



HAL
open science

Planification des activités d'une flotte de robots mobiles autonomes pour la logistique interne de systèmes de production

Elouan Blanchard, Cyrille Briand, Mohamed Amine Abdeljaouad

► To cite this version:

Elouan Blanchard, Cyrille Briand, Mohamed Amine Abdeljaouad. Planification des activités d'une flotte de robots mobiles autonomes pour la logistique interne de systèmes de production. 19ème Colloque national S.mart, Pôle S.mart Rhône-Alpes Ouest, May 2025, Vogüé, France. ⟨hal-05207636⟩

HAL Id: hal-05207636

<https://hal.science/hal-05207636v1>

Submitted on 12 Aug 2025

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

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



HAL Authorization



Planification des activités d'une flotte de robots mobiles autonomes pour la logistique interne de systèmes de production

E. Blanchard^{a, b, *}, C. Briand^a, Mohamed Amine Abdeljaouad^b

^a LAAS-CNRS, Université de Toulouse, CNRS, UT3, Toulouse, France

^b Direction de l'Innovation – Groupe ALTEN, Toulouse, France

* e-mail : elouan.blanchard@laas.fr

1. INTRODUCTION

Cet article s'intéresse à la planification au niveau tactique des activités d'une flotte de robots dans le contexte de la logistique interne des systèmes de production. Deux types de robots pour la logistique interne sont généralement considérés : les véhicules à guidage automatique (AGV) et les robots mobiles autonomes (AMR). Il s'agit dans les deux cas de robots capable d'évoluer dans un environnement in-door disposant d'une autonomie énergétique supposée ici suffisante pour accomplir les missions planifiées.

Les AGVs sont des robots à relativement faible coût. Ils suivent des chemins fixés par avance au sein de l'espace de production pour transporter leur chargement, ce qui requiert généralement des modifications d'infrastructure telles que l'intégration dans le sol de pistes magnétiques. Les AMRs constituent une nouvelle famille de robots un peu plus onéreuse qui exploitent des capteurs embarqués plus sophistiqués (caméras, lidars) et des algorithmes de navigation avancés, en particulier le célèbre algorithme SLAM (Simultaneous Localization and Mapping). Cette instrumentation confère aux robots la capacité de naviguer dans des environnements non contrôlés sans avoir besoin de suivre des chemins ou des pistes fixes.

Les AMRs sont plus flexibles que les AGVs puisqu'ils sont autonomes et permettent de détecter les obstacles et les contourner, en choisissant en temps réel le trajet alternatif. Leur navigation intelligente leur permet de se déplacer en toute sécurité dans des environnements dynamiques. Ils jouent un rôle important dans le développement de systèmes de production plus intelligents (voir [1]) car ils sont capables de s'adapter avec agilité aux changements de contexte de la production. Ils peuvent effectuer un large éventail

d'opérations logistiques telles que l'approvisionnement des postes de travail en kits de composants, le déplacement d'outils de production spécifiques, la collecte de conteneurs vides ou l'évacuation de déchets. Ils peuvent également être utilisés pour transporter des produits semi-finis entre les postes de production, cela exigeant parfois le montage sur le robot d'une interface mécanique adaptée au produit à transporter.

Dans ce nouveau contexte, la capacité à synchroniser la planification des opérations de la flotte de robots avec la conception du plan de production devient de plus en plus cruciale pour favoriser l'agilité du système de production. En effet, le retard d'une tâche de production, une panne de machine ou un défaut de qualité peuvent nécessiter une adaptation du plan de production, ce qui induit un impact sur le routage des robots puisque la date d'approvisionnement en composants peut se trouver retardée et les dates de déplacements des semi-produits reprogrammés. En cas de défaut, de nouveaux composants doivent parfois être livrés de façon imprévue. Inversement, lors de la mise en œuvre du plan de routage, les AMRs peuvent également subir des retards inattendus lorsqu'il s'agit d'éviter les collisions (en cas d'encombrement par exemple), si un itinéraire est temporairement fermé ou si les opérations de chargement/déchargement effectuées sur le robot prennent plus de temps que prévu, ce qui peut induire une mise à jour du plan de production. En conséquence, dans un environnement dynamique, le défi consiste donc à maintenir en temps réel la cohérence entre les plans de production sur les machines et celui de routage des robots.

Le travail décrit dans cet article fait suite à celui présenté dans [2] (dont nous rappelons les principales contributions) et se focalise en particulier sur la

problématique d'acheminement des produits semi-finis dans l'atelier, cela dans un contexte de job shop flexible. Nous montrons comment tirer parti, lors de la planification du routage des robots, de la flexibilité temporelle naturellement présente dans l'ordonnancement de production. Plus précisément, supposant initialement connu un ordonnancement de production optimal (déterminé par un solveur), nous montrons comment associer à chaque opération réalisée par une machine un intervalle de temps de livraison et de collecte, puis comment en déduire un plan de routage de la flotte de robot cohérent avec le plan de production initial sans, ni rompre sa cohérence, ni altérer sa performance. Nous considérons ici un environnement déterministe, l'objectif étant de construire un plan de routage prévisionnel, minimisant le work-in-progress (WIP) et satisfaisant les contraintes de logistique interne requises pour mettre en œuvre le plan de production.

Ce document est organisé comme suit. Le chapitre 2 propose une analyse de la littérature relative à notre problème, en motivant notre positionnement. La partie 3 montre comment notre problème peut être défini de manière générique comme un problème de Pick-up-and-Delivery (PDP) avec fenêtres temporelles (PDPTW). Une méthodologie de résolution fondée sur l'utilisation d'un solveur de programmation par contraintes (PPC), d'un programme linéaire en variables mixtes (MILP) et d'une heuristique est proposée dans le chapitre 4. La partie 5 décrit les résultats d'une première expérimentation menée sur un jeu de données de Job Shop Flexible avec temps de transport (issu de la littérature.) Quelques conclusions et perspectives de travail sont présentées dans la dernière section.

2. ÉTAT DE L'ART

Les problèmes d'acheminement par véhicules (VRP) constituent une famille de problèmes NP-difficiles largement étudiée en optimisation combinatoire. Ils ont de nombreuses applications pratiques car ils modélisent une grande variété de problèmes de transport, qu'il s'agisse de transport de personnes, de marchandises ou d'informations. D'une manière générale, il s'agit d'organiser la livraison de ressources à des clients sur une période donnée, en fonction de leurs besoins, à l'aide de véhicules localisés dans un ou plusieurs dépôts, cela en minimisant le coût total du transport et en respectant les contraintes logistiques (retour au dépôt, livraison à temps, capacités des véhicules). L'ouvrage de Toth et Vigo [3] fournit un aperçu complet de l'état de l'art, ainsi qu'une description des méthodes et des solutions pour ce problème et ses variantes. Une extension importante du VRP est le problème VRP avec fenêtres temporelles (VRPTW), où chaque livraison doit être effectuée dans un intervalle de temps spécifié, incluant éventuellement un temps de déchargement [4].

Le VRP avec enlèvement et livraison est une autre extension importante du VRP [5][6]. Appelé Pick-up and delivery problem (PDP), il diffère du VRP en ce que les

marchandises ou les personnes peuvent avoir plusieurs origines et plusieurs destinations, contrairement au VRP où toutes les marchandises sont stockées au dépôt. Certains clients ont donc des requêtes négatives pour indiquer qu'ils produisent des articles. Il peut y avoir plusieurs types d'articles à collecter et à livrer. Les véhicules ont une capacité qui doit être respectée. Comme les articles peuvent être chargés pendant la tournée, les contraintes de précédence ne sont pas vérifiées a priori et les articles doivent être différenciés, ce qui n'est pas nécessaire dans le VRP. Il existe deux types principaux de PDP, selon que l'on transporte des marchandises ou des personnes. Les PDPs pour les marchandises consistent à trouver des itinéraires élémentaires pour chaque véhicule qui minimisent le coût total du transport tout en respectant les contraintes de collecte et de livraison aux clients [18]. On distingue : i) les problèmes many-to-many, où chaque article est disponible à plusieurs endroits et doit être livré à plusieurs destinations, ii) les problèmes one-to-many-to-one, où certains articles proviennent d'un seul dépôt et doivent être livrés aux clients, et où d'autres articles doivent être collectés auprès des clients puis livrés au dépôt, et iii) les problèmes one-to-one, où chaque article a une seule origine et une seule destination. C'est dans cette dernière catégorie de problème que ce travail s'inscrit en considérant que chaque enlèvement ou livraison doit être effectué dans une fenêtre de temps spécifique, problème nommé dans la littérature PDP with time window (PDPTW). Les approches de résolution utilisant la génération de colonnes se sont avérées très efficaces pour résoudre les instances PDPTW de taille réaliste [8], notamment les instances dynamiques [9]. Li et Lim [10] proposent une métaheuristique pour résoudre le PDPTW et six ensembles de données de grande taille pour évaluer leur algorithme. Plus récemment, dans [11], une matheuristique est proposée qui fonctionne bien sur les instances de Li-et-Lim, ainsi que sur un nouvel ensemble d'instances.

L'existence d'opérations de transport entre les activités de production a également été étudiée dans la littérature de l'ordonnancement. Dans [12], Ulusoy et al. considèrent le Job Shop Problem with Transportation Times (JSPT). Ce problème implique de planifier simultanément les machines et d'acheminer les AGVs. Classiquement, l'objectif est de minimiser la durée totale de l'ordonnancement. Les auteurs proposent un algorithme génétique évalué sur un ensemble d'instances de problèmes librement disponibles, que nous avons utilisées dans ce travail. Deroussi et al. [13] proposent trois métaheures différentes (une recherche locale itérée, une procédure de recuit simulé et une approche hybride) qui exploitent une structure de voisinage reposant sur les itinéraires suivis par les AGVs. Un graphe disjonctif étendu et une recherche locale sont introduits par Lacomme et al. dans [14] pour modéliser efficacement à la fois le problème et les solutions, qui peuvent également être utilisés pour traiter un job-shop flexible avec plusieurs robots de transport. Dans [15], Gondran et al. montrent que, lorsque la qualité de service (QoS) est

prise en compte en plus du makespan, le problème devient proche du Dial-A-Ride Problem (DARP) où les opérations de ramassage et de livraison doivent être planifiées. Les auteurs proposent une approche de résolution itérative qui minimise d'abord le délai d'exécution, puis la qualité de service. Le problème d'ordonnancement flexible de job-shop avec des ressources de transport est considéré dans [16] où, en utilisant une modélisation de graphe disjonctif, les auteurs proposent une procédure de recherche tabou qui explore un grand ensemble de mouvements en temps constant. Il convient de noter qu'une étude récente des problèmes d'ordonnancement avec des contraintes de transport est proposée dans [28], qui passe en revue plus de 160 articles et introduit un nouveau système de classification en trois champs pour les caractériser.

Examinons les avantages et les inconvénients des approches précédentes dans le contexte de la logistique de production interne. Tout d'abord, nous remarquons que le problème d'approvisionnement d'une ligne de production invalide plusieurs hypothèses classiques du PDPTW. En effet, les temps de chargement/déchargement ne sont plus négligeables par rapport aux temps de transport puisque les itinéraires entre les postes de travail peuvent être très courts. Par ailleurs, les problèmes de transfert de produits semi-finis sortent du cadre du PDPTW classique puisque le temps séparant la livraison d'un produit semi-fini à une machine de sa collecte, doit être suffisant pour permettre la réalisation de l'opération. L'environnement JSPT semble également trop restrictif puisque, à l'inverse, il ne considère que le transport des produits semi-finis, mais pas le problème de l'approvisionnement des postes de travail en composants. De plus, comme indiqué dans [24], pour des raisons de complexité, le nombre de véhicules pris en compte dans JSPT est souvent limité à 1 ou 2, et leur capacité est supposée unitaire, ce qui semble restrictif par rapport aux pratiques industrielles réelles. Enfin, alors que minimiser les en-cours (WIP) est l'objectif fréquent en logistique interne (en respectant au mieux le plan de production), le PDPTW s'intéresse davantage à minimiser les coûts de transport, qui dépendent essentiellement de la longueur des itinéraires suivis par les véhicules et du nombre de véhicules. Le JSPT s'intéresse également à la minimisation du makespan, qui, bien qu'important, n'est pas central dans le contexte de la logistique interne.

Dans la section suivante, nous proposons de modéliser notre problème en utilisant le formalisme PDPTW, qui nous permet de considérer les différents types d'opérations de logistique interne de manière générique en tenant compte des durées opératoires.

3. DEFINITION FORMELLE DU PROBLEME

Nous supposons ici connaître l'ordonnancement de production (i.e., les affectations des opérations aux machines et leur séquençage ont été décidés) et la topologie de l'atelier de production, il est ainsi possible de dresser la liste des actions logistiques à accomplir pour assurer le bon déroulement de la production. Le problème

peut alors être représenté par un graphe complet orienté $G = (V, A)$. V est l'ensemble des sommets, partitionné en deux sous-ensembles : les sommets de collecte $P = \{1, \dots, n\}$ et ceux de livraison $D = \{n+1, \dots, 2n\}$, les sommets 0 et $2n+1$ étant les sommets d'origine et de fin de tournée de chaque véhicule (il est possible d'avoir le même lieu associé à plusieurs sommets pour représenter des collectes ou livraisons ayant lieu au même endroit mais à des moments distincts). A est l'ensemble des arêtes $(i, j) \in A$, chacune étant caractérisée par un temps de transport t_{ij} . Chaque nœud $i \in V$ doit être servi durant un intervalle de temps $[a_i, b_i]$, le temps de service dure s_i et la quantité requise est q_i (positive pour les collectes et négative pour les livraisons). La flotte est composée d'un ensemble K de k véhicules ($K = \{1, \dots, k\}$) et chaque véhicule $k \in K$ a une capacité de transport Q_k . Finalement, L désigne la longueur de l'horizon de planification. Ainsi, $[a_i, b_i] \in [0; L]$, $\forall i \in V$. L'ensemble $O \subseteq D \times P$ représente les opérations, chaque élément $(i, j) \in O$ étant formé d'un sommet de livraison $i \in D$ et d'un sommet de collecte $i \in P$, i et j étant localisés sur la même machine. On associe la durée opératoire $p_{i,j}$ à chaque élément $(i, j) \in O$. Sous ces hypothèses, le problème peut se modéliser sous la forme du programme linéaire en nombres entiers (PLNE) ci-dessous, inspiré du Pickup and Delivery Problem with Time Windows et de la formulation proposée par Toth et Vigo, dénoté PPDPTW (pour Production PDPTW).

(PPDPTW)

$$\min \left(\Delta \sum_{j \in P} \sum_{j \in P} x_{0jk} + \max_{i \in V} (b_i - (T_i + s_i)) \right) \quad (1)$$

$$s. t. \sum_{k \in K} \sum_{j \in D} x_{ijk} = 1 \quad \forall i \in P \quad (2)$$

$$\sum_{j \in D} x_{ijk} = \sum_{j \in D} x_{n+i,jk} \quad \forall i \in P, k \in K \quad (3)$$

$$\sum_{j \in P} x_{0jk} \leq 1 \quad \forall k \in K \quad (4)$$

$$\sum_{j \in P} x_{ijk} = \sum_{j \in P} x_{jik} \quad \forall i \in P \cup D, k \in K \quad (5)$$

$$T_{n+i} \geq T_i + s_i + t_{i,n+i} \quad \forall i \in P \quad (6)$$

$$a_i \leq T_i + s_i \leq b_i \quad \forall i \in V \quad (7)$$

$$T_j - (T_i + s_i) \geq p_{i,j} \quad \forall (i, j) \in O \quad (8)$$

$$T_j - T_i \geq M_{ij} + (s_i + t_{ij} - M_{ij})x_{ijk} \quad \forall i \in V, j \in V, k \in K \quad (9)$$

$$0 \leq Q_{ij} \leq \sum_{k \in K} Q_k x_{ijk} \quad \forall i \in V, j \in V \quad (10)$$

$$\sum_{j \in P} Q_{ji} - \sum_{j \in P} Q_{ij} = q_i \quad \forall i \in V \quad (11)$$

$$x_{ijk} \in 0,1 \quad \forall i \in V, j \in V, k \in K \quad (12)$$

$$Q_{ij}, T_i \in R \quad \forall i \in V, j \in V \quad (13)$$

Les variables T_i indiquent la date de visite du sommet $i \in V$. Les variables x_{ijk} sont égales à 1 si le véhicule k emprunte l'arc (i, j) , 0 sinon. Une formulation par flot pour les quantités transportées est adoptée où Q_{ij} indique la quantité transportée sur l'arc $(i, j) \in A$. La fonction objectif (1) minimise lexicographiquement le nombre de tournées de véhicules nécessaire pour satisfaire les objectifs de production, et ainsi la taille de la flotte de robots (la valeur Δ est choisie suffisamment grande pour qu'il soit toujours avantageux de réduire le premier terme de la fonction objectif par rapport au second). C'est ensuite l'écart maximal entre la date de visite et la fin de la fenêtre temporelle de livraison des articles qui est minimisé, dans une logique de minimisation du WIP. Les contraintes (2)-(5) sont inchangées par rapport à la formulation initiale de Toth et Vigo. Les contraintes (2) imposent que chaque demande soit traitée exactement une fois. Les contraintes (3) imposent que le véhicule assurant la collecte soit le même que celui assurant la livraison. Chaque véhicule ne peut faire qu'une seule tournée (contraintes (4)) et le flot entrant sur un sommet i est égal au flux sortant (contraintes (5)). Les contraintes temporelles (6)-(9) font le lien entre les variables T_i et x_{ijk} . Les contraintes (8) imposent que le temps de séjour sur une machine d'un produit semi-fini soit supérieur ou égale à la durée opératoire. Elles sont nécessaires pour tirer parti de la flexibilité temporelle naturellement présente dans le plan de production. Les intervalles $[a_i, b_i]$ et $[a_j, b_j]$ correspondent alors aux dates de début et de fin au plus tôt et au plus tard des opérations. Les valeurs M_{ij} sont des bornes supérieures suffisamment grande de la valeur $T_j - T_i$. Les contraintes (10)-(11) font le lien entre les quantités Q_{ij} véhiculées sur les arcs et les variables x_{ijk} .

4. METHODOLOGIE DE RESOLUTION PROPOSEE

Afin de résoudre le problème PPDPTW par le PLNE de la section 3, il est nécessaire de fixer préalablement un ordonnancement et de calculer les marges temporelles associées aux opérations. La connaissance de la topologie de l'atelier de production est elle-même nécessaire dans la construction de l'ordonnancement afin de connaître les durées minimums d'acheminement des pièces. Afin d'obtenir en résultat un plan de production optimisé et un plan d'acheminement des produits par les AMRs permettant de le réaliser, nous appliquons une méthodologie de résolution en 4 étapes.

4.1. Réalisation d'un plan de production avec les opérations de transport intégrées

Nous considérons des problèmes de type job shop flexible avec transport. Chaque job est ainsi composé d'opérations réalisables sur un ensemble prédéfini de machines. Du fait de l'existence de contraintes de transport entre les machines, nous ajoutons des opérations de transport avant la première opération de chaque job, à la fin de la dernière opération et entre chaque opération intermédiaire.

La première étape de la méthodologie est de résoudre le job shop flexible de façon optimale en prenant en compte les temps de transports minimaux entre les machines. Nous utilisons pour cela le solveur, hybridant programmation par contraintes et apprentissage de clauses, nommé Aries [18]. Un modèle est fourni par l'auteur permettant de résoudre des instances de job shop flexible. Nous l'avons adapté pour intégrer la prise en compte des temps de transport, cela afin d'obtenir un plan de production tout à la fois réalisable du point de vue de la logistique interne et optimal vis-à-vis du makespan. Nous obtenons ainsi un plan de production optimal considérant l'existence d'une flotte sans limite de capacité.

4.2. Calcul des marges temporelles

A partir de ce plan il nous est possible d'obtenir des fenêtres de temps des opérations correspondant à leur date de début au plus tôt et au plus tard, d'une part, et de fin au plus tôt et au plus tard, d'autre part. Ces dates sont utiles pour offrir de la flexibilité temporelle aux opérations de transport (un produit semi-fini devant subir une opération non critique sur une machine, peut être déposé sur cette machine entre la date de début au plus tôt et au plus tard, sans que cela n'altère le makespan).

Pour calculer les fenêtres de temps allouées aux opérations, nous utilisons les affectations des opérations aux machines proposées par le solveur Aries, ainsi que le séquençement de ces opérations sur les machines. Nous appliquons ensuite l'algorithme de Bellmann qui permet de calculer les dates au plus tôt et au plus tard des opérations en $O(n \log n)$ (le graphe étant sans circuit il suffit d'appliquer un tri topologique pour trouver l'ordre dans lequel les sommets doivent être visités pour le calcul des chemins les plus longs).

1.1. PPDPTW

Le modèle PPDPTW a été implémenté dans un solveur de PLNE (GUROBI). Sa résolution permet d'obtenir un plan de livraison des produits semi-finis qui réduit au maximum le nombre de robots à utiliser afin de répondre au plan de production sans dégrader le makespan et en tirant partie de sa flexibilité temporelle naturellement présente. Il optimise également le WIP dans le sens où les opérations de livraison sont réalisées juste-à-temps (c'est-à-dire le plus près possible de la date de livraison au plus tard).

Les premières expérimentations ont permis de valider la cohérence du modèle. Par contre, le solveur peut être très lent à trouver une solution initialement réalisable.

1.2. Première résolution heuristique

Afin de palier le problème précédent, une heuristique gloutonne a été développée pour déterminer une solution réalisable initialement. Cette heuristique est décrite dans [2] et consiste à sélectionner à chaque itération une tâche de pick-up-and-delivery et de l'insérer dans le planning de

roulage à l'endroit considéré le plus avantageux. Pour cela, l'heuristique visite le plan de roulage de chaque robot (une borne supérieure est initialement définie) et énumère toutes les positions d'insertion possibles afin de les évaluer, puis de sélectionner la meilleure. Un nouveau robot est utilisé dès lors qu'aucune position réalisable d'insertion n'est trouvée. La construction d'une solution initiale est donc très rapide. Une telle solution peut alors nourrir le solveur de PLNE à son initialisation, cela afin d'accélérer la résolution et de permettre au PLNE de l'améliorer pour ensuite trouver une solution optimale (en termes de nombres de robots, puis de WIP).

2. EXPERIMENTATION

L'expérimentation a été réalisée sur les instances de Job Shop Flexible déjà considérée dans [13]. Le jeu de données est constitué de 10 instances et une matrice définissant les temps de transport entre chaque machine (non symétrique). Pour chaque instance nous avons appliqué notre méthodologie de résolution à trois étapes : le problème de job-shop flexible est d'abord résolu de façon optimale par le solveur Aries en supposant une capacité de transport illimitée, puis nous utilisons notre heuristique pour définir un plan de roulage réalisable utilisant un nombre k_h de véhicules, enfin nous résolvons le PLNE PPDTW à optimalité en lui fournissant le plan de roulage déterminé par l'heuristique grâce au solveur GUROBI. La table 1 résume les résultats obtenus. Pour Chaque instance, nous reportons, son nom, le nombre de jobs de l'instance, le nombre total d'opérations, le nombre k_h de véhicules utilisé dans la solution heuristique, le nombre optimal de véhicules déterminé par le solveur lors de la résolution de la formulation PPDTW, la valeur optimale du WIP (correspondant à la minimisation de la plus grande avance de livraison des produits semi-finis), le temps de résolution utilisé par le solveur pour résoudre à optimalité le problème d'optimisation PPDTW, la valeur optimale du makespan C_{max}^* déterminée par Aries (inchangée par PPDTW) et enfin, la valeur $C_{max}^{[13]}$ du makespan déterminée par la méthodologie de résolution proposée dans [13] (en supposant que seuls deux véhicules de capacité unitaire sont considérés). Remarquons que nous n'avons pas reporté les temps de calcul du solveur Aries (inférieurs à quelques secondes sur chacune des instances), ni celui de l'heuristique dont l'exécution est quasi-instantanée.

Tableau 1 – Résultats expérimentations job shop flexible avec transport

Inst.	#jobs	#opés	k_h	k^*	WIP	T_{cpu} (s)	C_{max}^*	$C_{max}^{[13]}$
Jobset1	7	19	9	4	24	6	143	160
Jobset2	6	15	7	4	18	2	117	138
Jobset3	6	16	7	4	36	3	131	142
Jobset4	5	19	11	5	14	2	111	138
Jobset5	5	13	7	3	36	19	105	112
Jobset6	6	18	7	4	24	3	145	158
Jobset7	8	19	10	5	16	250	105	150
Jobset8	6	20	7	3	49	50	187	197
Jobset9	5	17	8	4	6	3	147	166
Jobset10	6	17	7	5	16	3	177	188

Une première observation positive est que le temps nécessaire à Gurobi pour résoudre à optimalité chacune des instances est plutôt court, excepté pour les instances 7 et 8 qui comptent davantage d'opérations. Cet enthousiasme doit être modéré par le fait que le nombre d'opérations reste tout de même faible dans ce jeu de données. On remarque également que la formulation PDPTW permet de réduire le nombre de véhicules utilisés par l'heuristique quasiment de moitié. Cette observation indique aussi que l'heuristique est relativement peu efficace quant à la minimisation des robots. La solution PDPTW étant optimale, on peut aussi assurer qu'il est impossible de baisser davantage le nombre de véhicules sans détériorer le makespan. Le tableau permet de préciser cette dégradation puisque, quand le nombre de véhicules est réduit à 2, comme dans [13], le makespan augmente en moyenne de 12%. Il serait certainement intéressant de pouvoir étudier plus finement l'impact d'une diminution de la flotte de robots sur la dégradation du makespan en traçant le front de Pareto de ces deux objectifs. Enfin, la mesure WIP permet de constater que l'optimisation profite effectivement de la flexibilité temporelle naturellement présente dans le plan de production, ce qui autorise certainement une diminution supplémentaire du nombre de véhicules.

3. CONCLUSION

Cet article a décrit une méthodologie en 3 étapes permettant de calculer un plan de roulage optimal pour une flotte de robots AMRs chargée d'assurer la logistique interne d'un atelier de production. Il s'est focalisé sur un environnement relativement générique de job shop flexible. Le plan de roulage est construit de sorte à respecter la performance d'un plan de production initial (déterminé à l'aide du solveur Aries) en tirant bénéfice de la flexibilité temporelle naturellement présente dans le plan (i.e., les marges temporelles associées aux opérations). La détermination du roulage est réalisée par résolution d'un PLNE minimisant le nombre de robots nécessaires (en priorité), puis le travail en cours (dans un second temps). L'approche permet donc de déterminer le nombre minimal de robots nécessaires permettant d'éviter toute dégradation du plan de production. Des expérimentations menées sur un jeu de données académique ont permis de valider la cohérence de l'approche et de mettre en évidence sa performance sur des instances de taille relativement moyenne.

Une perspective de cette étude est d'intégrer complètement la planification du plan de roulage des robots (i.e., la logistique interne) à la planification de la production (i.e., l'ordonnancement), cela afin d'optimiser conjointement ces deux plans, comme cela est proposé dans [16]. Il sera ainsi possible d'étudier la dégradation du makespan lorsque la taille de la flotte diminue pour atteindre progressivement plus qu'un seul robot (et de tracer le front de Pareto). À plus long terme, nous aimerions également adapter une méthode de branch-and-price pour la résolution du PPDTW, afin de les résoudre de manière dynamique en adaptant le plan en

temps réel et permettre de réagir aux aléas de la production et de la logistique, comme cela est proposé dans [9].

REMERCIEMENTS

Nous tenons particulièrement à remercier Laurent Deroussi, enseignant-chercheur au Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes de Clermont-Ferrand, pour nous avoir partagé ses instances de job shop flexible avec temps de transport et fourni ses résultats.

REFERENCES

- [1] Rodrigo Bernardo, João M. C. Sousa, and Paulo J. S. Gonçalves. Survey on robotic systems for internal logistics. *Journal of Manufacturing Systems*, 65:339–350, October 2022.
- [2] Elouan Blanchard, Arthur Bit-Monnot, Cyrille Briand, Grégoire Milliez, Mohamed Amine Abdeljaouad. Planning the tasks of an autonomous mobile robot fleet for internal logistics of production systems. *APMS 2024 CONFERENCE*, Sep 2024, Germany.
- [3] Vigo Daniele and Toth Paolo, editors. *Vehicle Routing*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, November 2014.
- [4] Guy Desaulniers, Oli B.G. Madsen, and Stefan Ropke. Chapter 5: The Vehicle Routing Problem with Time Windows. In *Vehicle Routing*, MOS-SIAM Series on Optimization, pages 119–159. Society for Industrial and Applied Mathematics, November 2014.
- [5] Guy Desaulniers, Jacques Desrosiers, Andreas Erdmann, Marius M. Solomon, and François Soumis. 9. VRP with Pickup and Delivery. In *The Vehicle Routing Problem, Discrete Mathematics and Applications*, pages 225–242. Society for Industrial and Applied Mathematics, January 2002.
- [6] Jan Dethloff. Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up. *OR Spektrum*, 23:79–96, 02 2001.
- [7] Maria Battarra, Jean-François Cordeau, and Manuel Iori. Chapter 6: Pickup-and-Delivery Problems for Goods Transportation. In *Vehicle Routing*, MOS-SIAM Series on Optimization, pages 161–191. Society for Industrial and Applied Mathematics, November 2014.
- [8] Stefan Ropke and Jean-François Cordeau. Branch and Cut and Price for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, 43(3):267–286, 2009.
- [9] Martin Savelsbergh and Marc Sol. Drive: Dynamic Routing of Independent Vehicles. *Operations Research*, 46(4):474–490, August 1998.
- [10] H. Li and A. Lim. A metaheuristic for the pickup and delivery problem with time windows. *Proceedings 13th IEEE International Conference on Tools with Artificial Intelligence. ICTAI 2001*, pages 160–167, 2001.
- [11] Carlo S. Sartori and Luciana S. Buriol. A study on the pickup and delivery problem with time windows: Matheuristics and new instances. *Computers & Operations Research*, 124:105065, December 2020.
- [12] Gündüz Ulusoy, Funda Sivrikaya-Şerifoglu, and Ümit Bilge. A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. *Computers & Operations Research*, 24(4):335–351, April 1997.
- [13] Laurent Deroussi, Michel Gourgand, and Nicolas Tchernev. A simple metaheuristic approach to the simultaneous scheduling of machines and automated guided vehicles. *International Journal of Production Research*, 46(8):2143–2164, April 2008.
- [14] Philippe Lacomme, Mohand Larabi, and Nikolay Tchernev. Job-shop based framework for simultaneous scheduling of machines and automated guided vehicles. *International Journal of Production Economics*, 143(1):24–34, May 2013.
- [15] Matthieu Gondran, Marie-José Hugué, Philippe Lacomme, Alain Quilliot, and Nikolay Tchernev. A Dial-a-Ride evaluation for solving the job-shop with routing considerations. *Engineering Applications of Artificial Intelligence*, 74:70–89, September 2018.
- [16] Lucas Berterottière, Stéphane Dauzère-Pérès, and Claude Yugma. Flexible job-shop scheduling with transportation resources. *European Journal of Operational Research*, 312(3):890–909, February 2024.
- [17] Amir Hosseini, Alena Otto, and Erwin Pesch. Scheduling in manufacturing with transportation: Classification and solution techniques. *European Journal of Operational Research*, 315(3):821–843, 2024.
- [18] Arthur Bit-Monnot. Enhancing Hybrid CP-SAT Search for Disjunctive Scheduling. In *ECAI 2023*, pages 255–262. IOS Press, 2023.