

Optimisation du routage des conteneurs dans les réseaux de transport maritime de lignes

Yousra El Ghazi Djamal Habet Cyril Terrioux

Aix Marseille Univ, CNRS, LIS, Marseille, France

prénom.nom@univ-amu.fr

Résumé

Étant donné un réseau de lignes maritimes, le problème de routage de conteneurs consiste à déterminer quelles routes maritimes vont emprunter les conteneurs pour être acheminés de leur port de chargement à leur port de destination. Ce problème d'optimisation sous contraintes constitue une tâche essentielle dans le cadre du transport maritime de marchandises, que ce soit au niveau de la conception des réseaux ou dans leur mise en œuvre opérationnelle. Il existe, dans la pratique et dans la littérature, plusieurs variantes de ce problème, qui se distinguent notamment par le point de vue considéré, les contraintes prises en compte et le critère à optimiser.

Dans cet article, nous traitons plusieurs variantes de ce problème à l'aide de la programmation par contraintes. Nous proposons, d'abord, deux modélisations sous la forme d'instances COP. Puis, dans un second temps, nous décrivons une méthode de recherche locale. Enfin, nous comparons expérimentalement nos deux modèles et la méthode proposée.

Abstract

Given a maritime line network, the container routing problem consists of determining which maritime routes containers will take to be transported from their loading port to their destination port. This constrained optimization problem is an essential task in the context of maritime goods transportation, whether at the network design level or in operational implementation. In practice and in the literature, there are several variants of this problem, which differ in terms of the point of view considered, the constraints taken into account and the criterion to be optimized.

In this article, we address several variants of this problem using constraint programming. First, we propose two models in the form of COP instances. Then, in a second step, we describe a local search method. Finally, we experimentally compare our two models and the proposed method.

1 Introduction

Le transport maritime conteneurisé joue un rôle crucial dans le commerce mondial, comme l'atteste le rapport de la Conférence des Nations Unies sur le commerce et le développement (CNUCED). Plus de 80 % du volume des échanges internationaux est actuellement acheminé par voie maritime, avec 1,95 milliard de tonnes transportées sous forme de conteneurs. Les marchandises conteneurisées sont transportées via des réseaux de transport maritime réguliers, empruntant des itinéraires cycliques appelés *services* ou *rotations*. Ces services visitent un ensemble de ports prédéterminés, selon un ordre établi et à des horaires réguliers. Les compagnies maritimes disposent d'une flotte de navires déployés sur ces différents services. Sur chaque service, un nombre prédéfini de navires homogènes opèrent à des fréquences régulières. Cependant, la gestion efficace de ces flux de conteneurs représente un défi complexe. En effet, acheminer des conteneurs de manière économique au sein d'un réseau maritime est une question cruciale dans le transport maritime régulier. Plusieurs itinéraires peuvent être envisagés pour le transport de conteneurs entre un port d'origine (POL) et un port destination (POD) donnés. Ces trajets peuvent varier en fonction de différents critères, tels que le temps de transit, le nombre de transbordements nécessaires, la capacité des navires et la fréquence des services. Ces choix impactent directement les coûts de transport, un enjeu majeur pour la compétitivité des compagnies maritimes.

Répondre à la question du routage optimal des conteneurs est l'objectif de cet article. Étant donné un réseau maritime défini par un ensemble de rotations desservant régulièrement un ensemble de ports dans un ordre déterminé, ainsi que la demande de conteneurs entre ces différents ports, le problème consiste à déterminer les routes optimales pour acheminer les conteneurs de leur port d'origine à leur destination finale. L'op-

timalité peut être appréciée selon différents critères comme maximiser les revenus engendrés par les marchandises transportées, maximiser la quantité totale de marchandises acheminées ou encore minimiser les distances parcourues par les conteneurs.

Dans la littérature, diverses variantes du problème sont étudiées en tenant compte de différentes hypothèses et propriétés. Ce problème est très similaire à celui de flot multi-commodités (MCF [9]). Étant donné un réseau candidat, le problème consiste à décider, parmi les conteneurs disponibles, lesquels doivent être acheminés et, le cas échéant, selon quels itinéraires. Ce problème est généralement formulé sous forme de modèles de programmation linéaire où le nombre de conteneurs est traité comme une variable de décision continue. Différentes études ont proposé diverses formulations pour modéliser le routage des conteneurs. Certaines de ces formulations se basent sur le flux de lien entre les origines et les destinations des conteneurs (formulations O-D [1, 4]), où l'on essaie de déterminer la quantité de conteneurs passant par chaque lien du réseau entre chaque paire d'origine et de destination. D'autres formulations se basent sur les origines des conteneurs [2, 18], où l'on essaie de déterminer comment les conteneurs sont acheminés à partir de différents points de départ. Il existe également des formulations basées sur des segments [3, 12], où l'on considère des séquences de liens consécutifs dans le réseau. En outre, certaines études adoptent des formulations basées sur les chemins [4, 17, 18, 19, 22, 21], où l'accent est mis sur la sélection des chemins optimaux pour les conteneurs, où chaque variable indique un chemin pour un certain conteneur. Dans [4], Brouer et al. résolvent le problème d'allocation de conteneurs avec repositionnement des conteneurs vides à l'aide d'une approche de génération de colonnes. Ils optimisent des instances à grande échelle, mais excluent l'optimisation de la vitesse et les temps de transit de leur formulation. Dans [22], Wang et al. présentent un modèle mathématique pour générer les trajets des conteneurs tout en tenant compte du temps de transit O-D et des contraintes de cabotage maritime. La vitesse des navires est supposée constante et aucune contrainte de capacité n'est imposée. Ils ont résolu leur modèle de programmation linéaire en nombres entiers sur 166 ports à l'aide d'IBM CPLEX. Dans [10], les auteurs formulent le problème comme un problème MCF avec contrainte de temps, en imposant une limite de temps de transit maximal pour chaque conteneur. Sans tenir compte de la conception du réseau ou fournir des détails sur la façon dont les navires spécifiques sont sélectionnés et affectés aux différentes rotations, ils se concentrent uniquement sur la façon dont les marchandises circulent à travers le réseau tout en minimisant les coûts de transport. Après avoir proposé des formulations en flux d'arc et en flux de chemin, ils constatent que la formulation en flux de chemin est plus rapide que la formulation en flux d'arc

en utilisant un algorithme de génération de colonnes.

Bien que des avancées aient été réalisées dans la résolution de ce problème, le déploiement des navires est généralement considéré comme une donnée d'entrée plutôt qu'une décision sujette à l'optimisation. Cependant, dans [13], les auteurs vont au-delà en proposant une extension d'un modèle de programmation linéaire en nombres entiers qui intègre l'optimisation du nombre de navires et des classes de navires pour chaque rotation. Face à la diversité des problèmes, dans cet article, nous abordons plusieurs variantes à l'aide de la programmation par contraintes. Afin de transporter un maximum de marchandises, nous envisageons la possibilité de diviser une marchandise en plusieurs parties. Notre objectif est alors de déterminer les routes empruntées par chacune de ces parties, en nous basant sur les rotations préétablies. Nous proposons deux modèles sous forme d'instances COP ainsi qu'une méthode de recherche locale. Ces trois approches peuvent exploiter n'importe lequel des trois critères d'optimisation mentionnés précédemment et ne nécessitent pas de fixer le type de navires au préalable.

Cet article est structuré comme suit. La section 2 introduit les notions nécessaires à la compréhension de l'étude et explore plus en détail le problème. Ensuite, dans les sections 3 et 4, nous présentons deux modélisations sous forme d'instances COP. Puis dans la section 5, nous décrivons une méthode de recherche locale. Enfin, nous procédons à une comparaison expérimentale de nos deux modèles ainsi que de cette méthode, avant de tirer nos conclusions dans la section 7

2 Préliminaires

2.1 Programmation par contraintes

Une instance *COP* (*Constraint Optimization Problem* [15]) P se définit par la donnée d'un quadruplet (X, D, C, f) . $X = \{x_1, \dots, x_n\}$ est un ensemble de n variables. À chaque variable x_i , on associe un domaine D_{x_i} , issu de $D = \{D_{x_1}, \dots, D_{x_n}\}$, qui contient les valeurs possibles pour x_i . $C = \{c_1, \dots, c_e\}$ désigne un ensemble de e *contraintes* qui traduisent les interactions entre les variables et définissent les combinaisons de valeurs autorisées. Enfin, f spécifie le critère à optimiser. Résoudre une instance *COP* $P = (X, D, C, f)$ revient à essayer de trouver une affectation de toutes les variables de X satisfaisant toutes les contraintes de C et optimisant le critère donné par f . Il s'agit d'un problème NP-difficile.

Un des atouts de la programmation par contraintes réside dans l'existence de contraintes spécialisées (les *contraintes globales*) qui vont faciliter la modélisation des problèmes, mais aussi, leur résolution grâce à des algorithmes de filtrage dédiés. Par la suite, nous exploiterons les contraintes globales suivantes :

- $\text{Elt}(Y, i) = k$ qui assure que la i ème valeur de Y est égale à k (Y pouvant être ici un ensemble de variables ou de valeurs),
- $\text{Ordered}(Y, \odot)$ qui impose $y_1 \odot y_2 \odot \dots \odot y_{|Y|}$.
- $\text{Sum}(Y, \Lambda) \odot k$ qui impose que la somme des valeurs de Y pondérées par les coefficients de Λ satisfait la condition imposée par la relation $\odot$ vis-à-vis de k . Par la suite, cette contrainte sera représentée sous la forme plus explicite $\sum_i \lambda_i \cdot y_i \odot k$.

2.2 Problèmes de routage de conteneurs

Depuis son introduction révolutionnaire dans les années 1950, le conteneur a transformé le paysage du transport maritime, offrant une méthode standardisée et efficace pour transporter une variété de marchandises. Il permet une manipulation rapide, un chargement facile entre différents modes de transport, et une protection accrue des marchandises du fait de son étanchéité. Il y a principalement deux tailles de conteneurs : les conteneurs de 20 pieds de long (environ 6,1 m) et les conteneurs de 40 pieds (12,2 m), pour une hauteur de 8,6 pieds (2,6 m) et une largeur de 8 pieds (2,4 m). Aussi, l'espace de stockage des navires est divisé en unités de 40 pieds, permettant d'y empiler des conteneurs de 40 pieds ou de 20 pieds. L'unité de mesure standard pour compter une quantité de conteneurs est l'équivalent vingt pieds ou EVP (Twenty-foot Equivalent Unit ou TEU en anglais), où un conteneur de 40 pieds équivaut à 2 EVP. Il existe plusieurs catégories de conteneurs. Les principales sont :

- les conteneurs *standards* (ou *dry*) : ces conteneurs sont les plus couramment utilisés. Ils sont adaptés au transport de marchandises sèches comme les textiles, les produits électroniques, les meubles, etc.
- les conteneurs *réfrigérés* (ou *reefer*) : conçus pour le transport de marchandises sensibles à la température telles que les produits alimentaires périssables, les produits pharmaceutiques, etc. Ces conteneurs sont équipés de systèmes de contrôle de la température pour maintenir des conditions spécifiques tout au long du voyage. Le maintien d'une température adéquate nécessite alors de les connecter au réseau électrique du bateau.
- les conteneurs *vides* (ou *empty*) : après avoir été vidés de leurs marchandises, ces conteneurs doivent être acheminés vers des ports pour lesquels le nombre de conteneurs sortant est supérieur au nombre de conteneurs entrant afin d'être à nouveau remplis. Leur gestion doit être efficace pour minimiser les temps d'inactivité et les coûts associés et permettre un envoi régulier de marchandises.
- les conteneurs *premium* : le client peut opter pour un conteneur *premium* moyennant un supplément, souvent important, pour garantir une

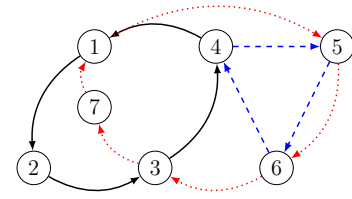


FIGURE 1 – Réseau avec trois rotations

livraison plus rapide de sa cargaison.

Plusieurs problèmes de planification à différents niveaux ont été étudiés dans le domaine de l'optimisation des réseaux de transport maritime. La prise de décision peut être analysée à trois niveaux distincts : la planification stratégique, tactique et opérationnelle [1]. La *planification stratégique* vise à définir la configuration optimale de la flotte, en tenant compte du nombre et de la taille des navires. L'objectif est de minimiser les coûts en capital et d'exploitation, des aspects cruciaux dans l'industrie du transport maritime où la possession de navires représente un investissement significatif [8, 20]. La phase de *planification tactique* s'intéresse à la planification des navires (ship-scheduling). À ce stade, la conception du réseau de services s'effectue par l'élaboration d'itinéraires maritimes et l'affectation des navires disponibles à ces itinéraires [1, 12, 5, 8, 10]. Enfin, lors de la phase de *planification opérationnelle*, les cargaisons à transporter et les itinéraires à utiliser pour leur expédition sont déterminés [17, 16, 11, 14]. Le transporteur peut décider de ne pas acheminer certaines cargaisons en raison de leur manque de rentabilité, préférant peut-être transporter d'autres cargaisons, issues ou à destination d'autres ports, qui sont plus rentables. Ce problème est également connu sous le nom de problème de routage des marchandises ou des cargaisons. La complexité de la logistique maritime réside dans les interdépendances entre ces niveaux de décision, les frontières entre le stratégique et le tactique, et entre le tactique et l'opérationnel, ne sont pas toujours claires. Dans cet article, nous nous intéressons au problème de routage de conteneurs qui consiste à déterminer comment attribuer efficacement les flux de cargaison aux différents services disponibles dans le réseau prédéfini. La figure 1 montre un réseau de transport maritime composé de trois rotations.

Considérons un ensemble $\mathcal{R}$ de rotations. Chaque rotation r de $\mathcal{R}$ est définie par une suite s_r de ports ($s_r = (p_1^r, p_2^r, \dots, p_{|s_r|}^r)$) et un ensemble $\mathcal{V}_r$ de type de navires compatibles avec les ports visités. Notons qu'un port peut apparaître plusieurs fois, par exemple, dans le cas de service complexe.

D'un point de vue pratique, on suppose que chaque rotation maintient une fréquence de service hebdomadaire et est desservie par une flotte homogène de navires présentant des caractéristiques similaires. Par

exemple, tous les navires de type v peuvent transporter jusqu'à $\kappa(v)$ conteneurs (exprimé en EVP) et disposent de $\rho(v)$ prises électriques pour les conteneurs réfrigérés. Chaque marchandise k est vue comme une quantité $q(k)$ de conteneurs (exprimée en EVP) à transporter du port d'origine $pol(k)$ au port de destination $pod(k)$, avec un revenu $rev(k)$ par EVP et une nature $nat(k)$ (premium, réfrigéré, standard ou vide). Pour optimiser le transport de marchandises, il est possible de diviser une marchandise en s parties, si nécessaire, chaque partie empruntant une route différente. La route suivie par une marchandise (ou une de ses parties) est définie par la succession des ports et rotations par lesquels elle transite. La route peut être directe quand le port d'origine et celui de destination de la marchandise se trouvent sur une même rotation. Dans le cas contraire, la marchandise devra transiter à travers plusieurs rotations. Le passage d'une rotation à une autre est une opération appelée *transbordement*, qui s'effectue dans un port commun aux deux rotations. On note $ts_{max}(k)$ le nombre maximal de transbordements autorisés pour la marchandise k . Ce nombre dépend généralement de la nature de la marchandise. Par exemple, les conteneurs premium et les conteneurs réfrigérés sont rarement transbordés, dans la pratique, afin de garantir les délais de livraison et le respect des températures.

Le problème de routage de conteneurs dans un réseau de transport maritime de ligne peut être défini comme suit : Étant donné un réseau prédéfini $\mathcal{R}$ de rotations desservant régulièrement un ensemble $\mathcal{P}$ de ports dans un ordre déterminé, la demande $\mathcal{K}_0$ de conteneurs entre ces différents ports, et un ensemble $\mathcal{V}$ de types de navires compatibles avec les ports visités. Le problème consiste à déterminer quelles routes maritimes vont emprunter les conteneurs pour être acheminés de leur port d'origine à leur port de destination. Les objectifs peuvent varier en fonction des besoins, allant de la maximisation des revenus générés par le transport des marchandises à l'optimisation de la quantité totale de marchandises acheminées, ou encore à la minimisation des distances parcourues par les conteneurs. La plupart des travaux de la littérature se concentrent sur un seul type de navire par rotation et ne prennent en compte que les conteneurs standards. Notre approche opère dans un cadre plus général. En effet, nous traitons quatre types différents de conteneurs et travaillons avec un ensemble de types de navires compatibles avec les ports visités de la rotation. Ce dernier point permet ainsi d'utiliser notre approche aussi bien dans la phase de conception d'un réseau que dans sa phase opérationnelle. De plus, nous permettons la division des marchandises en plusieurs parties, ce qui facilite le transport d'un maximum de marchandises.

La résolution de ce problème trouve des applications diverses et cruciales dans la gestion et la planification des opérations logistiques. Elle permet aux compagnies maritimes d'optimiser leurs réseaux, en identifiant les

$$\sum_{j \in \mathcal{K}_k^s} q_j \leq q(k) \quad k \in \mathcal{K}_0 \quad (\text{M.1})$$

$$\text{Ordered}(\{q_j | j \in \mathcal{K}_k^s\}, \geq) \quad k \in \mathcal{K}_0 \quad (\text{M.2})$$

$$\alpha_j = 1 \iff q_j > 0 \quad j \in \mathcal{K} \quad (\text{M.3})$$

FIGURE 2 – Les contraintes relatives aux marchandises

meilleures routes pour acheminer les conteneurs de manière plus efficace et rentable. En outre, en comprenant les demandes de transport à long terme et les schémas de flux de conteneurs, les investissements dans de nouveaux navires peuvent être planifiés de manière plus précise, minimisant ainsi les risques financiers et assurant une capacité adéquate pour répondre à la demande future. Une autre utilisation possible est liée à la simulation d'événements comme, par exemple, l'impossibilité d'emprunter le canal de Suez, afin de trouver la réponse la plus adéquate possible, éventuellement en acheminant les marchandises différemment.

3 Premier modèle

Le problème de routage des conteneurs étant un des sous-problèmes constituant le problème de conception de lignes (problème LSNDP [6]), ce premier modèle est inspiré d'une partie du modèle présenté dans [7]. Il permet d'aller au-delà en prenant en compte les services complexes. L'idée de ce premier modèle (noté $\mathcal{M}_1$) est de déduire les routes empruntées par les marchandises à partir des rotations.

3.1 Marchandises

Afin de transporter autant de marchandises que possible, nous considérons la possibilité qu'une marchandise soit divisée en s parties. Chacune de ces parties peut alors être transportée indépendamment des autres (c'est-à-dire en utilisant des routes différentes). Soit $\mathcal{K}$ l'ensemble des parties obtenues en divisant chaque marchandise de $\mathcal{K}_0$ en s parties. Nous notons $\mathcal{K}_k^s$ l'ensemble des parties relatives à la marchandise k . Chaque partie possède les mêmes POL, POD et nature que la marchandise initiale. Notons que la somme des nombres de EVP des parties de la marchandise k ne peut excéder $q(k)$ et qu'une partie peut être vide.

Pour chaque partie j d'une marchandise k , nous introduisons une variable entière q_j à valeur dans $[0, q(k)]$ afin de représenter le nombre d'EVP associé à la partie j et une variable booléenne α_j indiquant si la partie est utilisée ou non. La contrainte (M.1) (voir la figure 2) garantit que le nombre d'EVP transportés pour la marchandise k ne dépasse pas $q(k)$. La contrainte (M.2) permet de casser certaines symétries tandis que la contrainte (M.3) garantit la cohérence entre les valeurs des variables α_j et q_j .

3.2 Flux de marchandises

Pour chaque partie j d'une marchandise k , nous considérons une variable nbr_j ayant pour domaine $[0, ts_{max}(k) + 1]$ afin d'exprimer le nombre total de rotations utilisées pour transporter la partie j . Pour déterminer les ports où chaque partie est chargée ou déchargée, nous introduisons une variable booléenne $from_{k,i}^r$ (resp. $to_{k,i}^r$) qui vaut 1 si la partie j est chargée (resp. déchargée) dans le port p_i^r de la rotation r , 0 sinon.

La contrainte (F.1) (resp. (F.2)) de la figure 3 garantit qu'une partie j est transportée si et seulement s'il existe une rotation dans laquelle elle est chargée dans son port $pol(j)$ (resp. déchargée dans son port $pod(j)$). Le nombre de rotations utilisées pour transporter une partie j correspond au nombre de ports dans laquelle la partie j est (dé)chargée (contraintes (F.3) et (F.4)). Pour un port p_i^r , une partie ne peut être chargée et déchargée à la fois (contrainte (F.5)). Une partie j est chargée (resp. déchargée) dans un port p donné au plus une fois si elle est acceptée, 0 sinon (contraintes (F.6) et (F.7)). Si une partie j est chargée (resp. déchargée) dans un port p_i^r distinct de $pol(j)$ (resp. $pod(k)$), alors elle doit être déchargée (resp. chargée) dans le même port, mais, pour une autre rotation (contraintes (F.8) et (F.9)). La contrainte (F.10) garantit que, pour chaque rotation, une partie j est chargée et déchargée autant de fois. Soit d_r le nombre maximum d'occurrences d'un même port dans la rotation r . d_r est supérieur à 1 dans le cas de rotations complexes. Pour chaque rotation r , une partie j ne peut être (dé)chargée plus de d_r fois (contraintes (F.11) et (F.12)).

Enfin, il est nécessaire de savoir quelles sont les parties transportées à la sortie de chaque port de chaque rotation. Pour cela, nous ajoutons une variable booléenne $leave_{j,i}^r$ qui vaut 1 si la partie j est transportée par la rotation r à la sortie du port p_i^r , 0 sinon. Les contraintes (F.13)-(F.17) permettent de mettre en concordance les valeurs des variables $from_{j,i}^r$, $to_{j,i}^r$ et $leave_{j,i}^r$. La contrainte (F.18) garantit qu'une partie j ne peut pas passer plusieurs fois par le même port. De même, une partie j ne peut pas quitter son port de destination (contrainte (F.19)).

3.3 Capacités des navires

Nous devons garantir qu'à la sortie de chaque port, les navires ne dépassent pas leurs capacités en termes de nombre de conteneurs transportés et de nombre de prises utilisées pour les conteneurs réfrigérés. Les valeurs maximales de ces deux quantités n'étant pas corrélées (en particulier, le navire pouvant transporter le plus de conteneurs n'étant pas nécessairement celui ayant le plus de prises), nous devons garantir la cohérence de ces deux quantités en fonction des types de navires disponibles. Pour cela, nous introduisons,

$$\sum_{r \in \mathcal{R}} \sum_{p_i^r \in s_r | p_i^r = pol(j)} from_{j,i}^r = \alpha_j \quad j \in \mathcal{K} \quad (F.1)$$

$$\sum_{r \in \mathcal{R}} \sum_{p_i^r \in s_r | p_i^r = pod(j)} to_{j,i}^r = \alpha_j \quad j \in \mathcal{K} \quad (F.2)$$

$$\sum_{r \in \mathcal{R}} \sum_{p_i^r \in s_r} from_{j,i}^r = nbr_j \quad j \in \mathcal{K} \quad (F.3)$$

$$\sum_{r \in \mathcal{R}} \sum_{p_i^r \in s_r} to_{j,i}^r = nbr_j \quad j \in \mathcal{K} \quad (F.4)$$

$$from_{j,i}^r + to_{j,i}^r \leq 1 \quad j \in \mathcal{K}, r \in \mathcal{R}, p_i^r \in s_r \quad (F.5)$$

$$\sum_{r \in \mathcal{R}} \sum_{p_i^r \in s_r | p_i^r = p} from_{j,i}^r \leq \alpha_j \quad j \in \mathcal{K}, p \in \mathcal{P} \quad (F.6)$$

$$\sum_{r \in \mathcal{R}} \sum_{p_i^r \in s_r | p_i^r = p} to_{j,i}^r \leq \alpha_j \quad j \in \mathcal{K}, p \in \mathcal{P} \quad (F.7)$$

$$from_{j,i}^r = 1 \Rightarrow \sum_{r' \in \mathcal{R} \setminus \{r\}} \sum_{p_{i'}^{r'} \in s_{r'} | p_{i'}^{r'} = p_i^r} to_{j,i'}^{r'} = 1 \quad j \in \mathcal{K}, r \in \mathcal{R}, p_i^r \in s_r \quad (F.8)$$

$$to_{j,i}^r = 1 \Rightarrow \sum_{r' \in \mathcal{R} \setminus \{r\}} \sum_{p_{i'}^{r'} \in s_{r'} | p_{i'}^{r'} = p_i^r} from_{j,i'}^{r'} = 1 \quad j \in \mathcal{K}, r \in \mathcal{R}, p_i^r \in s_r \quad (F.9)$$

$$\sum_{p_i^r \in s_r} from_{j,i}^r = \sum_{p_i^r \in s_r} to_{j,i}^r \quad j \in \mathcal{K}, r \in \mathcal{R} \quad (F.10)$$

$$\sum_{p_i^r \in s_r} from_{j,i}^r \leq \alpha_j \cdot d_r \quad j \in \mathcal{K}, r \in \mathcal{R} \quad (F.11)$$

$$\sum_{p_i^r \in s_r} to_{j,i}^r \leq \alpha_j \cdot d_r \quad j \in \mathcal{K}, r \in \mathcal{R} \quad (F.12)$$

$$from_{j,i}^r = 1 \Rightarrow leave_{j,i}^r = 1 \quad j \in \mathcal{K}, r \in \mathcal{R}, p_i^r \in s_r \quad (F.13)$$

$$to_{j,i}^r = 1 \Rightarrow leave_{j,i}^r = 0 \quad j \in \mathcal{K}, r \in \mathcal{R}, p_i^r \in s_r \quad (F.14)$$

$$(from_{j,i+1}^r = 0 \wedge to_{j,i+1}^r = 0) \iff leave_{j,i}^r = leave_{j,i+1}^r \quad j \in \mathcal{K}, r \in \mathcal{R}, p_i^r \in s_r \quad (F.15)$$

$$(leave_{j,i}^r = 0 \wedge leave_{j,i+1}^r = 1) \Rightarrow from_{j,i+1}^r = 1 \quad j \in \mathcal{K}, r \in \mathcal{R}, p_i^r \in s_r \quad (F.16)$$

$$(leave_{j,i}^r = 1 \wedge leave_{j,i+1}^r = 0) \Rightarrow to_{j,i+1}^r = 1 \quad j \in \mathcal{K}, r \in \mathcal{R}, p_i^r \in s_r \quad (F.17)$$

$$\sum_{r \in \mathcal{R}} \sum_{p_i^r \in s_r | p_i^r = p} leave_{j,i}^r \leq \alpha_j \quad j \in \mathcal{K}, p \in \mathcal{P} \quad (F.18)$$

$$\sum_{r \in \mathcal{R}} \sum_{p_i^r \in s_r | p_i^r = pol(j)} leave_{j,i}^r = 0 \quad j \in \mathcal{K} \quad (F.19)$$

FIGURE 3 – Les contraintes relatives aux flux de marchandises

$$\text{El}t(\{\kappa(v)|v \in \mathcal{V}\}, v_r) = \kappa_r \quad r \in \mathcal{R} \quad (\text{C.1})$$

$$\text{El}t(\{\rho(v)|v \in \mathcal{V}\}, v_r) = \rho_r \quad r \in \mathcal{R} \quad (\text{C.2})$$

$$\sum_{j \in \mathcal{K}} q_j \cdot \text{leave}_{j,i}^r \leq \kappa_r \quad r \in \mathcal{R}, p_i^r \in s_r \quad (\text{C.3})$$

$$\sum_{j \in \mathcal{K}_{\text{refers}}} q_j \cdot \text{qlleave}_{j,i}^r \leq \rho_r \quad r \in \mathcal{R}, p_i^r \in s_r \quad (\text{C.4})$$

FIGURE 4 – Les contraintes de capacités

$$\max \sum_{j \in \mathcal{K}} q_j \quad (\text{O.1})$$

$$\max \sum_{j \in \mathcal{K}} q_j \cdot \text{rev}(j) \quad (\text{O.2})$$

$$\min \sum_{j \in \mathcal{K}} \sum_{r \in \mathcal{R}} \sum_{p_i^r \in s_r} \text{leave}_{j,i}^r \cdot q_j \cdot \delta(p_i^r, p_{i+1}^r) \quad (\text{O.3})$$

$$\sum_{j \in \mathcal{K}} q_j \geq Q \quad (\text{O.4})$$

FIGURE 5 – Fonctions objectifs possibles et contrainte additionnelle

pour chaque rotation r , une variable v_r spécifiant le type de navires considéré et deux variables entières κ_r et ρ_r représentant la capacité maximale en termes de conteneurs et le nombre maximale de prises disponibles.

Les contraintes (C.1) et (C.2) garantissent la cohérence des capacités maximales en fonction du type de navires tandis que les contraintes (C.3) et (C.4) assurent le respect de ces capacités maximales à la sortie de chaque port ($\mathcal{K}_{\text{refers}}$ désignant l'ensemble des parties j correspondant à des conteneurs réfrigérés).

3.4 Fonctions objectifs

La fonction (O.1) (voir la figure 5) permet de maximiser la quantité de marchandises transportée tandis que la fonction (O.2) permet de maximiser les revenus procurés par les marchandises transportées. La fonction (O.3) vise à minimiser les distances parcourues par les conteneurs (avec $\delta(p_i^r, p_{i+1}^r)$ la distance en miles entre le port p_i^r et le port p_{i+1}^r). Elle nécessite une contrainte supplémentaire (voir la contrainte (O.4)) garantissant de transporter une quantité minimale Q de marchandises. Sans cette contrainte, la solution optimale consisterait à ne rien transporter.

4 Second modèle

Dans ce second modèle (noté $\mathcal{M}_2$), nous considérons en entrée chacune des routes possibles pour chaque marchandise. Pour déterminer ces routes, nous construisons

$$\sum_{\ell \in \text{routes}(j)} \beta_j^\ell = \alpha_j \quad j \in \mathcal{K} \quad (\text{R.1})$$

$$\sum_{p_i^r \in \text{route}(j, \ell)} \beta_j^\ell = \text{leave}_{j,i}^r \quad j \in \mathcal{K}, r \in \mathcal{R}, p_i^r \in s_r \quad (\text{R.2})$$

FIGURE 6 – Contraintes relatives aux routes

d'abord le graphe de rotations. Ce graphe possède un sommet par rotation. Il existe une arête entre deux sommets si les deux rotations ont au moins un port en commun. Les arêtes sont alors étiquetées par les ports en commun. Ce graphe permet donc d'identifier rapidement les ports dans lesquels pourront s'effectuer les transbordements. Pour chaque marchandise k , nous calculons l'ensemble des routes possibles en calculant tous les chemins partant des rotations contenant le port $\text{pol}(k)$ et arrivant à une rotation contenant le port $\text{pod}(k)$ et contenant au plus $ts_{\max}(k)$ transbordements.

Étant donné une marchandise k , une route est une suite $(s_1^k, s_2^k, \dots, s_{t_k}^k)$ de suites de ports telle que :

- s_ℓ^k est une suite de ports $(p_1^{r_\ell}, p_2^{r_\ell}, \dots, p_{|s_\ell^k|}^{r_\ell})$ (avec $|s_\ell^k| \geq 2$) associée à la rotation r_ℓ telle que $p_m^{r_\ell}$ est le port précédant le port $p_{m+1}^{r_\ell}$ dans la rotation r_ℓ . Le premier port de la suite s_ℓ^k correspond au port dans lequel la marchandise entre dans la rotation r_ℓ tandis que le dernier est celui dans lequel elle sort de cette rotation.
- $p_1^{r_1} = \text{pol}(k)$.
- $p_{|s_\ell^k|}^{r_\ell} = p_1^{r_{\ell+1}}$, c'est-à-dire que ce port est utilisé pour effectuer un transbordement.
- $p_{|s_{t_k}^k|}^{r_{t_k}} = \text{pod}(k)$.

Notons que $t_k - 1$ est le nombre de transbordements de la marchandise k . Nous notons $\text{routes}(k)$ l'ensemble des routes possibles pour transporter la marchandise k et $\text{route}(k, i)$ la i -ème route de $\text{routes}(k)$. Nous étendons ces notations à chaque partie j de chaque marchandise k .

Ce second modèle reprend les variables q_j , α_j , $\text{leave}_{j,i}^r$, v_r , κ_r et ρ_r du premier modèle ainsi que les contraintes et fonctions objectifs des figures 2, 4 et 5. La différence entre les deux modèles réside simplement dans la manière de représenter les routes. Dans ce second modèle, pour chaque partie j , nous considérons une variable booléenne β_j^ℓ qui vaut 1 si la route $\text{route}(j, \ell)$ est utilisée pour transporter la partie j , 0 sinon. La contrainte (R.1) (voir la figure 6) garantit qu'une partie j n'utilise au plus qu'une seule route tandis que la contrainte (R.2) établit le lien avec les variables $\text{leave}_{j,i}^r$.

5 Méthode incomplète

Une instance réelle du problème de routage de conteneurs peut facilement impliquer plusieurs dizaines de ports et de rotations et plusieurs centaines de marchandises pour un volume total de plus de 100 000 EVP. La capacité des solveurs complets à résoudre de telles instances en temps raisonnable peut donc s'avérer rapidement limitée. Dans cette section, nous présentons une méthode incomplète qui se veut générique, dans le sens où elle ne dépend pas du critère à optimiser. Cette méthode (appelée *Router*, voir l'algorithme 1) prend, en entrées, l'ensemble $\mathcal{R}$ de rotations et l'ensemble $\mathcal{K}_0$ des marchandises à acheminer et va renvoyer la meilleure solution $\mathcal{K}_p^m$ qu'elle aura trouvée selon le critère considéré. Une solution est ici représentée par un ensemble de triplets (k, n, r) représentant l'affectation de n EVP de la marchandise k à la route r . La méthode *Router* débute par une phase d'initialisation (lignes 1-5) durant laquelle elle calculera notamment toutes les routes possibles pour chaque marchandise k de $\mathcal{K}_0$ (ligne 2) selon le même processus que le modèle $\mathcal{M}_2$. Une première solution $\mathcal{K}_p^{init}$ est utilisée pour initialiser la solution courante $\mathcal{K}_p$ (ligne 4). Elle peut, par exemple, être fournie par l'utilisateur ou construite à partir de zéro en utilisant la méthode *Placer* (voir l'algorithme 2). Par la suite, la méthode *Router* alternera entre une phase où elle déplace (ligne 8) ou supprime des marchandises (ligne 10) et une phase où elle place les marchandises sur des routes (ligne 11). Cette alternance est répétée nb_it fois (lignes 6-12). À l'issue de chaque itération, la meilleure solution $\mathcal{K}_p^m$ est mise à jour si nécessaire (ligne 12).

L'objectif de la méthode *Placer* est d'affecter autant de marchandises que possible aux routes disponibles dans la limite du nombre s_{max} de divisions possibles d'une même marchandise k . Ainsi, pour chaque marchandise k pour laquelle il reste des conteneurs à placer, cette méthode va identifier les routes possibles parmi les routes initialement disponibles pour la marchandise k . Une route est possible si les rotations qu'elle utilise sont en mesure d'accepter de la marchandise en termes de charge (pour les conteneurs premium, standards ou vides) ou en termes de charge et de prises électriques (pour les conteneurs réfrigérés). La fonction *routes* renvoie la liste des routes possibles pour une marchandise k vis-à-vis de la solution courante $\mathcal{K}_p$. Pour cela, elle s'appuie sur les types de navires disponibles pour chaque rotation. Les types disponibles évoluent en permanence suite à l'ajout, au déplacement ou retrait de marchandises de sorte à garantir qu'à chaque étape, il existe au moins un type de navires possible pour chaque rotation. La fonction *meilleure - route* renvoie la route r disponible qui permet de transporter le plus de conteneurs de la marchandise k en fonction de sa nature ainsi que ce nombre n de conteneurs. La méthode *Déplacer* (voir l'algorithme 3) repose sur un principe

Algorithme 1 : *Router*($\mathcal{R}, \mathcal{K}_0$)

```

1 pour chaque  $k \in \mathcal{K}_0$  faire
2   Calcul des routes possibles pour  $k$ 
3    $nb[k] \leftarrow q(k)$ 
4    $\mathcal{K}_p \leftarrow \mathcal{K}_p^{init}$ 
5    $\mathcal{K}_p^m \leftarrow \mathcal{K}_p$ 
6 pour  $i$  de 1 à  $nb\_it$  faire
7   si  $i$  est impair alors
8      $Déplacer(\mathcal{K}_p)$ 
9   sinon
10     $Retirer(\mathcal{K}_p, nb)$ 
11     $Placer(\mathcal{K}_p, nb)$ 
12    si  $\mathcal{K}_p$  meilleur que  $\mathcal{K}_p^m$  alors  $\mathcal{K}_p^m \leftarrow \mathcal{K}_p$ 
13 retourner  $\mathcal{K}_p^m$ 

```

similaire. La principale différence est qu'une quantité n de la marchandise k est initialement acheminée via une route r et qu'une partie ou la totalité de cette quantité va être transportée via une nouvelle route r' . L'idée sous-jacente est de libérer de la place (par exemple sur une route potentiellement saturée) pour permettre l'acheminement d'une autre marchandise. Enfin, la méthode *Retirer* (voir l'algorithme 4) permet de supprimer tout acheminement pour certaines marchandises sélectionnées aléatoirement, cette sélection aléatoire permettant ainsi de diversifier la recherche.

Différentes heuristiques peuvent être employées pour décider de l'ordre selon lequel les marchandises sont traitées dans les méthodes *Placer* et *Déplacer*. Par exemple, pour la méthode *Placer*, on peut commencer par traiter les marchandises qui ont le moins de routes possibles ou encore les trier selon la quantité restant à acheminer. Pour la méthode *Déplacer*, on peut traiter prioritairement les marchandises acheminées via des rotations dont les navires sont pleins à la sortie de certains ports. Plusieurs critères différents peuvent être envisagés (ligne 12 de l'algorithme 1) comme, par exemple, maximiser la quantité totale de conteneurs transportés ou les revenus engendrés ou encore minimiser la distance parcourue par l'ensemble des marchandises. L'approche présentée ici s'avère donc relativement générale et peut aisément être spécialisée en fonction des besoins. Elle est paramétrée par le nombre r_{max} de marchandises à supprimer à chaque appel à la méthode *Retirer* et par le nombre nb_it d'itérations réalisées dans la méthode *Router*.

6 Expérimentations

6.1 Protocole expérimental

En l'absence d'un jeu d'instances publiques pour le problème de routage de conteneurs, nous construisons des instances aussi réalistes que possible à partir

Algorithme 2 : $Placer(\mathcal{K}_p, nb)$

```
1 pour chaque  $k \in \mathcal{K}_0$  faire
2    $s \leftarrow |\{r | (k, x, r) \in \mathcal{K}_p\}|$ 
3   tant que  $nb[k] > 0$  et  $s \leq s_{max}$  et
      $routes(\mathcal{K}_p, k) \neq \emptyset$  faire
4      $(n, r) \leftarrow meilleure\_route(\mathcal{K}_p, k)$ 
5      $\mathcal{K}_p \leftarrow \mathcal{K}_p \cup \{(k, n, r)\}$ 
6      $nb[k] \leftarrow nb[k] - n$ 
7      $s \leftarrow s + 1$ 
```

Algorithme 3 : $Déplacer(\mathcal{K}_p)$

```
1 pour chaque  $(k, n, r) \in \mathcal{K}_p$  faire
2    $s \leftarrow |\{r | (k, x, r) \in \mathcal{K}_p\}|$ 
3   tant que  $n > 0$  et  $s \leq s_{max}$  et
      $routes(\mathcal{K}_p, k) \neq \emptyset$  faire
4      $(n', r') \leftarrow meilleure\_route(\mathcal{K}_p, k)$ 
5     si  $n \leq n'$  alors
6        $\mathcal{K}_p \leftarrow (\mathcal{K}_p \setminus \{(k, n, r)\}) \cup \{(k, n', r')\}$ 
7        $n \leftarrow 0$ 
8     sinon
9       si  $s < s_{max}$  alors
10         $\mathcal{K}_p \leftarrow \mathcal{K}_p \cup \{(k, n', r')\}$ 
11         $n \leftarrow n - n'$ 
```

des données de la LINER-LIB [5], qui constitue la référence pour les expérimentations concernant le problème LSNDP. Nous considérons les ports et les types de navires de la LINER-LIB. Toutefois, pour les navires, leur capacité de chargement est tirée aléatoirement entre 75 % et 100 % de leur capacité maximale, différents facteurs faisant qu'ils sont rarement exploités à leur capacité maximale. Par ailleurs, la LINER-LIB ne prenant pas en compte les conteneurs réfrigérés, nous fixons aléatoirement le nombre de prises entre 5 % et 10 % de la capacité maximale du navire afin de correspondre aux valeurs habituellement observées dans la pratique. Ensuite, concernant les marchandises, nous exploitons les couples (POL, POD) des instances de la LINER-LIB. Pour un couple donné, nous générons un nombre de conteneurs et des revenus associés en modulant aléatoirement les valeurs proposées par la

Algorithme 4 : $Retirer(\mathcal{K}_p, nb)$

```
1 pour  $i$  de 1 à  $r_{max}$  faire
2   Choisir aléatoirement  $k \in \mathcal{K}_0$ 
3   pour chaque  $(k, n, r) \in \mathcal{K}_p$  faire
4      $\mathcal{K}_p \leftarrow \mathcal{K}_p - \{(k, n, r)\}$ 
5    $nb[k] \leftarrow q(k)$ 
```

LINER-LIB. Entre 5 % et 12 % de ces conteneurs seront des conteneurs réfrigérés, le restant étant des pleins. Les cinq marchandises générant les plus forts revenus sont considérées comme des marchandises premium. Enfin, nous ajoutons aléatoirement un certain nombre de conteneurs vides de sorte que pour chaque port, la balance entre les nombres de conteneurs entrants et sortants ne soit pas déséquilibrée de plus de 25 %.

Dans nos expérimentations, nous générons aléatoirement 40 instances à partir des données des instances *Baltic*, *WAF* et *Mediterranean* et *Pacific* de la LINER-LIB (10 instances par instance de la LINER-LIB). Ces instances ont entre 12 et 45 ports, de 67 à 1 785 marchandises pour un nombre total d'EVP compris entre 14 487 et 157 915. Pour chacune de ces instances, nous considérons quatre jeux de rotations différents. Pour chaque rotation, nous exploitons tous les types de navires compatibles avec les ports visités par la rotation. Au final, nous disposons ainsi de 160 instances du problème de routage de conteneurs. Chacune de ces instances est résolue en considérant, d'une part, une méthode complète basée sur l'un des deux modèles $\mathcal{M}_1$ ou $\mathcal{M}_2$ et, d'autre part, notre méthode incomplète.

Les modèles $\mathcal{M}_1$ et $\mathcal{M}_2$ sont implémentés dans OR-Tools (version 9.6.2534¹) via son interface Python et résolus grâce au solveur CP-SAT d'OR-Tools. Ce choix est guidé par l'efficacité du solveur et la possibilité de pouvoir exploiter plusieurs processus. Hormis le nombre de threads fixé à 16, nous utilisons le paramétrage par défaut du solveur. Le temps d'exécution est limité à une heure par instance. La méthode incomplète *Router* est implémentée en Python 3. Son nombre d'itérations varie entre 50 et 200 et le nombre de retraits de marchandises est fixé à 20 % du nombre de marchandises de l'instance. Chaque instance est résolue 50 fois par la méthode *Router*. Toutes les expérimentations sont réalisées sur des serveurs dotés de processeurs Intel Xeon Gold 5218R cadencés à 2,1 GHz et dotés de 192 Go de mémoire. Enfin, les marchandises sont divisées en au plus deux parties.

6.2 Résultats

Nous comparons d'abord les modèles $\mathcal{M}_1$ et $\mathcal{M}_2$. Ces deux modèles sont relativement proches en termes de performances. Le modèle $\mathcal{M}_1$ résout optimalement 36 instances contre 38 pour le modèle $\mathcal{M}_2$ si on maximise la quantité de marchandises transportées (47 contre 46) si on maximise les revenus. Pour ces deux critères, il s'avère que les preuves d'optimalité ne sont trouvées que pour les instances basées sur *Baltic* et *WAF*. Ces instances sont les plus petites et correspondent à des services de type *feeder* (c'est-à-dire des services courts destinés à approvisionner les grandes lignes). Sur

1. https://developers.google.com/optimization/cp/cp_solver

ces instances, le modèle $\mathcal{M}_1$ permet souvent de trouver la meilleure solution plus rapidement. Il conduit aussi à un meilleur temps total de résolution dans la plupart des cas. Pour les instances basées sur **Mediterranean** et **Pacific** (c'est-à-dire des instances générales intégrant **feeder**ing et grandes lignes), au bout d'une heure, la résolution est toujours à la recherche de la solution optimale, quel que soit le modèle considéré. Par contre, pour ces instances, quand le nombre de rotations augmente, le modèle $\mathcal{M}_2$ permet d'aboutir à une solution de meilleure qualité au bout du temps imparti. La figure 7(a) présente le ratio entre la valeur de la meilleure solution trouvée par le modèle $\mathcal{M}_1$ et celle produite par le modèle $\mathcal{M}_2$ pour les deux critères (les instances étant triées selon un ratio croissant). Concernant le critère de minimisation des distances, quel que soit le modèle considéré, aucune instance n'est résolue de manière optimale. Pour certaines instances, le solveur ne parvient même pas à trouver une solution au problème dans le temps imparti. Cette variante du problème semble donc s'avérer bien plus difficile que les deux autres. L'amplitude de la fonction objectif avec des valeurs de l'ordre de plusieurs centaines de millions et la contrainte additionnelle (O.4) en sont probablement la cause.

Si nous considérons maintenant la méthode incomplète *Router*, nous pouvons d'abord noter qu'elle est relativement robuste. En effet, on peut constater, que d'une exécution à l'autre, les résultats sont sensiblement les mêmes avec un écart-type d'au plus 10 % et inférieur à 3 % en général. La figure 8 présente, par exemple, la dispersion des valeurs de la meilleure solution pour chacune des dix instances **Pacific** quand on maximise la quantité de marchandises transportées. Ensuite, la qualité de la solution trouvée est peu impactée par le nombre d'itérations comme le montrent les figures 7(b) et (c). Ces figures permettent d'estimer la qualité des solutions trouvées en représentant le ratio entre la valeur de la meilleure solution trouvée par un des deux modèles et celle de la meilleure solution de la pire exécution de la méthode incomplète selon le nombre d'itérations et le critère considérés. Ces comparaisons montrent également que la méthode incomplète permet d'obtenir de meilleurs résultats que l'approche complète pour la majeure partie des instances. Concernant les temps de calculs, ils sont au maximum de l'ordre de 20 secondes pour 50 itérations et d'une minute pour 200 itérations. À noter que les temps sont du même ordre quel que soit le critère exploité. La méthode incomplète permet donc d'avoir des résultats de qualité en temps raisonnable, même quand on minimise la somme des distances parcourues.

Notons que nous ne sommes pas en mesure de comparer nos résultats avec des méthodes issues de la littérature. D'une part, ces méthodes ne traitent pas la même variante du problème. En particulier, elles ne considèrent qu'un seul type de conteneurs (les conte-

neurs pleins) et un seul type de navire par rotation. D'autre part, leurs sources ou exécutables ne sont pas disponibles.

7 Conclusions et perspectives

Dans cet article, nous avons proposé deux modèles basés sur la PPC permettant de traiter le problème de routage de conteneurs ainsi qu'une méthode incomplète. Les résultats que nous avons obtenus, bien que préliminaires, montrent l'intérêt pratique de notre méthode incomplète. Ils mettent également en avant la différence de difficultés selon le critère d'optimisation considéré ou la nature des instances (**feeder**ing ou générale). Sur ce dernier point, une piste possible d'amélioration pour les instances générales consisterait à traiter séparément les services de type **feeder**ing et ceux de type grandes lignes.

Remerciements

Ce travail est soutenu par Bpifrance dans le cadre du projet PIA Transformation Numérique du Transport Maritime (TNTM).

Références

- [1] R. AGARWAL et Ö. ERGUN : Ship Scheduling and Network Design for Cargo Routing in Liner Shipping. *Transportation Science*, 42(2):175–196, 2008.
- [2] J. F. ÁLVAREZ : Joint routing and deployment of a fleet of container vessels. *Maritime Economics & Logistics*, 11(2):186–208, 2009.
- [3] M. BELL, X. LIU, P. ANGELOUDIS, A. FONZONE et S. HOSSEINLOO : A frequency-based maritime container assignment model. *Transportation Research Part B : Methodological*, 45:1152–1161, 2011.
- [4] B. BROUER, D. PISINGER et S. SPOORENDONK : Liner shipping cargo allocation with repositioning of empty containers. *INFOR Journal*, 49:109–124, 2011.
- [5] B. BROUER, J. F. ÁLVAREZ, C. E. M. PLUM, D. PISINGER et M. M. SIGURD : A base integer programming model and benchmark suite for liner-shipping network design. *Transportation Science*, 48(2):281–312, 2014.
- [6] M. CHRISTIANSEN, E. O. HELLSTEN, D. PISINGER, D. SACRAMENTO et C. VILHELMSSEN : Liner shipping network design. *European Journal of Operational Research*, 286, 2020.
- [7] Y. EL GHAZI, D. HABET et C. TERRIOUX : A CP Approach for the Liner Shipping Network Design Problem. *In CP*, pages 16 :1–16 :21, 2023.

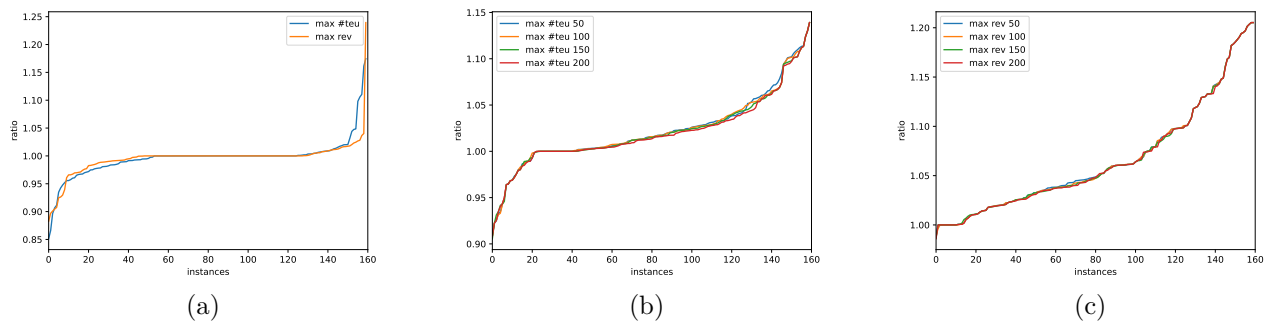


FIGURE 7 – Comparaisons de la qualité des solutions obtenues par les deux modèles (a) et par la méthode incomplète selon le nombre d’itérations réalisées et le critère considéré (b) et (c)

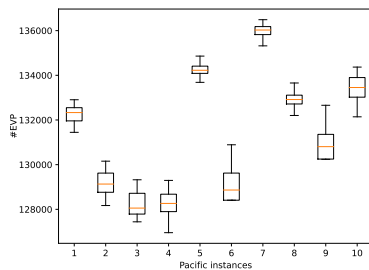


FIGURE 8 – Dispersion de la qualité de la méthode incomplète quand on maximise la quantité de marchandises transportées

- [8] S. GUERICKE et K. TIERNEY : Liner shipping cargo allocation with service levels and speed optimization. *Transportation Research Part E Logistics and Transportation Review*, 84, 12 2015.
- [9] S. IBARAKI et M. FUKUSHIMA : Primal-dual proximal point algorithm for multicommodity network flow problems. *Journal of The Operations Research Society of Japan*, 37:297–309, 1994.
- [10] C. V. KARSTEN, D. PISINGER, S. RØPKE et B. BROUER : The time constrained multi-commodity network flow problem and its application to liner shipping network design. *Transportation Research Part E : Logistics and Transportation Review*, 76:122–138, 2015.
- [11] D. KOZA : Liner shipping service scheduling and cargo allocation. *EJOR*, 275, 12 2018.
- [12] Q. MENG et S. WANG : Liner shipping service network design with empty container repositioning. *Transportation Research Part E : Logistics and Transportation Review*, 47(5):695–708, 2011.
- [13] D. MÜLLER, S. GUERICKE et K. TIERNEY : Integrating fleet deployment into the liner shipping cargo allocation problem. pages 306–320, 2017.
- [14] S. OZCAN, D. T. ELIYI et L. REINHARDT : Cargo allocation and vessel scheduling on liner shipping with synchronization of transshipments. *Applied Mathematical Modelling*, 77(1):235–252, 2020.
- [15] F. ROSSI, P. van BEEK et T. WALSH : *Handbook of Constraint Programming*. Elsevier, 2006.
- [16] D.-P. SONG et J.-X. DONG : Cargo routing and empty container repositioning in multiple shipping service routes. *Transportation Research Part B : Methodological*, 46:1556–1575, 12 2012.
- [17] S. WANG et Q. MENG : Schedule design and container routing in liner shipping. *Transportation Research Record : Journal of the Transportation Research Board*, 2222:25–33, 2011.
- [18] S. WANG et Q. MENG : Liner ship fleet deployment with container transshipment operations. *Transportation Research Part E : Logistics and Transportation Review*, 48(2):470–484, 2012.
- [19] S. WANG et Q. MENG : Sailing speed optimization for container ships in a liner shipping network. *Transportation Research Part E : Logistics and Transportation Review*, 48:701–714, 05 2012.
- [20] S. WANG et Q. MENG : Container liner fleet deployment : A systematic overview. *Transportation Research Part C : Emerging Technologies*, 77:389–404, 2017.
- [21] S. WANG, Q. MENG et Z. LIU : Containership scheduling with transit-time-sensitive container shipment demand. *Transportation Research Part B : Methodological*, 54:68 – 83, 2013.
- [22] S. WANG, Q. MENG et Z. SUN : Container routing in liner shipping. *Transportation Research Part E : Logistics and Transportation Review*, 49:1–7, 2013.