

Optimisation du bin packing 2D dans les camions et conteneurs

Rapport bibliographique

Dexter S. METSANOU TSAFACK, M2RO, Université Toulouse III/ENAC, 2023-2024
Sous la supervision de:

Martin LAINÉE, RENAULT

October 29, 2024

Abstract

Le problème du bin packing 2D est un problème crucial dans de nombreux domaines, notamment dans la logistique, où il est nécessaire de maximiser l'utilisation de l'espace disponible dans les entrepôts ou les conteneurs. Cette problématique est devenue d'autant plus importante avec la croissance des quantités de production et l'augmentation des volumes de marchandises à stocker et à expédier. Dans ce contexte, l'optimisation des algorithmes de bin packing 2D est essentielle pour minimiser le gaspillage d'espace et réduire les coûts logistiques. Ce rapport propose une analyse approfondie de la littérature sur les techniques de bin packing 2D, mettant en lumière les différentes approches proposées pour résoudre ce problème complexe. Nous examinons les algorithmes heuristiques, les méthodes exactes et les approches hybrides développées pour optimiser le placement des objets dans des camions, des conteneurs ou des espaces de stockage.

1 Introduction

L'optimisation combinatoire est un domaine d'intérêt tant sur le plan théorique que pratique. D'un côté, il repose sur des concepts novateurs élaborés au cours du siècle dernier et continuellement enrichis par une vaste bibliographie. D'un autre côté, il connaît un regain d'intérêt significatif dans des domaines pratiques tels que la logistique, où il joue un rôle crucial. Par exemple, les problèmes de type bin-packing (BP), initialement étudiés dès les années 1950 dans un contexte unidimensionnel (BP1D), ont vu leur complexité évoluer au fil du temps. De nos jours, la littérature traite de problèmes en deux dimensions (BP2D), voire en trois dimensions (BP3D). Ce rapport bibliographique se concentre spécifiquement sur le BP2D.

Dans le BP2D, un ensemble d'articles rectangulaires $j \in J = \{1, \dots, n\}$ est donné, chacun ayant une largeur W_j et une hauteur H_j , ainsi que des conteneurs (bins) rectangulaires identiques, ayant une largeur W et une hauteur H . L'objectif est d'attribuer tous les articles à un nombre minimal de bins, sans chevauchement, en disposant les articles de manière à ce que leurs côtés soient parallèles à ceux des bins, et en supposant que les articles ont une orientation fixe.

Bien que ce problème semble simple, il est en réalité NP-difficile [10], ce qui signifie qu'il est extrêmement complexe à résoudre pour des instances de taille significative. Néanmoins, son importance est incontestable dans de nombreux domaines industriels, notamment dans la

découpe (industries du bois et du verre) et le packing (transport et entreposage). Certaines applications peuvent nécessiter des contraintes et/ou des hypothèses supplémentaires.

Ce rapport présente un aperçu des principaux travaux récents sur le BP2D, en mettant particulièrement l'accent sur les algorithmes exacts et les approches heuristiques et métaheuristiques. Il est structuré comme suit : la section 2 propose une description détaillée du problème considéré, tandis que la section 3 présente quelques algorithmes de base ayant des implications pertinentes sur ce problème. Ensuite, la section 4 détaille l'état de l'art concernant les algorithmes exacts et les approches heuristiques. L'axe de travail envisagé est présenté à la section 5. Enfin, la conclusion du rapport est présentée dans la section 6.

2 Définition du problème

2.1 Problématique générale

Chaque jour, Renault déploie des milliers de camions pour acheminer des pièces vers ses usines dans le monde entier, suivant un plan logistique prévisionnel élaboré par la direction, incluant les itinéraires de collecte et la fréquence des camions. Ce processus logistique représente un coût considérable, s'élevant à plusieurs centaines de millions d'euros par an. Dans le but de réduire ces dépenses, Renault examine comment optimiser sa chaîne d'approvisionnement, et cette optimisation passe par le chargement optimal des camions. Ce problème de chargement de camions et conteneurs implique le transport d'un vaste nombre d'articles des fournisseurs vers les usines en utilisant des camions. Les fournisseurs produisent des articles qui peuvent être chargés dans les camions à partir des quais. Chaque fournisseur peut avoir plusieurs quais et produire différents types d'articles. De manière similaire, chaque usine peut avoir plusieurs quais où les articles peuvent être déchargés des camions. La figure 1 illustre un exemple de trajets des camions.

Un ensemble prédéfini d'articles doit être transporté d'un quai de fournisseur spécifique à un quai d'usine spécifique. Chaque article doit être livré dans une fenêtre de temps déterminée, comprenant une heure d'arrivée au plus tôt et une heure d'arrivée au plus tard. Si un article est livré avant l'heure d'arrivée maximale, des frais d'inventaire sont encourus pour chaque jour avant cette date. Tous les articles chargés dans un camion doivent être empilés, et chaque pile peut contenir un ou plusieurs articles. Diverses contraintes régissent la manière dont les articles peuvent être empilés, telles que des restrictions sur les dimensions, le nombre maximal d'articles empilés, et la densité maximale d'une pile.

Une flotte de camions planifiée est disponible. Chaque camion peut visiter plusieurs fournisseurs prédéfinis et charger des piles à partir de leurs quais respectifs. Ensuite, le camion se rend à une usine spécifique où il peut décharger les piles sur différents quais. L'ordre des visites chez des fournisseurs et des quais est préétabli, et les piles doivent être chargées dans les camions dans le même ordre. Chaque camion a une liste d'articles qu'il peut collecter. Des contraintes de placement des piles et de poids maximum par camion sont également prises en considération. Chaque camion planifié a un coût associé, facturé uniquement s'il est utilisé.

L'objectif principal est de livrer tous les articles et de minimiser les coûts totaux. Une solution inclut la définition de toutes les piles, leur attribution et chargement à des camions spécifiques, la connaissance des coordonnées tridimensionnelles de chaque article et bidimensionnelles de chaque pile dans le camion, ainsi que la liste des camions utilisés avec leurs caractéristiques de chargement.

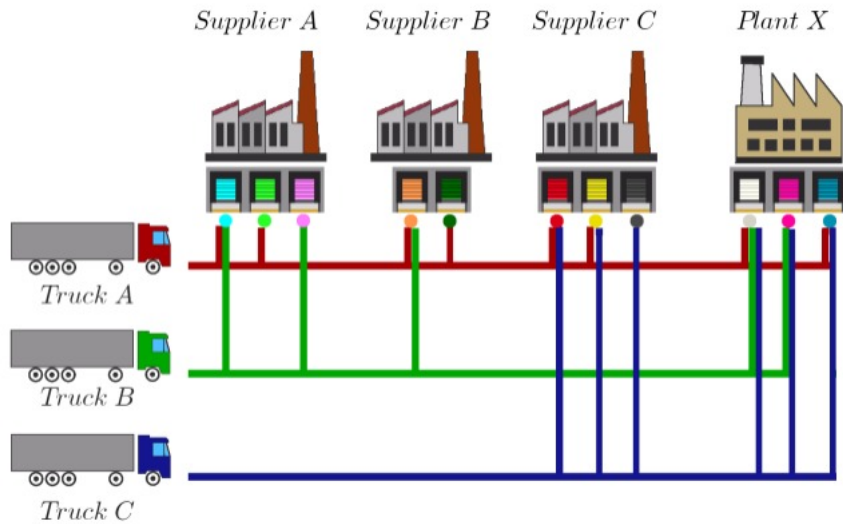


Figure 1: Exemple de trajets pris par trois camions différents étiquetés A, B et C. Le camion A s'arrête chez le Fournisseur A, où il visite à la fois les quais cyan et verts, puis il se dirige vers le Fournisseur B, s'arrêtant aux quais orange et vert foncé, avant de se rendre chez le Fournisseur C, où il s'arrête aux quais rouge et jaune. Enfin, il arrive à l'Usine X et s'arrête aux quais blanc et bleu-vert. Le camion B voyage depuis le Fournisseur A, s'arrêtant aux quais cyan et rose, avant de se rendre chez le Fournisseur B, où il s'arrête uniquement au quai orange. Ensuite, il arrive à l'Usine X et s'arrête aux quais blanc et rose. Le camion C se rend chez le Fournisseur C et s'arrête aux trois quais avant de se rendre à l'Usine X, où il visite également tous les quais disponibles. [12]

2.2 Objectif du stage

Dans le cadre de notre travail, nous concentrerons notre attention sur le défi de l'optimisation du chargement des camions, plus précisément sur le problème de placement en 2D, dans le but de maximiser leur capacité de chargement en utilisant les différentes piles définies au préalable. Pour résoudre ce problème, nous développerons des modèles mathématiques et mettrons en œuvre différents algorithmes.

3 Algorithmes de base

La plupart des algorithmes de la littérature sont de type glouton et peuvent être classés en deux familles :

- les algorithmes à une phase dans laquelle on charge directement les articles dans les bins finis.
- les algorithmes à deux phases dont la première phase consiste à charger les articles dans une seule bande, c'est-à-dire un bin ayant une largeur W et une hauteur infinie. Dans la deuxième phase, la solution de la bande est utilisée pour construire un packing dans des bins finis.

De plus, la plupart des approches sont des algorithmes de niveaux, c'est-à-dire que le packing du bin est obtenu en plaçant les articles, de gauche à droite, en rangées formant des niveaux.

Le premier niveau est le fond du bin, et les niveaux suivants sont produits par la ligne horizontale coïncidant avec le sommet de l'article le plus grand emballé sur le niveau inférieur. Trois stratégies classiques pour le packing par niveaux ont été dérivées à partir d'algorithmes célèbres pour le cas unidimensionnel. Dans chaque cas, les articles sont initialement triés par hauteur décroissante et chargés dans la séquence correspondante. Soit j le numéro de l'article actuel, et s le dernier niveau créé:

- Algorithme Next-Fit Decreasing Height (**NFDH**) : l'article j est placé à gauche sur le niveau s , s'il convient. Sinon, un nouveau niveau ($s := s + 1$) est créé, et j est placé à gauche pour ce nouveau niveau.
- Algorithme First-Fit Decreasing Height (**FFDH**) : l'article j est placé à gauche sur le premier niveau où il convient, le cas échéant. Si aucun niveau ne peut accueillir j , un nouveau niveau est initialisé comme dans NFDH.
- Algorithme Best-Fit Decreasing Height (**BFDH**) : l'article j est placé à gauche sur le niveau pour lequel l'espace horizontal inutilisé est minimal. Si aucun niveau ne peut accueillir j , un nouveau niveau est initialisé comme dans NFDH.

Ces différents Algorithmes sont illustrés à la figure 2.

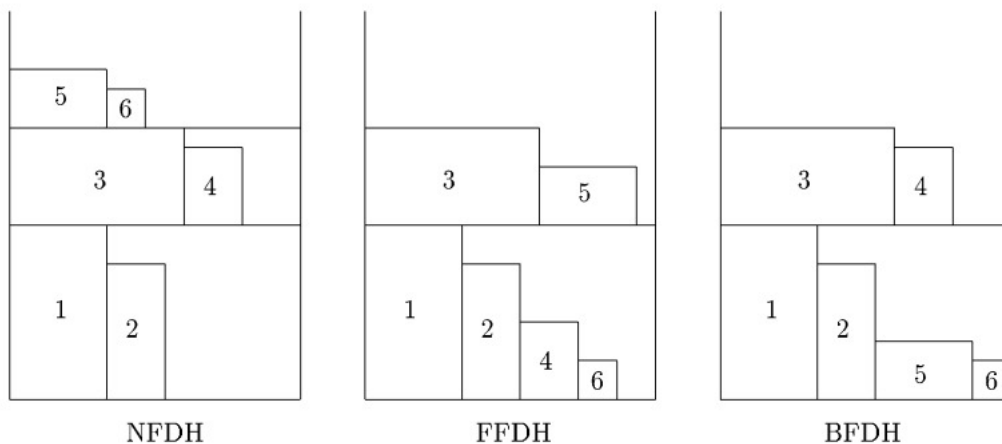


Figure 2: Algorithme de packing par niveaux [17]

3.1 Algorithme en une phase

Deux algorithmes en une phase ont été présentés et évalués expérimentalement par Berkey and Wang [3].

L'algorithme Finite Next-Fit (FNF) empaquette directement les articles dans des bins finis exactement de la même manière que l'algorithme HNF de la section 3.2.

L'algorithme Finite First-Fit (FFF) adopte plutôt la stratégie FFDH. L'élément actuel est placé sur le niveau le plus bas du premier bin où il s'insère. Si aucun niveau ne peut l'accueillir, un nouveau niveau est créé soit dans le premier bin approprié, soit en initialisant un nouveau bin (si aucun bin n'a assez d'espace vertical disponible).

3.2 Algorithmes en deux phases

Un algorithme en deux phases pour le problème de bin packing fini, appelé Hybrid First-Fit (HFF), a été proposé par Chung et al. [4]. Dans la première phase, un packing en bande [13] est obtenu par la stratégie FFDH. Une solution de bin packing fini est ensuite obtenue en résolvant de manière heuristique un problème de bin packing unidimensionnel (avec des tailles d'articles H_i et une capacité de bin H) à travers l'algorithme First-Fit Decreasing : initialiser le bin 1 pour charger le niveau 1, et, pour $i = 2, 3, \dots$, charger le niveau courant i dans le bin indexé le plus bas où il convient, le cas échéant. Si aucun bin ne peut accueillir i , initialiser un nouveau bin.

Chung et al. [4] ont prouvé que, si les hauteurs sont normalisées à un, alors $HFF(I) \leq \frac{17}{8} \cdot OPT(I) + 5$ où $OPT(I)$ est la valeur de la solution optimale.

Berkey and Wang [3] ont proposé et évalué expérimentalement un algorithme en deux phases, appelé Finite Best-Strip (FBS), qui est une variation de HFF. La première phase est réalisée en utilisant la stratégie BFDH. Dans la deuxième phase, le problème de bin packing unidimensionnel est résolu à travers l'algorithme Best-Fit Decreasing: charger le niveau actuel dans un bin disponible, en choisissant celui qui offre le moins d'espace vertical inutilisé parmi ceux où le niveau peut s'insérer, ou en créant un nouveau bin si nécessaire.

Frenk and Galambos. [9] ont proposé l'algorithme Hybrid Next-Fit (HNF) dont la performance dans le pire des cas est $HNF(I) \leq 3.382 \cdot OPT(I) + 9$ en supposant que les hauteurs et les largeurs sont normalisées à un.

Il existe également des algorithmes non basés sur les niveaux. La principale stratégie non basée sur les niveaux est connue sous le nom de Bottom-Left (BL), et consiste à placer l'article actuel dans la position la plus basse possible, à gauche. Baker et al.. [2] ont analysé les performances dans le pire des cas de l'algorithme résultant pour le problème de packing en bande [13], et ont prouvé que:

- si aucun ordre d'articles n'est utilisé, BL peut être arbitrairement mauvais.
- si les articles sont ordonnés par largeur décroissante alors $BL(I) \leq 3 \cdot OPT(I)$.

Berkey and Wang.[3] ont proposé l'approche BL pour le cas des bins finis. Leur algorithme Finite Bottom-Left (FBL) trie initialement les articles par largeur décroissante. L'article actuel est ensuite placé dans la position la plus basse à gauche de tout bin initialisé. Si aucun bin ne peut le contenir, un nouveau est initialisé.

Lodi et al. [16] ont proposé une approche différente non basée sur les niveaux, appelée directions alternatives (AD). L'algorithme initialise L bins (L étant une borne inférieure sur la valeur de la solution optimale) en plaçant sur leurs fonds un sous-ensemble de l'ensemble des articles, en suivant une politique BFDH. Les articles restants sont placés, un bin à la fois, dans des bandes, alternativement de gauche à droite et de droite à gauche. Dès qu'aucun article ne peut être placé dans l'une ou l'autre direction dans le bin actuel, le prochain bin initialisé ou un nouveau bin vide devient le bin actuel.

4 Etat de l'art

De nombreux travaux ont été réalisés sur le problème de bin packing en deux dimensions. Ils proposent des méthodes de résolution de ce problème, que ce soit à travers des algorithmes exacts ou des approches heuristiques.

4.1 Algorithmes exacts

Martello and Vigo [19] ont proposé une approche énumérative pour obtenir la solution exacte du BP2D. Leur méthode commence par trier les articles par ordre décroissant de leur aire, suivi d’une procédure de réduction visant à déterminer l’emballage optimal de certains bacs afin de réduire la taille de l’instance. Ensuite, une première solution, appelée solution initiale et notée z^* , est obtenue de manière heuristique. Cette méthode repose sur un schéma de branchement à deux niveaux : l’arbre de décision de la branche externe attribue chaque article à un bin sans spécifier sa position exacte, tandis que l’arbre de décision de la branche interne détermine un packing réalisable pour les articles actuellement assignés à un bin, éventuellement en énumérant tous les motifs possibles. Cependant cet algorithme génère de nombreuses redondances puisque le même article peut être packé à plusieurs reprises dans une position donnée pendant l’énumération.

Afin d’éviter d’explorer des solutions redondantes, Fekete and Schepers[8] introduisent un modèle théorique basé sur les graphes. Ils montrent qu’une classe de packing (c’est-à-dire un ensemble d’emballages ayant des propriétés communes) peut être associée à une paire de graphes d’intervalles. Ils proposent également un algorithme énumératif [7] pour construire de tels graphes d’intervalles. Leur méthode réduit considérablement le nombre de redondances par rapport à la méthode proposée par Martello and Vigo [19].

Clautiaux et al.[5] proposent deux méthodes de Branch and Bound pour le problème de packing orthogonal en deux dimensions (2OPP). Le premier algorithme (LMAO) est une amélioration de la méthode proposée par Martello and Vigo[19]. Au lieu de tester le packing des articles dans toutes les positions possibles, l’algorithme énumère les chargements des articles uniquement dans la position la plus basse à gauche. Le deuxième algorithme (TSBP) est une méthode de Branch and Bound en deux étapes basée sur une nouvelle relaxation du problème. Dans la première étape, toutes les solutions d’un problème relaxé sont énumérées. Pour chaque solution trouvée, une seconde méthode énumérative est utilisée pour rechercher une solution pour le problème initial correspondant à la solution actuelle. Cette méthode renvoie de meilleurs résultats sur plusieurs instances.

Han et al.[11] ont abordé une variante du problème BP2D, connue sous le nom de "multi-type two-dimensional bin packing problem" (M2BP). Dans cette variante, ils ont exploré trois approches distinctes : une heuristique gloutonne simple appelée "First Fit by Ordered Deviation" (FFOD), une méthode basée sur le recuit simulé (SA) et un algorithme exact basé sur la génération de colonnes avec Branch and Bound (CG). Pour l’approche CG, ils ont d’abord transformé le M2BP en un problème de couverture d’ensemble, puis ont suivi les étapes suivantes : ils ont commencé par la phase de génération de colonnes, où ils ont créé le problème de couverture d’ensemble initial à partir de la solution FFOD, appelé le Problème Maître Restreint (RMP), et ont résolu ce problème maître relaxé en utilisant la méthode de génération de colonnes. Cette étape est répétée jusqu’à ce qu’aucune colonne avec un coût réduit négatif ne puisse être générée. Ensuite, si la solution est entière, l’algorithme se termine, sinon, on passe à l’étape de Branch and Bound pour poursuivre la résolution. SA et FFOD seront décrits dans la section 4.2.

4.2 Approches heuristiques

Ces dernières années, les techniques de métaheuristique sont devenues un outil populaire pour résoudre de manière approximative les problèmes d’optimisation combinatoire difficiles. Lodi et al.[15, 18] ont développé des algorithmes de recherche tabou efficaces pour le BP2D et certaines de ses variantes. La recherche tabou est caractérisée par l’utilisation d’un schéma de recherche et d’un voisinage indépendants du problème de packing spécifique à résoudre. Ce cadre peut ainsi être appliqué à pratiquement toutes les variantes du BP2D. À partir d’une solution courante,

les mouvements modifient le packing d'un sous-ensemble S d'articles pour vider un bin cible spécifié. Le bin cible est sélectionné en minimisant une fonction dépendant du contenu actuel des bins. Une fois le bin cible déterminé, un nouveau chargement pour S est obtenu en exécutant un algorithme heuristique approprié. Si le mouvement permet de vider le bac cible, un nouvel article est sélectionné pour un nouveau chargement. Sinon, le voisinage est élargi en augmentant la taille du paramètre k , qui définit la taille et la structure du voisinage. Une solution initiale est obtenue en exécutant l'algorithme heuristique sur l'ensemble des articles, tandis que la solution de recherche tabou initiale consiste à charger un article par bin. L'exécution de l'algorithme s'arrête dès qu'une solution optimale est trouvée ou qu'une limite de temps est atteinte.

Liu et al.[14] ont proposé une heuristique pour résoudre le BP2D. Cette heuristique reformule le BP2D en un schéma de programmation dynamique, visant à minimiser la zone de gaspillage totale des bins utilisés. Pour un ensemble de N items, représenté par un vecteur B de dimension N , la zone de gaspillage totale pour α items packés dans θ bacs est déterminée par l'équation de récurrence $\gamma(\alpha, \theta) = \min_{\alpha' < \alpha} \{\gamma(\alpha', \theta - 1) + h(\alpha, \alpha', \theta)\}$, où $h(\alpha, \alpha', \theta)$ est le coût de transition de l'état $(\alpha', \theta - 1)$ à l'état (α, θ) . Ici, α est défini comme $|B|$. La récurrence est basée sur le calcul de la fonction de transition et est résolue par une heuristique dont l'idée principale consiste à essayer de remplacer un petit élément contenu dans la solution actuelle par un élément plus grand non inclus dans la solution.

L'algorithme FFOD introduit à la section 4.1 proposé par, Han et al.[11] pour le M2BP, renvoie une solution qui est soit satisfaisante, soit fournir une solution réalisable initiale dont la valeur fournit une borne supérieure z_{ub} et un point de départ pour l'algorithme SA ou CG. Il considère séquentiellement les objets et attribue l'objet soit à un bin existant, soit à un nouveau bin. L'attribution se fait en fonction du "coût d'opportunité"[11] minimum. Cette définition du "coût d'opportunité" permet d'emballer les objets de manière compacte dans les bins tout en maintenant l'équilibre des charges de ressources sur chaque bin. Pour l'algorithme SA[11], l'espace des solutions, \mathcal{F} , est l'ensemble de toutes les partitions possibles des N objets en k sous-ensembles et de toutes les affectations possibles de types de bins à chaque sous-ensemble au sein d'une partition. Les partitions de \mathcal{F} ne sont pas contraintes à celles qui permettent uniquement des sous-ensembles faisables d'objets (c'est-à-dire des sous-ensembles d'objets pouvant être assignés à au moins l'un des types de bacs disponibles sans violer aucune contrainte de ressources). Un état de solution, S , est toute partition. Un sous-ensemble de S est noté s . Le mécanisme de génération d'états de solution et de structure du voisinage est fait par un schéma de transfert à sens unique. En considérant un état de solution S , le transfert à sens unique génère un nouvel état en déplaçant un objet d'un bin à un autre ou en déplaçant un objet vers un nouveau bin. Tous les transferts à sens unique possibles à partir d'un état donné définissent la structure du voisinage. Le fonctionnement général de l'algorithme est le suivant: il commence par obtenir une solution initiale S à l'aide de l'heuristique FFOD, puis détermine la température initiale T_0 et la fonction de refroidissement $g(T)$. Ensuite, il exécute une boucle principale jusqu'à ce que la température soit figée ou qu'un critère d'arrêt soit atteint. À chaque itération, il génère un voisinage aléatoire S' de S et évalue la variation d'énergie Δ entre les deux solutions. Si $\Delta < 0$, il accepte toujours le voisinage S' comme la nouvelle solution courante S . Sinon, il évalue la probabilité d'accepter le voisinage S' en fonction de la température actuelle T et de Δ , et décide de l'accepter avec une certaine probabilité. La température est ensuite mise à jour selon la fonction de refroidissement $g(T)$. L'algorithme se termine par la sortie de la meilleure solution trouvée S^* et de sa valeur z_{ub} , qui est utilisée comme borne supérieure de la solution optimale.

Abeysooriya et al.[1] présentent une étude sur le BP2D irrégulier avec plusieurs bins homogènes (2DIBPP). Ils proposent une méthode heuristique basée sur l'algorithme "Jostle" [6]

initialement conçu pour le problème de packing en bandes pour résoudre le 2DIBPP. Ils explorent également différentes stratégies de rotation des pièces et de critères de placement, révélant que certains choix peuvent générer de meilleurs résultats que d'autres. Les performances des algorithmes qu'ils proposent sont évaluées sur différents ensembles d'instances avec des formes convexes et non convexes, démontrant leur efficacité pour résoudre différentes variantes du BP2D.

Martinez-Sykora et al.[20] présentent un modèle de programmation en nombres entiers pour déterminer l'association entre les articles et les bins, suivi d'un modèle de programmation en nombres entiers mixtes pour placer les articles dans les bins. Leur méthode prend en compte la rotation des articles. Les résultats indiquent que cette approche offre de bonnes performances en termes de qualité lorsqu'elle est évaluée sur divers ensembles d'instances présentant différentes propriétés.

5 Axe de travail envisagé

Après avoir présenté le problème du bin packing 2D et examiné différentes approches de résolution existantes, nous envisageons d'explorer, comme axe de travail, l'utilisation d'un algorithme de recherche arborescente heuristique. Cette méthode d'optimisation repose sur l'exploration d'un arbre de recherche et comprend deux composantes clés :

- Un schéma de branchement : représentant l'espace de recherche sous la forme d'un arbre de décision implicite.
- Un algorithme de recherche arborescente : explorant cet arbre de recherche de manière intelligente pour visiter en priorité les régions les plus prometteuses.

Nous envisageons que cette algorithme puisse être intégrée à nos outils d'optimisation du chargement des camions et des conteneurs. Plus précisément, elle pourrait remplacer l'algorithme interne de bin packing 2D dans les cas où elle se révélerait plus efficace.

6 Conclusion

Cette analyse bibliographique offre une vue approfondie des différentes approches utilisées dans le domaine du bin packing 2D. Elle met en évidence l'importance cruciale de développer des algorithmes efficaces pour maximiser l'utilisation de l'espace. Les différentes méthodes examinées, allant des heuristiques aux approches exactes, illustrent la diversité des solutions possibles pour résoudre ce problème complexe. Cette synthèse souligne les progrès réalisés dans le domaine du bin packing 2D et met en avant les opportunités prometteuses pour l'optimisation des opérations logistiques.

References

- [1] Ranga P. Abeysooriya, Antonio. Martinez-Sykora, and Julia A Bennell. Jostle heuristics for the 2d-irregular shapes bin packing problems with free rotation. *International Journal of Production Economics*, 195:12–26, 2018.
- [2] B S Baker, E G Colman Jr., and R L Rivest. Orthogonal packing in two dimensions. *SIAM Journal on Computing*, 9:846–855, 1980.
- [3] J.O. Berkey and P.Y. Wang. Two-dimensional finite bin packing algorithms. *J. Oper. Res. Soc.*, 38:423–429, 1987.

- [4] F K R Chung, M R Garey, and D S Johnson. On packing two-dimensional bins. *SIAM Journal on Algebraic and Discrete Methods*, 3:66–76, 1982.
- [5] François Clautiaux, Jacques Carlier, and Aziz Moukrim. Exact method for the two-dimensional orthogonal packing problem. *Elsevier Science*, April 2005.
- [6] K.A. Dowsland, W.B. Dowsland, and J.A. Bennell. Jostling for position: local improvement for irregular cutting patterns. *J. Oper. Res. Soc.*, 49(6):647–658, 1998. ISSN 0160-5682.
- [7] Sándor Fekete and Jörg Schepers. An exact algorithm for higher-dimensional orthogonal packing. *Operations Research*, Volume(Number):Pages, Year. Revised for publication.
- [8] Sándor P Fekete and Jörg Schepers. A combinatorial characterization of higher-dimensional orthogonal packing. *Mathematics of Operations Research*, 29(2):353–368, 2004.
- [9] J.B. Frenk and G.G. Galambos. Hybrid next-fit algorithm for the two-dimensional rectangle bin-packing problem. *Computing*, 39:201–217, 1987.
- [10] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman & Co., New York, NY, USA, 1979.
- [11] Bernard T Han, George Diehr, and Jack S Cook. Multiple-type, two-dimensional bin packing problems: Applications and algorithms. *Annals of Operations Research*, 50(1-4): 239–261, 1994.
- [12] Tomáš Hromada. Roadeff challenge 2022: Optimization of truck fleet loading. Master’s thesis, CZECH TECHNICAL UNIVERSITY, May 2023.
- [13] E. G. Coffman Jr., M. R. Garey, D. S. Johnson, and R. E. Tarjan. Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, 9: 801–826, 1980.
- [14] Ya Liu, Chengbin Chu, and Kanliang Wang. A new heuristic algorithm for a class of two-dimensional bin-packing problems. *International Journal of Advanced Manufacturing Technology*, 57(9-12):1235–1244, 2011. doi: 10.1007/s00170-011-3351-1.
- [15] A Lodi, S Martello, and D Vigo. Approximation algorithms for the oriented two-dimensional bin packing problem. *European Journal of Operational Research*, 112:158–166, 1999.
- [16] A Lodi, S Martello, and D Vigo. Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *INFORMS Journal on Computing*, 11:345–357, 1999.
- [17] A. Lodi, S. Martello, and D. Vigo. Recent advances on two-dimensional bin packing problems. *Discrete and Applied Mathematics*, 123:379–396, 2002.
- [18] A Lodi, S Martello, and D Vigo. Neighborhood search algorithm for the guillotine non-oriented two-dimensional bin packing problem. In Stefan Voss, Silvano Martello, Ibrahim H. Osman, and Gérard Roucairol, editors, *Metaheuristics: Progress in Complex Systems Optimization*, pages 139–150. Springer, 2004.
- [19] Silvano Martello and Daniele Vigo. Exact solution of the two-dimensional finite bin packing problem. *Management Science*, 44(3):388–399, 1998.

- [20] A. Martinez-Sykora, R. Alvarez-Valdes, J.A. Bennell, R. Ruiz, and J.M. Tamarit. Matheuristics for the irregular bin packing problem with free rotations. *European Journal of Operational Research*, 258(2):440–455, 2017.