou les tempêtes qui abîment les toitures et impliquent des réparations rapides. La pluie en revanche peut être considérée comme une cause indirecte d'une baisse des ventes de tuiles, car il est généralement admis que les couvreurs travaillent moins lorsqu'il pleut et que les propriétaires remettent aux beaux jours leurs travaux de toitures. La pluie contribue à

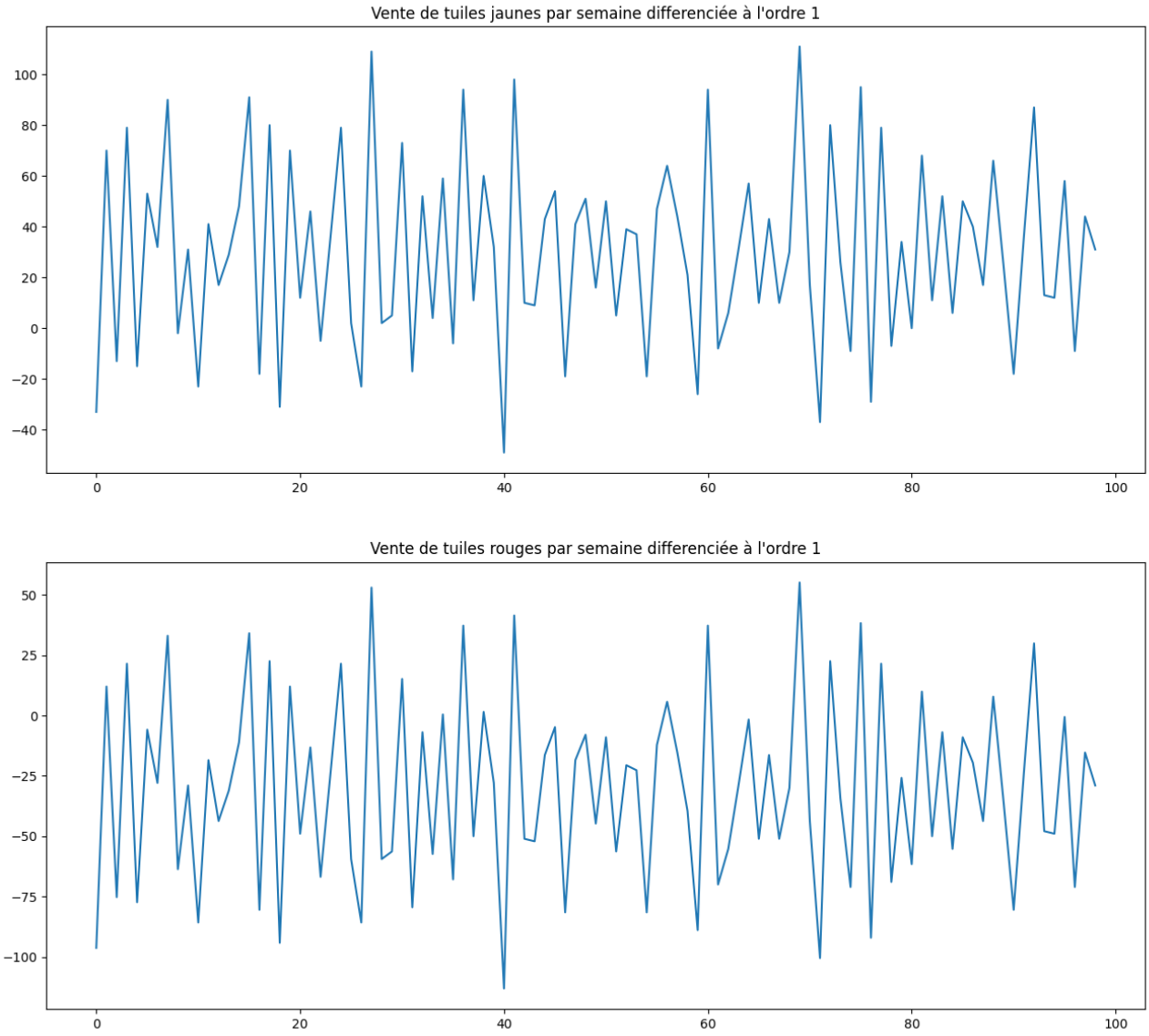

FIGURE A.2 – Vente hebdomadaire de tuiles en fonction de leurs couleurs différenciées à l'ordre 1.

un ensemble de circonstances qui ont un impact sur la demande étudiée, mais elle dépend surtout de l'attitude des couvreurs et des propriétaires.

Dans l'exemple des tuiles, la régression fallacieuse est causée par la non-stationnarité des séries étudiées, parce qu'il existe une évolution temporelle de celles-ci causée par une tendance à long terme. Les mêmes observations existent pour les tendances saisonnières,

et pourraient par exemple mener à considérer que les ventes de glaces et le nombre de noyades sont corrélées, alors que ce sont les températures estivales qui favorisent les deux. S'il est possible de baser une régression sur une telle corrélation, le manque de causalité est dangereux : le nombre de glaces vendues ne baissera pas si des mesures supplémentaires sont prises pour réduire le nombre de noyades, il n'est donc pas avisé de baser une régression qui prédira les ventes de glace sur les statistiques de noyades, il faudra plutôt s'intéresser à la relation de cause à effet avec les températures.

Annexe B

Valeurs considérées pour les paramètres des différentes méthodes, paramétrage utilisé pour la partie expérimentale

---

Le paramétrage de chaque modèle expérimenté au chapitre 4 est détaillé par un tableau : le tableau B.1 est dédié au paramétrage DHR, le tableau B.2 au paramétrage LSTM, le tableau B.3 au paramétrage RF. Pour chaque modèle, la plage des valeurs possibles de chaque paramètre est spécifiée avec les notations suivantes : $[MIN : MAX, STEP]$ signifie que l'on parcourt le domaine en allant de MIN à MAX avec un pas de $STEP$, $[VAL1, VAL2, ...]$ correspond à la trajectoire des éléments mentionnés. Le nombre de combinaisons considérées est indiqué sur la dernière ligne de chaque tableau.

Table B.1 – détail de la paramétrisation d'une DHR.

paramètre	plage possible
ordre d'auto-regression	[0 : 5 , 1]
ordre de moyenne glissante de l'erreur	[0 : 5 , 1]
nombre de combinaisons	36

TABLE B.2 – détail de la paramétrisation d'un LSTM.

paramètre	plage possible
epoch	100
batch-size	[5, 25, 90]
n-layers	[3]
n-steps	[5, 10, 25]
units	[16, 32, 64]
dropout	[0.4]
learning-rate	[0.001, 0.01, 0.02]
earling-stopping patience	[3]
shuffle batch	[True, False]
nombre de combinaison	162

TABLE B.3 – détail de la paramétrisation d'une RF.

paramètre	plage possible
profondeur maximale	[6 : 10 , 1]
nombre d'arbres	[800 : 1400 , 200]
nombre de combinaison	20

B.1 Efficacité des filtres pour chaque modèle

Une partie des descripteurs des modèles est retirée en se basant sur l'importance de ces descripteurs dans le processus de génération de la forêt aléatoire. Comme le montre le tableau B.4, seule la DHR est améliorée par ce processus de sélection.

TABLE B.4 – Efficacité du filtrage par importance de la forêt aléatoire.

modèle	erreur moyenne sans filtre	erreur moyenne avec filtre
DHR	30.9 %	**29.4 %**
LSTM	**34.0 %**	38.5 %
RF	**34.5** %	35.7 %

Annexe C

Résultats complets des méthodes de résolution pour les instances faciles et difficiles

C.1 Résultats des instances faciles et difficiles

méthode de résolution	*détérioration (%)*	*temps d'exécution*	*temps early stopping*
QUAD	référence (0)	13.58 min	N.C.
SA	39.26	58.98 s	51.58 s
Greed	22.98	8.74 s	N.C.
GA	**9.02**	16.34 min	**16.34 min**
QUAD+GA	-0.04	29.82 min	13.56 min
QUAD+GA+SA	-0.07	30.80 min	13.56 min
QUAD+SA	**-0.08**	14.97 min	**13.98 min**
QUAD+SA+GA	-0.08	31.33 min	13.98 min
QUAD+Greed	14.12	13.80 min	13.80 min
QUAD+Greed+GA	2.52	30.15 min	28.08 min
QUAD+Greed+GA+SA	2.04	31.12 min	28.80 min
QUAD+Greed+SA	**0.56**	15.27 min	**14.95 min**
QUAD+Greed+SA+GA	**0.32**	31.56 min	**26.68 min**
GA+Greed	14.66	16.44 min	16.44 min
GA+SA+Greed	14.43	17.42 min	17.32 min
SA+GA+Greed	14.25	17.42 min	16.87 min
SA+Greed	12.68	67.85 s	60.45 s
SA+GA	9.78	17.27 min	16.71 min
GA+SA	8.17	17.28 min	17.19 min
Greed+GA	7.18	16.43 min	16.43 min
Greed+SA	**6.25**	67.28 s	**56.27 s**
Greed+GA+SA	5.98	17.40 min	17.20 min
SA+Greed+GA	5.19	17.44 min	16.78 min
Greed+SA+GA	**5.98**	17.43 min	**15.18 min**
GA+Greed+SA	**2.06**	17.41 min	**17.16 min**

C.2 Résultats moyens pour les 13 instances difficiles

méthode de résolution	*détérioration (%)*	délai cumulé	*temps d'exécution (s)*	*early stopping (s)*
QUAD	référence (0)	350.15	24.71 min	N.C.
GA	18.14	376.85	20.72 min	20.72 min
Greed	16.58	383.08	60.08 s	N.C.
SA	**5.69**	355.31	5.27 min	**5.24 min**
QUAD+GA	référénce (0)	350.15	45.89 min	25.04 min
QUAD+SA	-0.90	344.08	30.68 min	26.44 min
QUAD+SA+GA	-0.90	344.08	51.42 min	1586.42 min
QUAD+GA+SA	**-0.90**	344.08	51.12 min	**26.02 min**
QUAD+Greed	7.27	365.08	26.11 min	26.11 min
QUAD+Greed+GA	6.14	362.92	46.89 min	42.46 min
QUAD+Greed+SA	**-0.59**	343.77	31.17 min	**31.13 min**
QUAD+Greed+SA+GA	-0.59	343.77	51.93 min	31.13 min
QUAD+Greed+GA+SA	**-0.86**	343.08	52.15 min	**47.68 min**
GA+Greed	14.19	379.62	21.78 min	21.78 min
Greed+GA	11.87	368.31	21.87 min	20.20 min
GA+SA+Greed	9.15	365.08	27.13 min	27.13 min
SA+Greed	8.04	365.31	6.27 min	6.25 min
SA+GA+Greed	7.32	364.62	26.96 min	6.24 min
SA+Greed+GA	6.59	364.23	26.99 min	10.08 min
SA+GA	5.04	354.62	25.98 min	5.22 min
GA+SA	3.98	352.62	26.10 min	26.10 min
Greed+SA+GA	0.25	347.23	27.11 min	6.25 min
Greed+SA	**0.25**	347.23	6.27 min	**6.24 min**
Greed+GA+SA	**0.16**	346.31	27.13 min	**25.44 min**
GA+Greed+SA	**-0.20**	345.69	26.91 min	**26.91 min**

Annexe D

Optimisation de la distribution : recherche du cycle optimal d'un produit

Notre volonté est d'automatiser la détermination du cycle optimal des produits avec une approche basée sur les concepts de traitement du signal. En effet, nous sommes convaincu qu'une optimisation rapide mais pertinente pour tous les produits d'une entreprise requiert la concentration des ressources de calcul sur les produits clé. Pour ceux-ci, il faudra appliquer les méthodes de résolution décrites dans le chapitre 5. Pour les produits moins importants, il convient d'attribuer une petite partie de la ressource, pour s'assurer une maîtrise du contexte général. C'est l'objet de l'étude présentée dans ce chapitre : déterminer des cycles de réapprovisionnement automatiques.

Principe de l'automatisation de la recherche de cycle

Ce principe, c'est celui utilisé au début des années 60 et décrit section 3.2.2 de l'état de l'art. Il convient de déterminer un cycle optimal de réapprovisionnement pour chaque produit. Puis de regrouper les produits de cycles multiples pour réduire des coûts majeurs, comme par exemple réduire le coût de transport à 0 en atteignant les frais de port gratuits.

Dans la pratique, ce cycle dépend de plusieurs paramètres que l'on considère connus : la prédiction des ventes du produit, le délai de réception du produit, les réceptions déjà prévues de ce produit et le stock initial disponible. On effectue ensuite une déduction de cycle optimal sur la période temporelle considérée. Le processus étant une somme de signaux de consommation, de réception et de compensation d'erreur qui évoluent dans le temps, il peut être représenté à l'aide d'outils de traitement du signal, comme par exemple Simulink et Matlab.

Le cycle de commande d'un produit

Si l'on considère que la granularité temporelle est la semaine, et qu'il est possible de commander chez un fournisseur toutes les x semaines, on déduit que la borne minimale du cycle commande est x. La borne maximale du cycle elle, correspond au stock maximal du produit divisé par sa consommation moyenne sur la période à couvrir.

Déterminer le cycle optimal d'un produit

Pour les négoces de matériaux, le cycle optimal est facile à déterminer dans la mesure où les coûts de stocks ne sont pas considérés. Dans ce cas, le cycle optimal est le cycle maximal puisque le coût de commande crée un déséquilibre avec le coût de stockage nul et les produits non périssables. Pour d'autres métiers en revanche, un équilibre entre coût de stockage et coût de commande doit être trouvé ; c'est la motivation principale du travail amorcé dans cette annexe.

Mise en place d'une simulation : principe

La figure D.1 illustre le fonctionnement théorique que nous allons chercher à modéliser pour simuler une consommation variable, mais connue à l'avance à un certain horizon noté h (bloc violet), et une correction sur les prédictions constatées (bloc marron). Enfin, en se basant sur ces deux informations, la meilleure façon d'influer sur le futur à l'instant t doit être déterminée (bloc blanc). C'est ce bloc blanc qui cristallise l'essence du problème : un réapprovisionnement doit être effectué en tenant compte du stock maximal, des prédictions de réception et de consommation. Dans l'idéal, il faut à partir de cette représentation déterminer un cycle de réapprovisionnement fixe qui limite le stock maximal, et évite les ruptures de stock.

Modélisation de la simulation dans Simulink

Simulink est une plate-forme de simulation de systèmes dynamiques, à l'aide d'un environnement graphique intégré à Matlab.

Nous détaillons ici comment l'outil Simulink permet d'implémenter le schéma de la figure D.1 : la prédiction des ventes, le stock ou les réceptions de commande sont utilisés comme des signaux.

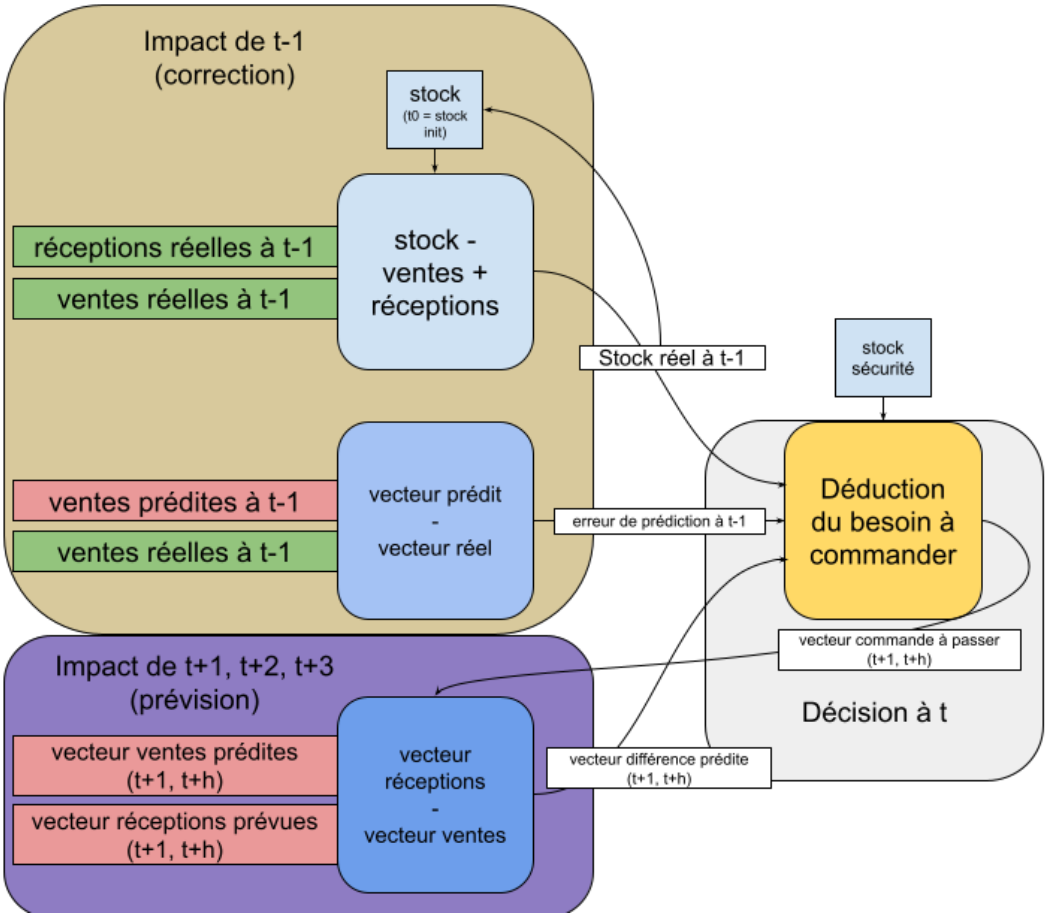

FIGURE D.1 – Principe de la gestion de flux de vente appuyée par prédiction, avec correction.

Traitement général La figure D.2 reprend les quatre blocs de traitement présentés par la figure D.1. On y retrouve les entrées et sorties principales de la modélisation et l'on extrait à la fin de ce traitement la première valeur du bloc 4, qui vient enrichir l'historique du stock. La figure D.7 retrace cet historique au fil du temps.

Bloc 1 : variation de stock par vente / réception Le bloc 1, détaillé par la figure D.3, prend en entrée deux signaux : la vente et la réception de produits pour la journée t. Ce bloc permet de calculer le stock à t : c'est le stock à $t-1$ augmenté de la différence entre la réception et la consommation. En sortie, on retourne le stock, ultérieurement considéré comme le stock de la veille (à $t-1$). En effet, ce délai est nécessaire parce qu'il n'est pas possible de savoir à t quelles sont les ventes et les réceptions qui

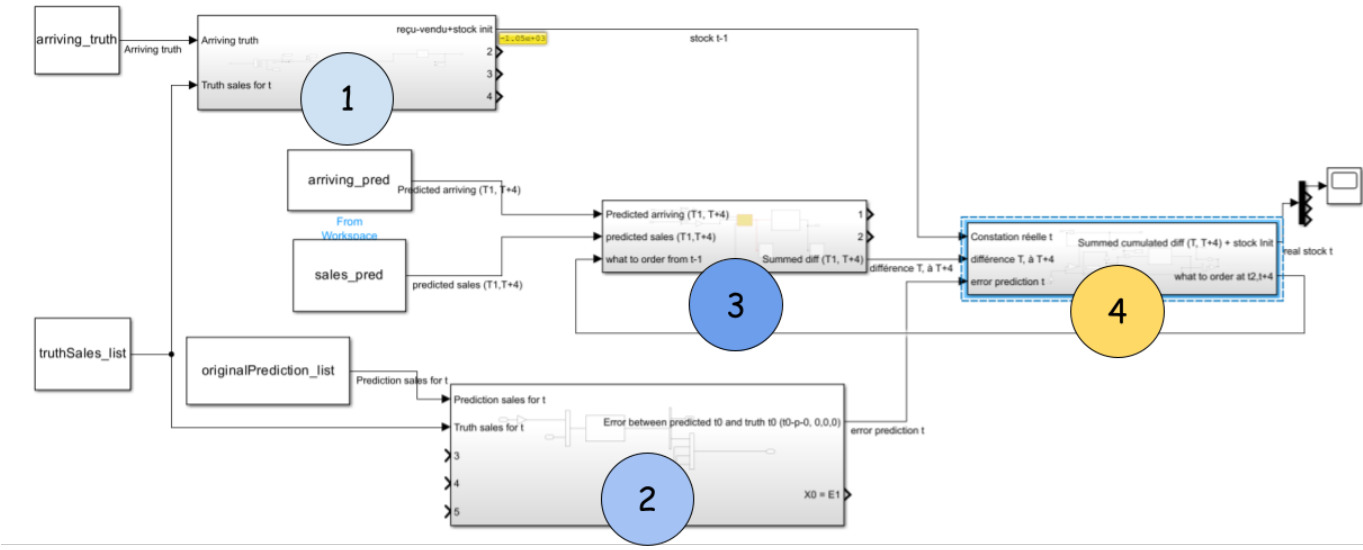

FIGURE D.2 – Enchaînement des quatre blocs décrits précédemment.

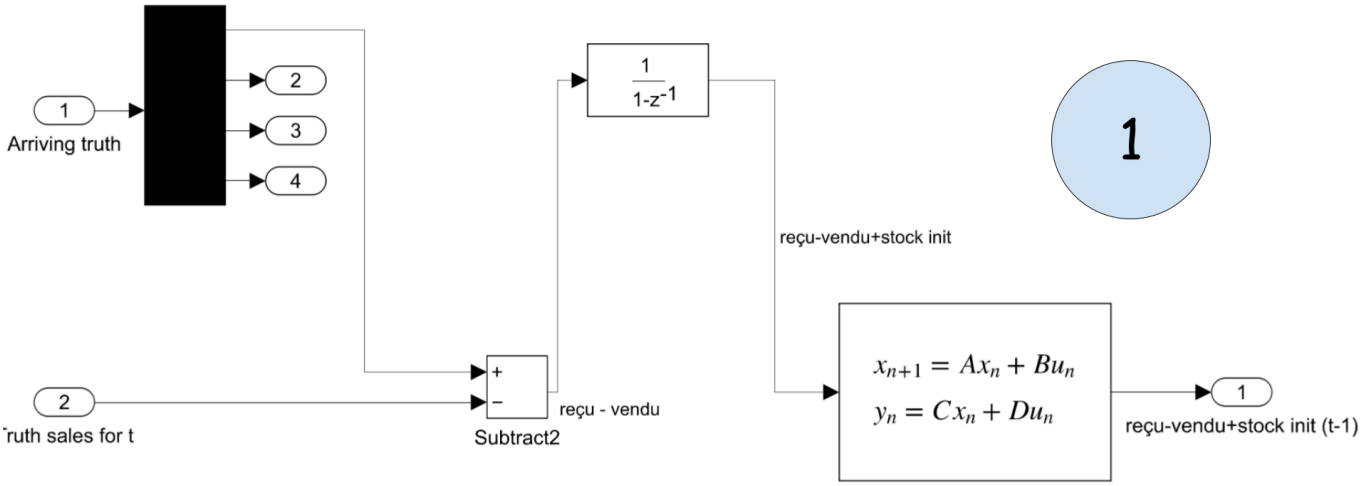

FIGURE D.3 – Bloc 1 : variation de stock.

ont eu lieu. Ainsi, le stock constaté ne peut être mis à jour qu'en fin de journée, et les compensations qu'il implique ne peuvent être prises qu'à $t + 1$.

Bloc 2 : constatation de l'erreur entre la prédiction et la consommation réelle

Le bloc 2, présenté figure D.4, prend en entrée la prédiction des ventes et leur valeur réelle puis transmet en sortie la différence, à $t - 1$ dans l'optique de compenser l'erreur.

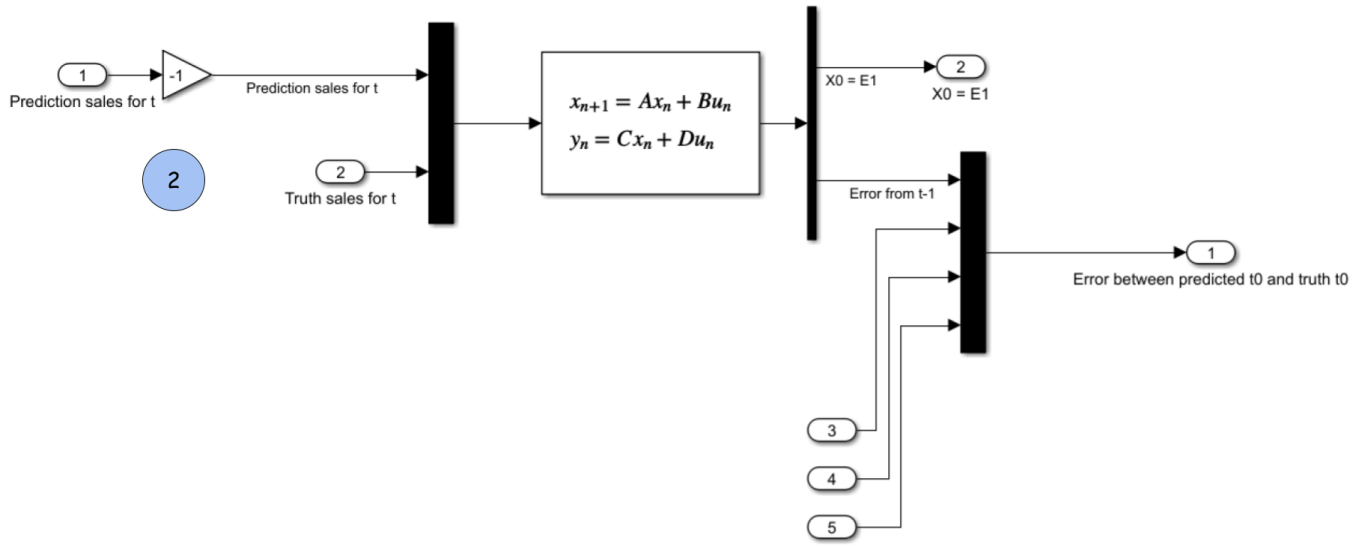

FIGURE D.4 – Bloc 2 : erreur entre la prédiction et la consommation réelle.

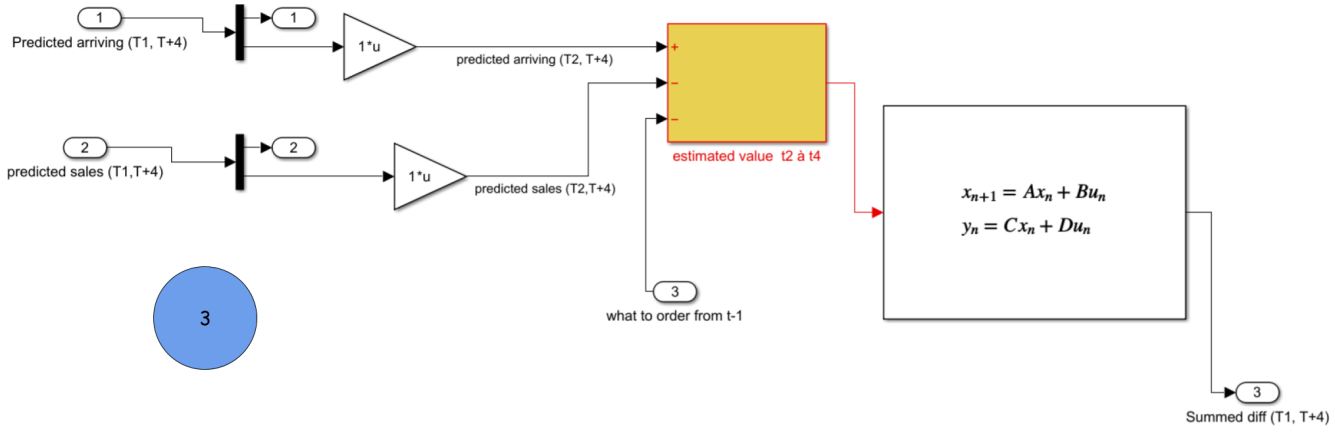

FIGURE D.5 – Bloc 3 : prédiction des variations de stock.

Bloc 3 : prédiction des variations de stock Le bloc 3 (figure D.5) prend en entrée les prédictions de vente et de réception prévues entre t à $t+h$ ainsi que les décisions de commandes prises à $t-1$ pour effectuer une estimation des variations à venir entre t à $t+h$.

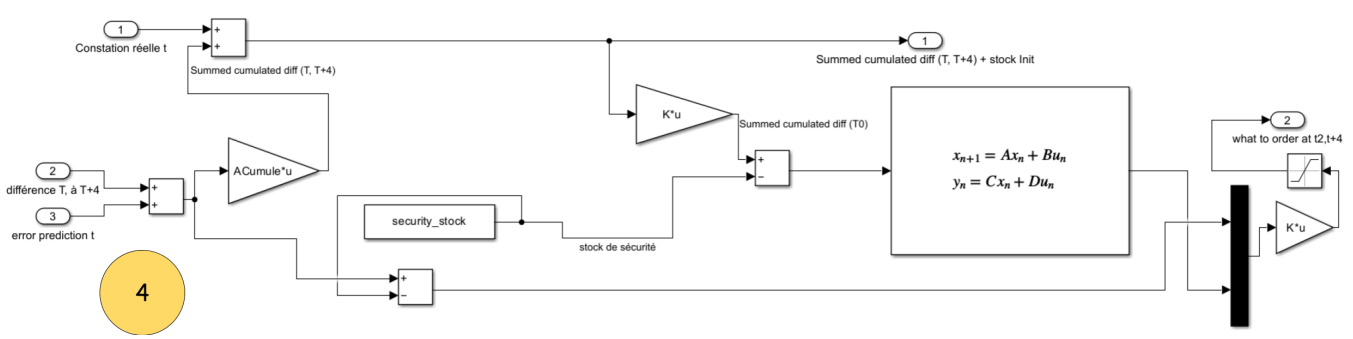

FIGURE D.6 – Bloc 4 : commande à effectuer.

Bloc 4 : prise de décision sur la commande à effectuer Le bloc 4 (figure D.6) est le bloc décisionnel du système. En se basant sur les variations du stock à venir et l'erreur entre la prédiction et les ventes réelles à $t-1$, on déduit la variation qu'il va falloir compenser, et donc commander dans le futur proche. Finalement, on ajoute le stock de sécurité pour s'y maintenir. On retourne alors la commande à passer à t, et on affiche le stock constaté à t actuel dans la figure D.7.

Conclusion

En l'état actuel, la simulation permet un réapprovisionnement sans contrainte et une réception dès le sur-lendemain de la commande. Elle considère un stock initial ainsi qu'un stock de sécurité sur lequel elle essaie de stabiliser son stock.

Cependant, des ajustements sont nécessaires. La prise en compte des coûts de commande, de stockage et de vente ratée est obligatoire pour équilibrer un tel modèle, qui a tendance à commander chaque jour en petite quantité pour palier les besoins à venir. De même, la prise en compte d'un horizon de prédiction plus élevé, du stockage maximum et du délai fournisseur réel aura une importance déterminante pour le modèle.

Pour autant, les résultats sont encourageants : les variations de stock présentées figure D.7 montrent que la simulation se stabilise autour d'un stock minimal (2000 dans notre exemple).

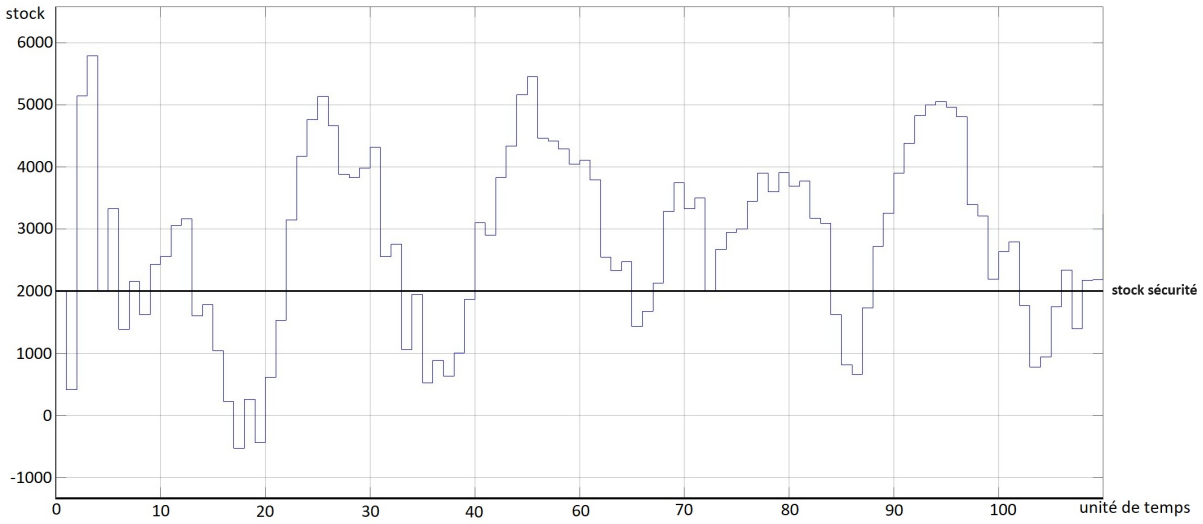

FIGURE D.7 – Régulation du flux de vente appuyée par prédiction, avec correction.

Perspectives

Regrouper les cycles de commande

Une fois les cycles de tous les produits déterminés, il est nécessaire de regrouper ces produits afin de limiter le coût majeur de commande. Ce regroupement a toujours le même objectif : atteindre le seuil de transport gratuit pour annuler ce coût majeur. Pour ce faire, on va commencer par regrouper ensemble tous les produits de même cycle : par exemple, on commandera tous les produits de cycle 14 semaines en même temps. Ce regroupement ne suffira cependant pas à garantir que toutes les commandes atteignent le seuil de transport gratuit. On pourra ensuite regrouper les produits en réduisant les cycles de commande déterminés. Par exemple, au lieu de commander la quantité maximale d'un produit toutes les 12 semaines, on pourra en commander deux fois moins toutes les 6 semaines, ce qui permettra d'ajouter ce produit à la commande passée toutes les 6 semaines et rapprocher cette commande cyclique du transport gratuit.

Regrouper les commandes selon les cycles d'approvisionnement produits

L'optimisation de ce regroupement des commandes pose question. Sachant que plusieurs fournisseurs vendent potentiellement le produit, déterminer chez qui le commander

et comment regrouper les commandes cycliques n'a pas de réponse évidente. Cette réflexion doit encore être approfondie mais il semble qu'un solveur linéaire pourrait répondre à cette problématique assez efficacement dans la mesure où la variable à déterminer pour chaque produit (le cycle optimal) a une plage de valeur possible assez faible. De plus, cette optimisation, même si elle s'avérait coûteuse, ne serait à réaliser qu'une fois pour une période longue et pour tous les produits en une seule fois.

La simulation permet de recevoir une commande à n'importe quel moment entre $t+1$ et $t+h$ (h l'horizon) pour éviter la rupture, son but est de limiter ces possibilités de réceptions pour qu'elles correspondent à un cycle de commande. Par exemple, avec un horizon de 12 semaines, et la possibilité de commander uniquement à $t+4$, $t+8$ et $t+12$ (en cycle de 4 semaines donc). Bien que la quantité à commander soit automatisée (elle résulte d'une prédiction), il n'est pas raisonnable de commander une quantité fixe d'un produit sans égard du stock disponible. Il faut donc passer commande toutes les 4 semaines de la partie du stock vendue entre ces 4 semaines. Le but sera alors de faire ces traitement avec une matrice produit pour déterminer un cycle pour chacun d'entre eux, permettant de proposer une *régulation automatique du flux des produits*, en fonction des conditions de transport gratuit à atteindre et d'un coût de stockage s'il est fourni. Cette régulation allège les efforts d'optimisation pour les concentrer sur les produits clés.

BIBLIOGRAPHIE

[1] Steven B. Achelis, *Technical Analysis from A to Z*, Eng, McGraw-Hill Professional, 2014, chap. PART TWO : Reference - CHAIKIN OSCILLATOR.

[2] Esther Arkin, Dev Joneja et Robin Roundy, « Computational complexity of uncapacitated multi-echelon production planning problems », in : *Operations research letters* 8.*2* (1989), p. 61-66.

[3] Jushan Bai et Serena Ng, « Tests for skewness, kurtosis, and normality for time series data », in : *Journal of Business & Economic Statistics* 23.*1* (2005), p. 49-60.

[4] Ronald H. Ballou, « Evaluating inventory management performance using a turnover curve », in : *International Journal of Physical Distribution & Logistics Management* 30.*1* (2000), p. 72-85.

[5] Dimitris Bertsimas et John Tsitsiklis, « Simulated annealing », in : *Statistical science* 8.*1* (1993), p. 10-15.

[6] F.F. Boctor, G. Laporte et J. Renaud, « Models and algorithms for the dynamic-demand joint replenishment problem », in : *International Journal of Production Research* 42.*13* (2004), p. 2667-2678.

[7] Fayez Fouad Boctor, Gilbert Laporte et Jacques Renaud, *The Dynamic-demand Joint Replenishment Problem Revisited : New Solution Procedures and Comparative Evaluation*, Faculté des sciences de l'administration de l'Université Laval, 2003.

[8] Nahla Boutouria, Salah Ben Hamad et Imed Medhioub, « Investor Behaviour Heterogeneity in the Options Market : Chartists vs. Fundamentalists in the French Market », Eng, in : *Journal of Economics and Business* 3 (2020).

[9] George E.P. Box et GWILYM M. Jenkins, *Time Series Analysis : Forecasting and Control*, Eng, Holden-Day, 1976.

[10] Leo Breiman, « Random Forests », Eng, in : *Machine Learning* 45 (2001), p. 5-32.

[11] Leo Breiman, « Stacked regressions », in : *Machine learning* 24.*1* (1996), p. 49-64.

[12] Heleen Buldeo Rai et al., « Logistics outsourcing in omnichannel retail : State of practice and service recommendations », in : *International Journal of Physical Distribution & Logistics Management* 49.*3* (2019), p. 267-286.

[13] Gerard P. Cachon, Santiago Gallino et Joseph Xu, « Free Shipping Is Not Free : A Data-Driven Model to Design Free-Shipping Threshold Policies. », in : *SSRN 3250971* (2018).

[14] Vítor Cerqueira et al., « Arbitrated ensemble for time series forecasting », in : *Joint European conference on machine learning and knowledge discovery in databases*, Springer, 2017, p. 478-494.

[15] Robert P. Cerveny et Lawrence W Scott, « A survey of MRP implementation », in : *Production and Inventory Management Journal* 30.*3* (1989), p. 31.

[16] Che-Jung Chang et al., « A latent information function to extend domain attributes to improve the accuracy of small-data-set forecasting », in : *Neurocomputing* 129 (2014), p. 343-349.

[17] Chris Chatfield et al., « Neural networks : Forecasting breakthrough or passing fad ? », in : *International Journal of Forecasting* 9.*1* (1993), p. 1-3.

[18] Chris Chatfield et al., « A new look at models for exponential smoothing », in : *Journal of the Royal Statistical Society : Series D (The Statistician)* 50.*2* (2001), p. 147-159.

[19] C.F. Cheung et F.L. Li, « A quantitative correlation coefficient mining method for business intelligence in small and medium enterprises of trading business », in : *Expert systems with applications* 39.*7* (2012), p. 6279-6291.

[20] Robert B. Cleveland et al., « STL : A seasonal-trend decomposition procedure based on loess », Eng, in : *Journal of Official Statistics* 6 (1990), p. 3-73.

[21] Ligang Cui et al., « A novel locust swarm algorithm for the joint replenishment problem considering multiple discounts simultaneously », in : *Knowledge-Based Systems* 111 (2016), p. 51-62.

[22] George B. Dantzig, « Discrete-variable extremum problems », in : *Operations research* 5.*2* (1957), p. 266-288.

[23] Jan G. De Gooijer et Rob J. Hyndman, « 25 years of time series forecasting », in : *International journal of forecasting* 22.*3* (2006), p. 443-473.

[24] Jan G. De Gooijer et Kuldeep Kumar, « Some recent developments in non-linear time series modelling, testing, and forecasting », in : *International Journal of Forecasting* 8.*2* (1992), p. 135-156.

[25] Elvis Dohmatob et al., « Speeding-up model-selection in GraphNet via early-stopping and univariate feature-screening », in : *2015 International Workshop on Pattern Recognition in NeuroImaging*, IEEE, 2015, p. 17-20.

[26] M.J.G. Van Eijs, « A note on the joint replenishment problem under constant demand », in : *The Journal of the Operational Research Society* 44.*2* (1993), p. 185-191.

[27] M.J.G. Van Eijs, R.M.J. Heuts et J.P.C. Kleijnen, « Analysis and comparison of two strategies for multi-item inventory systems with joint replenishment costs », in : *European Journal of Operational Research* 59.*3* (1992), p. 405-412.

[28] S. Selcuk Erenguc, « Multiproduct dynamic lot-sizing model with coordinated replenishments », in : *Naval Research Logistics (NRL)* 35.*1* (1988), p. 1-22.

[29] Marshall L. Fisher, « The Lagrangian relaxation method for solving integer programming problems », in : *Management science* 27.*1* (1981), p. 1-18.

[30] Donald W. Fogarty et Robert L. Barringer, « Joint order release decisions under dependent demand », in : *PROD. INVENTORY MANAGE.* 28.*1* (1987), p. 55-61.

[31] Dennis Fok, Dick van Dijk et Philip Hans Franses, « Forecasting aggregates using panels of nonlinear time series », in : *International Journal of Forecasting* 21.*4* (2005), p. 785-794.

[32] Jay Wright Forrester, « Industrial dynamics », in : *Journal of the Operational Research Society* 48.*10* (1997), p. 1037-1041.

[33] Simone A French, « Pricing effects on food choices », in : *The Journal of nutrition* 133.*3* (2003), 841S-843S.

[34] Everette S Gardner Jr, « Exponential smoothing : The state of the art », in : *Journal of forecasting* 4.*1* (1985), p. 1-28.

[35] P.C. Gilmore et R.E. Gomory, « The theory and computation of knapsack functions », in : *Operations Research* 14.*6* (1966), p. 1045-1074.

[36] Robert Goodell Brown, « Statistical forecasting for inventory control », in : (1959).

[37] Suresh K Goyal et Ahmet T Satir, « Joint replenishment inventory control : deterministic and stochastic models », in : *European journal of operational research* 38.*1* (1989), p. 2-13.

[38] Suresh K. Goyal, « Determination of optimum packaging frequency of items jointly replenished », in : *Management Science* 21.*4* (1974), p. 436-443.

[39] Varun Gupta, Dmitry Ivanov et Tsan-Ming Choi, « Competitive pricing of substitute products under supply disruption », in : *Omega* 101 (2021), p. 102279.

[40] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, 2023, URL : `https://www.gurobi.com`.

[41] Jose Gutiérrez et al., « Effective replenishment policies for the multi-item dynamic lot-sizing problem with storage capacities », in : *Computers & Operations Research* 40.*12* (2013), p. 2844-2851.

[42] James D Hamilton, « A new approach to the economic analysis of nonstationary time series and the business cycle », in : *Econometrica : Journal of the econometric society* (1989), p. 357-384.

[43] Saeed Heravi, Denise R. Osborn et C.R. Birchenhall, « Linear versus neural network forecasts for European industrial production series », in : *International Journal of forecasting* 20.*3* (2004), p. 435-446.

[44] Mario Hermann, Tobias Pentek et Boris Otto, « Design principles for industrie 4.0 scenarios », in : *2016 49th Hawaii international conference on system sciences (HICSS)*, IEEE, 2016, p. 3928-3937.

[45] Sepp Hochreiter et Jürgen Schmidhuber, « Long Short-Term Memory », Eng, in : *Neural Computation* 9 (1997), p. 1735-80.

[46] Charles C. Holt, « Forecasting seasonals and trends by exponentially weighted moving averages », in : *International journal of forecasting* 20.*1* (2004), p. 5-10.

[47] T. C. Hu, Leo Landa et Man-Tak Shing, « The unbounded knapsack problem », in : *Research Trends in Combinatorial Optimization* (2009), p. 201-217.

[48] Guowei Hua, Shouyang Wang et TC Edwin Cheng, « Price and lead time decisions in dual-channel supply chains », in : *European journal of operational research* 205.*1* (2010), p. 113-126.

[49] Rob Hyndman, « It's time to move from 'what' to 'why' », in : *International Journal of Forecasting* 17 (jan. 2001), p. 567-570.

[50] Rob J. Hyndman et George Athanasopoulos, *Forecasting : Principles and Practice*, Eng, OTexts, 2013.

[51] Rob J. Hyndman et al., *Prediction intervals for exponential smoothing state space models*, rapp. tech., Monash University, Department of Econometrics et Business Statistics, 2001.

[52] Zahir Irani, « Investment justification of information systems : a focus on the evaluation of MRPII », in : (1998).

[53] Jonathan E. Jackson et Charles L. Munson, « Common replenishment cycle order policies for multiple products with capacity expansion opportunities and quantity discounts », in : *International Journal of Production Economics* 218 (2019), p. 170-184.

[54] Jitendra Kumar Jaiswal et Rita Samikannu, « Application of random forest algorithm on feature subset selection and classification and regression », in : *2017 World Congress on Computing and Communication Technologies (WCCCT)*, IEEE, 2017, p. 65-68.

[55] Elizabeth John et E. Alper. Yildirim, « Implementation of warm-start strategies in interior-point methods for linear programming in fixed dimension », in : *Computational Optimization and Applications* 41.*2* (2008), p. 151-183.

[56] Peter Løchte Jørgensen, « An analysis of the Solvency II regulatory framework's Smith-Wilson model for the term structure of risk-free interest rates », Eng, in : *Journal of Banking and Finance* 97 (2018), p. 219-237.

[57] Arthur F. Veinott Jr, « Minimum concave-cost solution of Leontief substitution models of multi-facility inventory systems », in : *Operations Research* 17.*2* (1969), p. 262-291.

[58] Goyal Sureh K. et Deshmukh S.G., « Discussion A note on 'The economic ordering quantity for jointly replenishing items' », in : *The International Journal of Production Research* 31.*12* (1993), p. 2959-2961.

[59] Konstantinos Kalpakis, Dhiral Gada et Vasundhara Puttagunta, « Distance measures for effective clustering of ARIMA time-series », in : *Proceedings 2001 IEEE international conference on data mining*, IEEE, 2001, p. 273-280.

[60] Moshe Kaspi et Meir J. Rosenblatt, « An improvement of Silver's algorithm for the joint replenishment problem », in : *IIE Transactions* 15.*3* (1983), p. 264-267.

[61] Moshe Kaspi et Meir J. Rosenblatt, « On the economic ordering quantity for jointly replenished items », in : *The International Journal of Production Research* 29.*1* (1991), p. 107-114.

[62] Hans Kellerer, Ulrich Pferschy et David Pisinger, « Multidimensional Knapsack Problems », in : Springer, jan. 2004, p. 235-283, isbn : 978-3-642-07311-3.

[63] Moutaz Khouja et Suresh Goyal, « A review of the joint replenishment problem literature : 1989–2005 », in : *European journal of operational Research* 186.*1* (2008), p. 1-16.

[64] Moutaz Khouja, Zbigniew Michalewicz et Sandeep S. Satoskar, « A comparison between genetic algorithms and the RAND method for solving the joint replenishment problem », in : *Production Planning & Control* 11.*6* (2000), p. 556-564.

[65] Handi Koswara et Dharma Lesmono, « Direct and indirect grouping strategies for a multi-item probabilistic inventory model », in : *Journal of Physics : Conference Series* 1317.*1* (2019), p. 12003.

[66] Bjoern Krollner, Bruce J Vanstone, Gavin R Finnie et al., « Financial time series forecasting with machine learning techniques : a survey. », in : *ESANN*, 2010.

[67] Milan Kumar, Preetam Basu et Balram Avittathur, « Pricing and sourcing strategies for competing retailers in supply chains under disruption risk », in : *European Journal of Operational Research* 265.*2* (2018), p. 533-543.

[68] Jianli Li et Liwen Liu, « Supply chain coordination with quantity discount policy », in : *International Journal of Production Economics* 101.*1* (2006), p. 89-98.

[69] R.A. Low et J.F. Waddington, « The determination of the optimum joint replenishment policy for a group of discount-connected stock lines », in : *Journal of the Operational Research Society* 18.*4* (1967), p. 443-462.

[70] Liang Lu et Xiangtong Qi, « Dynamic lot sizing for multiple products with a new joint replenishment model », in : *European Journal of Operational Research* 212.*1* (2011), p. 74-80.

[71] Bacel Maddah et al., « Joint replenishment model for multiple products with substitution », in : *Applied Mathematical Modelling* 40.*17-18* (2016), p. 7678-7688.

[72] Wiboon Masuchun, S Davis* et JW Patterson, « Comparison of push and pull control strategies for supply network management in a make-to-stock environment », in : *International Journal of Production Research* 42.*20* (2004), p. 4401-4419.

[73] Hardik Meisheri et al., « Scalable multi-product inventory control with lead time constraints using reinforcement learning », in : *Neural Computing and Applications* 34.*3* (2022), p. 1735-1757.

[74] Stefan Minner et Edward A. Silver, « Replenishment policies for multiple products with compound-Poisson demand that share a common warehouse », in : *International Journal of Production Economics* 108.*1-2* (2007), p. 388-398.

[75] Melanie Mitchell, « Genetic algorithms : An overview », in : *Complexity* 1.*1* (1995), p. 31-39.

[76] I.K. Moon, S.K. Goyal et B.C. Cha, « The joint replenishment problem involving multiple suppliers offering quantity discounts », in : *International Journal of Systems Science* 39.*6* (2008), p. 629-637.

[77] John F. Muth, « Optimal properties of exponentially weighted forecasts », in : *Journal of the american statistical association* 55.*290* (1960), p. 299-306.

[78] James O Peckham, « The consumer speaks », in : *Journal of Marketing* 27.*4* (1963), p. 21-26.

[79] C Carl Pegels, « Exponential forecasting : Some new variations », in : *Management Science* (1969), p. 311-315.

[80] Lu Peng, Lin Wang et Sirui Wang, « A review of the joint replenishment problem from 2006 to 2022 », in : *Management System Engineering* 1.*1* (2022), p. 9.

[81] Lutz Prechelt, « Early stopping-but when ? », in : *Neural Networks : Tricks of the trade*, Springer, 2002, p. 55-69.

[82] Edward Qian, « Risk parity and diversification », in : *The Journal of Investing* 20.*1* (2011), p. 119-127.

[83] Hui Qu, Lin Wang et Rui Liu, « A contrastive study of the stochastic location-inventory problem with joint replenishment and independent replenishment », in : *Expert Systems with Applications* 42.*4* (2015), p. 2061-2072.

[84] Parthasarathi Raghavan, *The multi-item lot-sizing problem with joint replenishments*, New York University, Graduate School of Business Administration, 1993.

[85] Frank Rosenblatt, « The perceptron : a probabilistic model for information storage and organization in the brain. », in : *Psychological review* 65.*6* (1958), p. 386.

[86] Marin Rusanescu, « ABC analysis, model for classifying inventory », in : *Hidraulica* 2 (2014), p. 17.

[87] Bhaba R. Sarker et Bingqing Wu, « Optimal models for a single-producer multi-buyer integrated system of deteriorating items with raw materials storage costs », in : *The International Journal of Advanced Manufacturing Technology* 82 (2016), p. 49-63.

[88] Steve Satchell et Allan Timmermann, « On the optimality of adaptive expectations : Muth revisited », in : *International Journal of Forecasting* 11.*3* (1995), p. 407-416.

[89] Maher Selim et al., « Reducing error propagation for long term energy forecasting using multivariate prediction. », in : *CATA*, 2020, p. 161-169.

[90] Arun Sharma et Michael Levy, « Categorization of customers by retail salespeople », in : *Journal of Retailing* 71.*1* (1995), p. 71-81.

[91] Alex Sherstinsky, « Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network », in : *Physica D : Nonlinear Phenomena* 404 (2020), p. 132306.

[92] Edward A Silver, « A control system for coordinated inventory replenishment », in : *International Journal of Production Research* 12.*6* (1974), p. 647-671.

[93] Edward A. Silver, « A simple method of determining order quantities in joint replenishments under deterministic demand », in : *Management science* 22.*12* (1976), p. 1351-1361.

[94] Edward A. Silver, « Coordinated replenishments of items under time-varying demand : dynamic programming formulation », in : *Naval Research Logistics Quarterly* 26.*1* (1979), p. 141-151.

[95] Craig Silverstein, Sergey Brin et Rajeev Motwani, « Beyond market baskets : Generalizing association rules to dependence rules », in : *Data mining and knowledge discovery* 2 (1998), p. 39-68.

[96] A. Smith et T. Wilson, *Fitting Yield Curves with Long Term Constraints*, rapp. tech., Bacon et Woodrow, 2000.

[97] Marc Souply et al., « Sales Volume Prediction and Application to Materials Trading », in : *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*, IEEE, 2022, p. 200-205.

[98] Norman R. Swanson et Halbert White, « Forecasting economic time series using flexible versus fixed specification and linear versus nonlinear econometric models », in : *International journal of Forecasting* 13.*4* (1997), p. 439-461.

[99] Edgar Talavera et al., « Data augmentation techniques in time series domain : a survey and taxonomy », in : *arXiv preprint arXiv :2206.13508* (2022).

[100] Lingling Tang, Yulin Yi et Yuexing Peng, « An ensemble deep learning model for short-term load forecasting based on ARIMA and LSTM », in : *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)* (2019), p. 1-6.

[101] Lingling Tang, Yulin Yi et Yuexing Peng, « An ensemble deep learning model for short-term load forecasting based on ARIMA and LSTM », in : *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, IEEE, 2019, p. 1-6.

[102] Zaiyong Tang et Paul A. Fishwick, « Feedforward neural nets as models for time series forecasting », in : *ORSA journal on computing* 5.*4* (1993), p. 374-385.

[103] Christoph Teller et al., « Retail store operations and food waste », in : *Journal of Cleaner Production* 185 (2018), p. 981-997.

[104] Ayse Soy Temur, Melek Akgün et Gunay Temur, « Predicting housing sales in Turkey using ARIMA, LSTM and hybrid models », in : (2019).

[105] Howell Tong, *Threshold Models in Non-linear Time Series Analysis*, Wiley, 1983.

[106] Yu-Chung Tsao et Jye-Chyi Lu, « Trade promotion policies in manufacturer-retailer supply chains », in : *Transportation Research Part E : Logistics and Transportation Review* 96 (2016), p. 20-39.

[107] Arthur F. Veinott Jr, « Optimal policy for a multi-product, dynamic, nonstationary inventory problem », in : *Management science* 12.*3* (1965), p. 206-222.

[108] José A. Ventura et al., « A multi-product dynamic supply chain inventory model with supplier selection, joint replenishment, and transportation cost », in : *Annals of Operations Research* 316.*2* (2022), p. 729-762.

[109] S. Viswanathan, « A new optimal algorithm for the joint replenishment problem », in : *The Journal of the Operational Research Society* 47.*7* (1996), p. 936-944.

[110] S. Viswanathan, « An algorithm for determining the best lower bound for the stochastic joint replenishment problem », in : *Operations research* 55.*5* (2007), p. 992-996.

[111] S. Viswanathan, « On optimal algorithms for the joint replenishment problem », in : *The Journal of the Operational Research Society* 53.*11* (2002), p. 1286-1290.

[112] Vito Volterra, *Theory of Functionals and of Integro-differential Equations...* Blackie, 1931.

[113] Jack G.A.J. van der Vorst, Adrie JM Beulens et Paul van Beek, « Modelling and simulating multi-echelon food systems », in : *European Journal of Operational Research* 122.*2* (2000), p. 354-366.

[114] Xiaozhe Wang, Kate Smith et Rob J. Hyndman, « Characteristic-based clustering for time series data », in : *Data mining and knowledge Discovery* 13.*3* (2006), p. 335-364.

[115] Norbert Wiener, *Non-linear problems in random theory.* Wiley, 1958.

[116] Tiaojun Xiao et Xiangtong Qi, « Price competition, cost and demand disruptions and coordination of a supply chain with one manufacturer and two competing retailers », in : *Omega* 36.*5* (2008), p. 741-753.

[117] Haiyan Xie et Deepa Palani, « Analysis of overstock in construction supply chain and inventory optimization », in : *Construction Research Congress 2018*, 2018, p. 29-39.

[118] Qing-Song Xu et Yi-Zeng Liang, « Monte Carlo cross validation », in : *Chemometrics and Intelligent Laboratory Systems* 56.*1* (2001), p. 1-11.

[119] Shilei Yang et al., « Price competition for retailers with profit and revenue targets », in : *International Journal of Production Economics* 154 (2014), p. 233-242.

[120] G.U. Yule, « On a Method of Investigating Periodicities in Disturbed Series with a Special Reference to Wolfer's Sunspot numbers, Philosphical Transactions, 1927, 226A », in : *Rs. crore IPFS* (1927).

[121] Willard I. Zangwill, « A deterministic multiproduct, multi-facility production and inventory model », in : *Operations Research* 14.*3* (1966), p. 486-507.

[122] Yinlian Zeng, Xiaoqiang Cai et Hairong Feng, « Cost and emission allocation for joint replenishment systems subject to carbon constraints », in : *Computers & Industrial Engineering* 168 (2022), p. 108074.

GREYC

UNIVERSITÉ CAEN NORMANDIE

Titre : Un système d'aide à la décision pour le négoce de matériaux

Mot clés : système de distribution, optimisation multi-objectif, prédiction de vente, recherche opérationnelle, exécution limitée

Résumé : Cette thèse détaille la mise en œuvre d'un système d'aide à la décision pour le négoce de matériaux. Le contexte industriel dans lequel évolue ce négoce y est notamment décrit, et justifie les deux axes clé où des améliorations significatives peuvent être apportées : la prédiction de la demande, et l'optimisation du réapprovisionnement. Une exploration des solutions existantes dans ces deux domaines est proposée, et une discussion sur l'applicabilité des méthodes récentes rattachées à l'industrie 4.0 est menée. Il apparaît que les industries de tailles moyennes et petites n'ont ni le besoin ni les moyens de déployer les modèles issus du big data. Pour ces raisons, la thèse propose des processus économes en puissance de calcul, mêlant des méthodes traditionnelles bien connues avec des concepts plus récents pour circonscrire les prévisions et les réapprovisionnements autour de ce qui compte réellement pour le négoce de matériaux : des résultats fiables sur les produits clé, obtenus selon des durées opérationnelles réaliste. Pour ces raisons, le travail s'appuie principalement sur des expérimentations comme la prévision avec extraction de saisonnalité automatisée et le choix en amont du meilleur modèle prédictif parmi quatre : le modèle ARIMAX, la forêt aléatoire, le LSTM et une moyenne mobile. L'optimisation, elle, se voit accélérée par l'enchaînement de méthodes de résolution appuyées par de l'early-stopping et du warm-start, tout en tenant compte des nombreuses contraintes spécifiques au négoce. Quatre méthodes de résolution sont ainsi comparées : un algorithme glouton, un solveur quadratique, le recuit simulé et un algorithme génétique.

Title: A decision-support system for the material trade

Keywords: distribution system, multi-objective optimization, sales prediction, operational research, limited execution

Abstract: This thesis describes the implementation of a decision support system for the material trade. It describes the industrial context in which this trade operates, and justifies the two key areas where significant improvements can be made: demand prediction and replenishment optimization. Existing solutions in these two areas are explored, and the applicability of recent Industry 4.0 methods is discussed. It appears that small and medium-sized industries have neither the need nor the means to deploy Big Data models. For these reasons, this work proposes processes that save computing power, blending well-known traditional methods with more recent concepts to circumscribe forecasts and replenishments around what really matters for the material trade: reliable results on key products, obtained within realistic operational timescales. For these reasons, the study relies mainly

on experiments such as forecasting with automated seasonality extraction and upstream selection of the best predictive model from four: the ARIMAX model, the random forest, the LSTM and a moving average. Optimization, on the other hand, is accelerated by a sequence of resolution methods supported by early-stopping and warm-start, while taking into account the numerous constraints specific to this trading field. Four optimization methods are compared: a greedy algorithm, a quadratic solver, simulated annealing and a genetic algorithm.