

Gestion d'une flotte de véhicules automatisés dans un environnement portuaire

Danloup Nicolas¹, Trouillet Benoît¹, Toguyéni Armand¹, and Bourdeaud'huy Thomas¹

Université de Lille 1
Cité Scientifique 59655 Villeneuve d'Ascq Cedex, France
ndanloup@hotmail.fr
benoit.trouillet@ec-lille.fr
armand.toguyeni@ec-lille.fr
thomas.bourdeaud_huy@centralelille.fr

Résumé

Le transport de marchandises intègre un grand nombre de problème d'optimisation tout au long de la chaîne logistique. Cet article est dédié à la problématique de chargement/déchargement de containers dans un environnement portuaire. L'automatisation des processus est devenue une priorité pour les plus grands acteurs du secteur portuaire contraints de s'adapter à la croissance du volume de containers transportés liée à la globalisation de l'économie mondiale et aux capacités des nouvelles générations de bateaux. La concurrence entre terminaux et la nécessité de réduire les délais de chargement et déchargement des bateaux nécessitent de la part des opérateurs portuaires une augmentation de la productivité globale, une réduction des coûts d'exploitation tout en garantissant une sécurité maximale.

Cet article propose la description d'algorithmes qui permettent de trouver des solutions faisable pour le problème de routage d'une flotte d'AGV, dans un temps raisonnable. La résolution exacte du problème montre très vite ses limites lors du passage à l'échelle. L'approche proposée ici est de coupler une résolution exacte avec des heuristiques. L'idée est de proposer des éléments de solution dans le but de faire chuter la complexité intrinsèque de ce type de problème. Ainsi, une solution peut être proposée avec un temps de résolution raisonnable vis à vis des contraintes imposées par les opérateurs portuaires.

1 Introduction

Durant les dernières décennies, le commerce à distance s'est très fortement développé. La globalisation des échanges à l'échelle mondiale nécessite de transporter des quantités de marchandises en perpétuelle augmentation. Les voies navigables offrent une opportunité de transporter de gros volumes sur de longues distances tout en permettant de réduire les coûts de transport. Le transport maritime constitue un moyen de transport à moindre frais ; il est trente fois moins cher que le transport terrestre. L'augmentation du volume à transporter a vu l'augmentation de la taille des navires et, en conséquence, des capacités d'accueil des lieux portuaires pour l'amarrage des navires mais aussi en termes de chargement et/ou déchargement.

L'idée actuelle est de proposer des solutions automatisées de chargement/déchargement des cargos. Ce qui est motivé par le développement de véhicules automatisés sur le marché : les AGV (pour Automated Guided Vehicles) et des conteneurs de taille standard (équivalents 20 ou 40 pieds). Le but est de minimiser le temps opérationnel et de garantir une efficacité de fonctionnement. Ainsi, l'automatisation vise à permettre l'accélération du transfert des conteneurs du navire vers les clients et inversement.

L'article s'articule autour de cinq parties. Nous commencerons par décrire le problème étudié. Dans la partie suivante, le problème est formalisé vers une abstraction mathématique. La quatrième partie est dédiée à différents algorithmes qui permettent d'accélérer la résolution du problème. Puis, la cinquième partie propose une analyse du problème du plus court chemin dans le contexte portuaire et des difficultés pour l'obtenir en tenant compte des contraintes de sécurité. Finalement, des résultats de simulation viennent illustrer la fin de cet article. La partie finale est dédiée à la conclusion de cet étude.

2 Description du problème

Les problèmes de gestion des opérations dans les terminaux à conteneurs est une thématique très importante dans la littérature scientifique en recherche opérationnelle et logistique. Il existe des synthèses de résultats dans les articles suivants : [?, ?, ?, ?]. Il existe de nombreux problèmes au sein des activités portuaires. On trouve, par exemple, des études d'allocation de quai aux bateaux, d'ordonnancement des grues de quai, celui de l'allocation de stockage aux conteneurs. . .

La zone d'étude peut être représentée schématiquement par la figure 1. On y trouve toutes les caractéristiques d'une zone de déchargement de bateaux. deux grues de déchargement sont représentées. Elles intègrent des emplacements de stationnement (ici cinq zones sont proposées). Des véhicules viennent recevoir ou amener des conteneurs, pour ensuite rejoindre une grille de circulation. Les véhicules chargés transportent leur cargaison vers la zone de stockage.

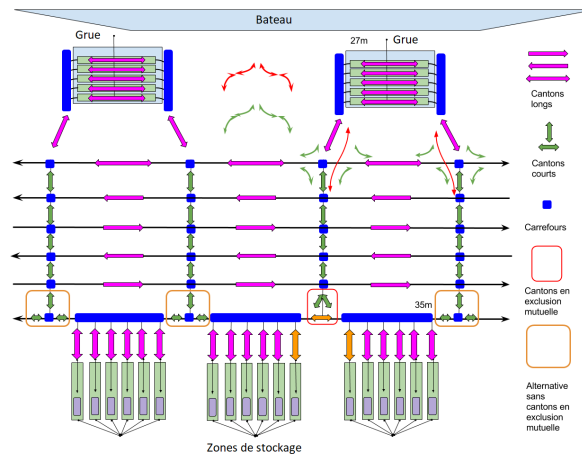


FIGURE 1 – Topologie du deuxième réseau

Notre étude porte sur la problématique de routage des AGV au travers de ce type de réseau. L'environnement est constitué de bateaux à décharger, les grues de quai sont en place. L'opérateur d'exploitation du port (*Terminal Operating System*) donne des ordres de déchargement au fil de l'eau. L'exécution est confiée à un prestataire qui est chargé d'envoyer un véhicule pour transporter les conteneurs de la grue de quai vers un emplacement de stockage et inversement. Nous devons donc gérer une flotte d'AGV pour honorer les ordres reçus. Plus précisément, il faut affecter un AGV à chaque mission et lui trouver une route qui permette l'exécution de la mission dans le temps imparti.

3 Formalisation du problème

Dans ce paragraphe, les éléments de formalisation sont rapidement présentés. Une description plus détaillée se trouve dans [?, ?].

3.1 Notations

Les notations utilisées, par la suite, sont :

- $\mathcal{A} = \{a_1, a_2, \dots, a_A\}$ l'ensemble des AGV avec $A = |\mathcal{A}|$.
- $\mathcal{B} = \{b_1, b_2, \dots, b_B\}$ l'ensemble des cantons avec $B = |\mathcal{B}|$.
- ${}^A\mathcal{B}_i$ l'ensemble des cantons accessibles depuis b_i .
- $\mathcal{C} = \{c_1, c_2, \dots, c_C\}$ l'ensemble des carrefours avec $C = |\mathcal{C}|$.
- ${}^i c^j$ l'unique carrefour situé entre les cantons b_i et b_j .

- $s_m \in \mathcal{B}$ le canton de départ de l'AGV a_m .
- $t_m \in \mathcal{B}$ le canton d'arrivée de l'AGV a_m .
- d_{ij}^b la durée de parcours du canton b_j lorsque l'AGV vient du canton b_i .
- t_{ij}^b la durée de sortie du canton b_j lorsque l'AGV vient du canton b_i .
- d_{ij}^c la durée de parcours du carrefour $i^c j$ en allant du canton b_i au canton b_j .
- t_{ij}^c la durée de sortie du carrefour $i^c j$ en allant du canton b_i au canton b_j .

En plus de ces notations, les nouvelles notations suivantes sont introduites :

- $\mathcal{E}_i^b = \{e_1, e_2, \dots, e_E\}$ l'ensemble des intervalles horaires autorisés pour le canton b_i avec $E = |\mathcal{E}|$.
- $\mathcal{E}_p^c = \{e_1, e_2, \dots, e_E\}$ l'ensemble des intervalles horaires autorisés pour le carrefour c_p avec $E = |\mathcal{E}|$.
- $e_k = (e_k^+, e_k^-)$ avec e_k^+ de début de l'intervalle e_k et e_k^- la fin de l'intervalle e_k .
- $r_m = (s_m, c_p, b_i, \dots, t_m)$ est le chemin emprunté par l'AGV a_m .
- $R = \{r_1, r_2, \dots, r_A\}$ est l'ensemble de toutes les routes et représente une solution entière faisable.

3.2 Moteur d'affectation

Cette couche se base sur un algorithme d'affectation de type glouton. C'est-à-dire que l'on procède par itérations en faisant des choix localement optimaux, ce qui ne garantit toutefois pas l'optimalité globale de la solution.

Elle fait l'hypothèse que le nombre d'AGVs disponibles en circulation est supérieur ou égal au nombre de missions à réaliser. Si ce n'est pas le cas, la couche agrégation devra limiter le nombre de missions à traiter par la couche macro.

Son principe est le suivant.

Soit M l'ensemble des missions à affecter.

1. Pour chaque mission $m \in M$, identifier l'AGV a étant le plus proche de la localisation (départ ou arrivée) de cette mission et sauvegarder le triplet (m, a, d) dans une liste L , où d est la distance de l'AGV à la localisation de la mission.
2. Sélectionner dans L le triplet (m, a, d) correspondant à la plus petite distance d parmi toutes les paires sauvegardées à l'étape précédente.
3. Eliminer la mission m qui vient d'être choisie de la liste M .
4. Recommencer les étapes précédentes à partir de l'étape 1 jusqu'à ce que l'ensemble M soit vide.

3.3 Moteur de routage

Cette section décrit le modèle mathématique proposé pour traiter la problématique de routage des AGV vers les emplacements correspondants aux missions auxquelles ils ont été affectés.

3.3.1 Topologie

Les AGV peuvent se déplacer en suivant un réseau de transport virtuel, inspiré de l'organisation ferroviaire, défini par des cantons et des carrefours, qui permettent de représenter un *graphe* dans lequel les AGV évolueront.

Cantons (*blocks* b_i, b_j). D'après nos hypothèses, chaque canton $b_i \in \mathcal{B}$ est associé à une durée de parcours *nominale* moyenne dénotée par $d_i^b \in \mathbb{N}^+$, et une durée moyenne de sortie dénotée par $t_i^b \in \mathbb{N}^+$ exprimée en secondes. De plus, chaque paire de cantons $(b_i, b_j) \in \mathcal{B} \times {}^A\mathcal{B}_i$ est associée à la durée de parcours *nominale* de j lorsque l'AGV vient de i dénotée par $d_{ij}^b \in \mathbb{N}^+$, et une durée de sortie de j dénotée par $t_{ij}^b \in \mathbb{N}^+$ exprimée en secondes.

L'ensemble \mathcal{B} est partitionné en cantons *courts* \mathcal{B}^c et *longs* \mathcal{B}^l tels que $\mathcal{B} = \mathcal{B}^c \cup \mathcal{B}^l$. On dénote \mathcal{B}^c et \mathcal{B}^l leurs cardinaux respectifs.

Carrefours (*crossroads* c_p). D'après nos hypothèses, pour chaque paire de cantons $(b_i, b_j) \in \mathcal{B} \times {}^A\mathcal{B}_i$, une durée de parcours dénotée $d_{ij}^c \in \mathbb{N}^+$, et une durée de sortie dénotée $t_{ij}^c \in \mathbb{N}^+$, exprimées en secondes, sont associées à la traversée du carrefour lorsque l'AGV va du canton b_i au canton b_j .

Relations entre cantons et carrefours. La topologie considérée correspond à un enchaînement de cantons et carrefours, sous la forme d'un graphe *biparti*¹.

Pour tout canton $b_i \in \mathcal{B}$, on dénote par \mathcal{C}_i l'ensemble des carrefours accessibles² depuis b_i .

Pour tout canton $b_i \in \mathcal{B}$, on dénote par ${}^A\mathcal{B}_i$ l'ensemble des cantons accessibles depuis b_i .

La topologie est telle qu'il ne peut y avoir qu'un unique carrefour entre deux cantons accessibles entre-eux. Autrement dit, on ne peut passer d'un canton à un autre que par au plus un seul carrefour.

Formellement, pour tout canton $b_i \in \mathcal{B}$, et tout canton $b_j \in {}^A\mathcal{B}_i$, il existe un unique carrefour c_p tel que $b_j \in {}^A\mathcal{B}_i^p$. On dénote cet unique carrefour entre les cantons b_i et b_j par l'expression ${}^i c^j$.

Missions (*AGV* a_m, a_n). Nous faisons l'hypothèse que le moteur d'affectation a affecté à chaque mission un AGV. Nous *unifions* donc dans notre modélisation les vocabulaires de mission et d'AGV.

Les AGV qui ne sont affectés à aucune mission ne font pas partie de cet ensemble. Leurs positions (dans des cantons longs, puisqu'ils sont stationnés) seront toutefois transmises par le moteur d'affectation au moteur de routage, afin que ce dernier puisse interdire l'utilisation des cantons correspondants. On dénote par \mathcal{B}^f (*forbidden*)³ l'ensemble des cantons qui ne peuvent être utilisés dans la résolution courante.

Chaque mission est donc associée à un lieu de départ correspondant à la position courante de l'AGV au moment où cette mission est transmise. Chaque mission est également associée à un lieu d'arrivée, en fonction de son type (grues de quai ou grues de stock pour les missions de type 1 et 2, cantons pour les missions de type 3). Ces lieux doivent nécessairement être des cantons longs.

Pour chaque AGV $a_m \in \mathcal{A}$, on dénotera par $s_m \in \mathcal{B}^l$ son canton de départ (*source*) et par $t_m \in \mathcal{B}^l$ son canton d'arrivée.

3.4 Résolution exacte du problème et premiers résultats

De manière à pouvoir proposer une formulation linéaire du problème considéré, on introduit un certain nombre de constantes et variables en plus des termes introduits dans la formalisation précédente. Pour plus de détails, l'article [?] propose une formalisation détaillée du problème linéaire.

De nombreuses contraintes de sécurité sont à prendre en compte. On peut citer parmi les contraintes de routage :

- Respect de l'origine et de la destination de la mission
- Respect de la topologie
- La connexité du parcours obtenu
- Interdiction de demi-tour (et les éliminations des sous-tours)

Ce qui justifie des temps de réponse au dessus des délais imposés par une utilisation dans des conditions réelles. Différents jeux d'essai ont été créés pour tester la modélisation et vérifier que les contraintes du système sont respectées. Pour cela, deux réseaux simples qui illustrent des problématiques particulières à résoudre ont été créés. Le premier jeu d'essai n'utilise qu'une grue de quai et une grue de stockage dans la problème représenté par la figure 1. Ce réseau est utilisé par trois jeux d'essai :

1. Il ne peut y avoir deux cantons successifs qui ne soient pas reliés par un carrefour et inversement.
2. Si l'on considère un canton b_i qui ne puisse être parcouru que dans un seul sens, le carrefour en amont de ce canton ne sera pas considéré comme *accessible*.
3. Non utilisé dans le modèle pour le moment

- jeu1 : une mission est gérée par un AGV. La mission est un transfert de la grue de quai (haut de la figure) vers la grue de stock (bas de la figure). Il permet de vérifier la capacité du modèle mathématique à fournir des solutions à la problématique du routage.
- jeu2a : deux missions sont gérées par deux AGVs. La première mission est un transfert de la grue de quai vers la grue de stock. La deuxième mission est un transfert de la grue de stock vers la grue de quai. Il permet de vérifier la gestion de carrefour et des circulations verticales.
- jeu2b : deux missions sont gérées par deux AGVs. Les deux missions sont un transfert de la grue de quai vers la grue de stock.

Le modèle a été résolu à l'aide du solveur CPLEX sur un PC disposant d'un processeur cadencé à 2,7GHz et de 8Go de RAM. Un temps limite de 1h a été imposé au solveur. Les résultats sont résumés dans le tableau 1. La colonne "Coût" représente le coût de la fonction objectif, la colonne "Temps" représente le temps d'exécution du solveur en secondes et la colonne "gap" représente la différence entre la solution trouvée par le solveur et la borne inférieure. Un gap égal à 0 indique que la solution optimale a été trouvée.

Ces résultats permettent de voir que les contraintes physiques du système sont bien respectées, notamment au niveau des contraintes de sécurité. En effet, il est important que plusieurs AGVs ne peuvent pas se trouver dans le même canton ou carrefour au même moment. Ils permettent également de mettre en évidence la limite de résolution optimale rapidement atteinte par le solveur. A partir de quatre AGVs, le solveur n'est plus en mesure de trouver la solution optimale. Il faut cependant noter qu'une solution (non optimale) est tout de même trouvée par celui-ci en moins d'une seconde pour les jeux d'essai.

Jeu	Coût	Temps (s)	Gap (%)
jeu1	122	< 1	0
jeu2a	254	< 1	0
jeu2b	285	< 1	0
jeu2	328	3600	30

TABLE 1 – Résultats des différents jeux d'essai

4 Algorithmes de résolution

Trois algorithmes ont été développés pour accélérer la résolution du problème. Dans ces algorithmes, les AGV sont insérés dans la solution un par un. Pour insérer un AGV, le plus court chemin entre son origine et sa destination est calculé en considérant les temps de parcours. Comme plusieurs AGV ne peuvent se trouver dans le même canton ou le même carrefour au même instant, le plus court chemin est calculé en prenant en compte le trajet emprunté par les AGV déjà insérés dans la solution et leurs horaires. Ceci permet d'éviter les blocages qui pourraient survenir. Les algorithmes contiennent donc deux parties. Dans la première, il faut définir un ordre d'insertion pour les AGV. Dans la deuxième, il faut calculer le plus court chemin pour chaque AGV. Pour cette dernière, un modèle mathématique décrit dans la section suivante est utilisé.

4.1 Algorithme 0

L'algorithme 0 est l'algorithme le plus basique. Les AGV ne sont pas triés, ils sont insérés dans l'ordre dans lequel ils sont présent dans \mathcal{A} .

4.2 Algorithme 1

Dans l'algorithme 1, le plus court chemin est calculé pour chaque AGV sans prendre en compte la présence éventuelle d'autres véhicules sur le circuit. Les AGV sont ensuite triés du plus proche au plus éloigné de leur destination. Ils sont ensuite insérés dans la solution en prenant en compte

Algorithm 0

```

1: function HEUR0( $\mathcal{A}$ )
2:   for all  $a_m \in \mathcal{A}$  do
3:      $r_m =$  shortest path between  $s_m$  and  $t_m$ 
4:     add  $r_m$  in  $R$ 
5:   end for
6:   return  $R$ 
7: end function

```

cet ordre. Ainsi, l'AGV le plus proche de sa destination sera inséré en premier, et le plus éloigné sera inséré en dernier.

Algorithm 1

```

1: function HEUR1( $\mathcal{A}$ )
2:    $\mathcal{A}' = \mathcal{A}$  sorted by the AGV origin-destination distance
3:   for all  $a_m \in \mathcal{A}'$  do
4:      $r_m =$  shortest path between  $s_m$  and  $t_m$ 
5:     add  $r_m$  in  $R$ 
6:   end for
7:   return  $R$ 
8: end function

```

4.3 Algorithme glouton

Le troisième algorithme développé est un algorithme glouton. Dans celui-ci, tous les ordres d'insertion possibles pour les AGV sont testés, et la meilleure solution est conservée. Pour cela, nous utilisons un algorithme récursif⁴ (Alg. 2). Le premier appel de cet algorithme est fait avec comme paramètre d'entrées \mathcal{A} et deux ensemble vides (Alg. 3).

Algorithm 2 Algorithme glouton

```

1: function GREEDY( $\mathcal{A}', R, bestR$ )
2:   if  $\mathcal{A}' = \emptyset$  then
3:     if  $R$  better than  $bestR$  then
4:        $bestR = R$ 
5:     end if
6:     return  $bestR$ 
7:   end if
8:   for all  $a_m \in \mathcal{A}'$  do
9:      $r_m =$  shortest path between  $s_m$  and  $t_m$ 
10:    add  $r_m$  in  $R$ 
11:     $\mathcal{A}' = \mathcal{A}' \setminus \{a_m\}$ 
12:    GREEDY( $\mathcal{A}', R, bestR$ )
13:    remove  $r_m$  from  $R$ 
14:     $\mathcal{A}' = \mathcal{A}' + \{a_m\}$ 
15:   end for
16: end function

```

4. Un algorithme récursif est un algorithme qui fait appel à lui-même.

Algorithm 3 Algorithme glouton général

```

1: function MAINGREEDY( $\mathcal{A}$ )
2:   return GREEDY( $\mathcal{A}, \emptyset, \emptyset$ )
3: end function

```

5 Calcul du plus court chemin

Les différents algorithmes créés fonctionnent tous sur le même principe. Chaque AGV est inséré dans la solution en calculant le plus court chemin de son origine à sa destination. Il existe plusieurs méthodes pour calculer le plus court chemin comme la modélisation linéaire en nombre entier ou l'utilisation d'algorithmes spécifiques tel que l'algorithme de Dijkstra. Ce dernier est très efficace même sur des réseaux complexes.

Cependant, ces méthodes ne prennent pas en compte les contraintes de temps qu'il peut y avoir au niveau des nœuds. Or, à cause des contraintes de sécurité, lorsque plusieurs AGV sont déjà présents dans la solution, plusieurs intervalles de temps peuvent être interdits dans un nœud à notre AGV lors du calcul du plus court chemin. Par exemple dans la figure 2, le nœud ne sera disponible pour un nouvel AGV que dans les intervalles $[0; 10]$, $[17; 27]$ et $[36; +\infty[$.

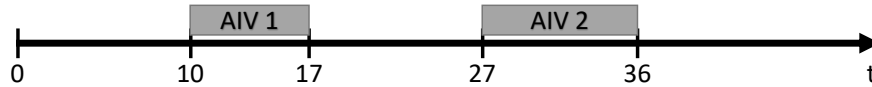


FIGURE 2 – Illustration de l'utilisation d'un nœud par des AGV en fonction du temps

Ceci complique grandement le calcul du plus court chemin puisque le nombre de solution disponible explose avec le nombre d'AGV déjà présent. En effet les nœuds occupés peuvent être par exemple contournés, ou l'AGV peut attendre que le nœud soit de nouveau libre. Malheureusement il n'existe pas à notre connaissance d'algorithmes qui permettent de résoudre ce problème. Nous avons donc créé un modèle mathématique qui permet de répondre à cette problématique. Ce modèle est décrit dans cette section.

5.1 Variables de décision

De manière à pouvoir proposer une formulation linéaire du problème considéré, on introduit un certain nombre de constantes et variables.

- $X_{ij} = 1$ si l'AGV passe du canton b_i au canton b_j dans cet ordre, 0 sinon.
- $Y_{ik}^b = 1$ si l'AGV passe dans le canton b_i dans l'intervalle autorisé e_k , 0 sinon.
- $Y_{pk}^c = 1$ si l'AGV passe dans le carrefour c_p dans l'intervalle autorisé e_k , 0 sinon.
- $\alpha_i \in \mathbb{R}$ dénote la date d'arrivée de l'avant de l'AGV dans le canton b_i .
- $\beta_i \in \mathbb{R}$ dénote la date de sortie de l'avant de l'AGV du canton b_i .
- $\gamma_i \in \mathbb{R}$ dénote la date de sortie de l'arrière de l'AGV du canton b_i .
- $\phi_p \in \mathbb{R}$ dénote la date d'arrivée de l'avant de l'AGV dans le carrefour c_p .
- $\psi_p \in \mathbb{R}$ dénote la date de sortie de l'avant de l'AGV du carrefour c_p .
- $\delta_p \in \mathbb{R}$ dénote la date de sortie de l'arrière de l'AGV du carrefour c_p .
- $\varepsilon_i \in \mathbb{R}$ dénote la durée de stationnement de l'AGV dans le canton long $b_i \in \mathcal{B}^l$.

5.2 Contraintes de routage

Pré-calculs Les variables de décision X_{ij} doivent être nulles si les cantons i et j ne sont pas accessibles entre eux. Les variables X_{ii} sont trivialement nulles également.

$$\forall b_i \in \mathcal{B}, \forall b_j \notin {}^A\mathcal{B}_i, \quad X_{ij} = 0 \quad (1)$$

$$\forall b_i \in \mathcal{B}, \quad X_{ii} = 0 \quad (2)$$

Origine et Destination. Chaque AGV doit sortir de son canton de départ. On contraint donc que parmi les variables X_m^{ij} dénotant la sortie de l'AGV a_m du canton s_m , une et une seule soit non nulle et que l'AGV ne puisse pas revenir dans son canton d'origine.

$$\sum_{b_j \in \mathcal{B} \setminus \{s\}} (X_{sj} - X_{js}) = 1 \quad (3)$$

De manière réciproque, chaque AGV doit entrer dans son canton de destination, et ne plus en ressortir.

$$\sum_{b_j \in \mathcal{B} \setminus \{t\}} (X_{tj} - X_{jt}) = -1 \quad (4)$$

Respect de la topologie. Depuis un canton b_i , un AGV ne peut se déplacer que vers un seul autre canton b_j , à condition que ce dernier soit accessible depuis b_i .

$$\forall i \in \mathcal{B}, \quad \sum_{b_j \in \mathcal{B} \setminus \{i\}} X_{ij} \leq 1 \quad (5)$$

L'opérateur d'inégalité permet d'autoriser la somme à être nulle. En effet, si l'AGV n'entre pas dans le canton b_i considéré, toutes les variables X_{ij} devront être nulles, ce que permet l'inégalité.

Au contraire, si la somme est égale à 1, c'est que l'AGV quitte le canton b_i pour se rendre vers un seul autre canton b_j .

Connexité des parcours des AGV. Si un AGV entre dans un canton b_j (sauf le canton de destination t), il doit en sortir. Inversement, si cet AGV sort du canton b_i (sauf le canton de départ s), il doit y être entré.

$$\forall i \in \mathcal{B} \setminus \{s, t\}, \quad \sum_{b_j \in \mathcal{B} \setminus \{i\}} (X_{ij} - X_{ji}) = 0 \quad (6)$$

La première somme permet de caractériser si l'AGV entre dans le canton b_j . En effet, lorsque cette somme vaut 1, c'est qu'il existe un canton b_i pour lequel $X_{ij} = 1$: l'AGV passe de b_i à b_j . La seconde somme indique que l'AGV sort de b_j . La contrainte d'égalité permet d'imposer le comportement souhaité.

Au contraire, si l'une des sommes vaut 0, c'est que l'AGV ne passe pas par le canton b_j : il ne doit ni y entrer, ni en sortir.

5.2.1 Intervalles horaires d'occupation des cantons et carrefours

Nous introduisons dans cette section des contraintes permettant de définir les intervalles horaires de présence de l'AGV dans chacun de ces emplacements. Une fois ces intervalles de temps connu, il sera possible d'imposer les contraintes de sécurité permettant d'interdire à deux AGV de se trouver dans le même canton ou carrefour au même moment.

Intervalle horaire d'occupation d'un canton. On rappelle que l'intervalle horaire d'occupation d'un canton b_i par l'AGV est dénoté par $[\alpha_i, \gamma_i]$, où les bornes temporelles sont exprimées en secondes, à partir du temps zéro correspondant à la date de réception des missions par la couche « macro ». Celles-ci sont contraintes par la durée de traversée nominale d_{ij}^b et la durée de sortie du canton considéré en fonction du canton traversé précédemment.

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i \setminus \{s\}, \quad (X_{ij} = 1) \Rightarrow \beta_j = \alpha_j + \varepsilon_j + d_{ij}^b \quad (7)$$

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i \setminus \{s\}, \quad (X_{ij} = 1) \Rightarrow \gamma_j = \beta_j + t_{ij}^b \quad (8)$$

Pour linéariser ces relations, nous utilisons un double encadrement.

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i \setminus \{s\}, \quad \beta_j - \alpha_j - \varepsilon_j - d_{ij}^b \leq R(1 - X_{ij}) \quad (9)$$

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i \setminus \{s\}, \quad \alpha_j + \varepsilon_j - \beta_j + d_{ij}^b \leq R(1 - X_{ij}) \quad (10)$$

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i \setminus \{s\}, \quad \gamma_j - \beta_j - t_{ij}^b \leq R(1 - X_{ij}) \quad (11)$$

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i \setminus \{s\}, \quad \beta_j - \gamma_j + t_{ij}^b \leq R(1 - X_{ij}) \quad (12)$$

Succession de cantons. Lorsqu'un AGV passe d'un canton b_i à un canton b_j , les intervalles horaires d'occupation des cantons b_i et b_j doivent se succéder, en prenant en compte la durée d_{ij}^c de traversée du carrefour ${}^i c^j$. Nous cherchons à imposer la relation :

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i, \quad (X_{ij} = 1) \Rightarrow \alpha_j = \beta_i + d_{ij}^c \quad (13)$$

Pour linéariser cette relation, nous utilisons un *double encadrement*.

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i, \quad \alpha_m^j - \beta_m^i - d_{ij}^c \leq R(1 - X_{ij}) \quad (14)$$

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i, \quad \beta_m^i + d_{ij}^c - \alpha_m^j \leq R(1 - X_{ij}) \quad (15)$$

Lorsque $X_{ij} = 1$, le double encadrement par 0 permet d'assurer la contrainte de séquençement recherchée. Au contraire, lorsque $X_{ij} = 0$, les inégalités des équations (14) et (15) sont toujours vérifiées.

Intervalle horaire d'occupation d'un carrefour. On rappelle que l'intervalle horaire d'occupation d'un carrefour c_p par l'AGV est dénoté par $[\phi_p, \delta_p]$, où les bornes temporelles sont exprimées en secondes, à partir du temps zéro correspondant à la date de réception des missions par la couche « macro ». Celles-ci sont contraintes par la durée de traversée nominale d_{ij}^c et par la durée de sortie t_{ij}^c du carrefour ${}^i c^j$ considéré en fonction du canton traversé précédemment et du canton de destination.

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i, \quad (X_{ij} = 1) \Rightarrow \psi_{i c^j} = \phi_{i c^j} + d_{ij}^c \quad (16)$$

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i, \quad (X_{ij} = 1) \Rightarrow \delta_{i c^j} = \psi_{i c^j} + t_{ij}^c \quad (17)$$

Pour linéariser ces relations, nous utilisons un double encadrement.

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i, \quad \psi_{i c^j} - \phi_{i c^j} - d_{ij}^c \leq R(1 - X_{ij}) \quad (18)$$

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i, \quad \phi_{i c^j} - \psi_{i c^j} + d_{ij}^c \leq R(1 - X_{ij}) \quad (19)$$

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i, \quad \delta_{i c^j} - \psi_{i c^j} - t_{ij}^c \leq R(1 - X_{ij}) \quad (20)$$

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i, \quad \psi_{i c^j} - \delta_{i c^j} + t_{ij}^c \leq R(1 - X_{ij}) \quad (21)$$

Traversée des carrefours Lors du passage d'un canton b_i à un canton b_j , l'AGV traverse le carrefour $i^c j$. L'intervalle horaire correspondant à la traversée de ce carrefour dépend de sa date de sortie du canton b_i . Nous souhaitons imposer la relation suivante :

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i, \quad (X_{ij} = 1) \Rightarrow \phi_{i^c j} = \beta_i \quad (22)$$

Pour linéariser cette relation, nous utilisons un double encadrement.

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i, \quad \phi_{i^c j} - \beta_i \leq R(1 - X_{ij}) \quad (23)$$

$$\forall b_i \in \mathcal{B}, \forall b_j \in {}^A\mathcal{B}_i, \quad \beta_i - \phi_{i^c j} \leq R(1 - X_{ij}) \quad (24)$$

5.3 Contraintes de respect des intervalles autorisés

Les contraintes de respect des intervalles autorisés rendent le calcul du plus court chemin compliqué. Elles sont utiles pour respecter les contraintes de sécurité du problème général lorsqu'on calcul le plus court chemin d'un AGV avec d'autres AGV déjà présents dans la solution. Plus le nombre d'AGV déjà présents dans la solution est grand, plus le nombre d'intervalles autorisés par canton ou carrefour sera important et plus le chemin sera difficile à calculer. Pour exprimer ces contraintes, nous utilisons des variables de sélection qui permettent de définir dans quel intervalle horaire l'AGV va occuper chaque canton et carrefour traversé, et également les variables de temps définies précédemment.

Respect des intervalles des cantons Si l'AGV passe dans un canton à un intervalle donné, alors celui-ci doit rentrer dans le canton après le début de l'intervalle, et sortir entièrement du canton avant la fin de l'intervalle.

$$\forall b_i \in \mathcal{B}, \forall e_k \in \mathcal{E}_i^b, \quad (Y_{ik}^b = 1) \Rightarrow \alpha_i \geq e_k^+ \quad (25)$$

$$\forall b_i \in \mathcal{B}, \forall e_k \in \mathcal{E}_i^b, \quad (Y_{ik}^b = 1) \Rightarrow \gamma_i \leq e_k^- \quad (26)$$

Pour linéariser ces équations, nous utilisons une constante suffisamment grande R .

$$\forall b_i \in \mathcal{B}, \forall e_k \in \mathcal{E}_i^b, \quad \alpha_i - e_k^+ \geq -R(Y_{ik}^b - 1) \quad (27)$$

$$\forall b_i \in \mathcal{B}, \forall e_k \in \mathcal{E}_i^b, \quad \gamma_i - e_k^- \leq R(1 - Y_{ik}^b) \quad (28)$$

Respect des intervalles des carrefours Si l'AGV passe dans un carrefour à un intervalle donné, alors celui-ci doit rentrer dans le carrefour après le début de l'intervalle, et sortir entièrement du carrefour avant la fin de l'intervalle.

$$\forall c_p \in \mathcal{C}, \forall e_k \in \mathcal{E}_p^c, \quad (Y_{pk}^c = 1) \Rightarrow \phi_p \geq e_k^+ \quad (29)$$

$$\forall c_p \in \mathcal{C}, \forall e_k \in \mathcal{E}_p^c, \quad (Y_{pk}^c = 1) \Rightarrow \delta_p \leq e_k^- \quad (30)$$

Pour linéariser ces équations, nous utilisons une constante suffisamment grande R .

$$\forall c_p \in \mathcal{C}, \forall e_k \in \mathcal{E}_p^c, \quad \phi_p - e_k^+ \geq -R(Y_{pk}^c - 1) \quad (31)$$

$$\forall c_p \in \mathcal{C}, \forall e_k \in \mathcal{E}_p^c, \quad \delta_p - e_k^- \leq R(1 - Y_{pk}^c) \quad (32)$$

Nombre d'intervalles utilisés par nœud Comme il existe au moins un intervalle par canton et par carrefour, l'AGV doit passer en utilisant exactement un intervalle.

$$\forall b_i \in \mathcal{B}, \quad \sum_{e_k \in \mathcal{E}^b} Y_{ik}^b = 1 \quad (33)$$

$$\forall c_p \in \mathcal{C}, \quad \sum_{e_k \in \mathcal{E}^c} Y_{pk}^c = 1 \quad (34)$$

Les cantons et carrefours dans lesquels l'AGV ne passent pas doivent également utiliser un intervalle exactement. Cet intervalle peut être choisi arbitrairement, ainsi que les temps de passage associés. Cela n'affectera en rien la qualité de la solution puisque seuls les nœuds traversés sont comptabilisés dans le calcul du plus court chemin.

5.4 Fonction objectif

La fonction objectif est la fonction qui va permettre de calculer la valeur du plus court chemin. Cette fonction dépend de l'objectif du problème global. Dans cette étude, l'objectif du problème global étant de minimiser le temps total que mettent tous les AGV pour arriver à leur destination (temps d'attente compris), la valeur du plus court chemin doit être le temps minimum que met l'AGV pour arriver à sa destination. Cette fonction est la suivante :

$$\text{Min} \sum_{b_i \in \mathcal{B}} \sum_{b_j \in \mathcal{B}} d_{ij}^b X_{ij} + \sum_{b_i \in \mathcal{B}} \sum_{b_j \in \mathcal{B}} d_{icj}^b X_{icj} + \sum_{b_i \in \mathcal{B}} \varepsilon_i \quad (35)$$

6 Résultats

Les trois algorithmes ont été testés sur différents jeux d'essai basés sur le circuit FACT2 simplifié (FACT2s) représenté par la figure 3. Un circuit supplémentaire a été construit à partir du FACT2s. Pour ce faire, deux circuits FACT2s ont été accolés et reliés pour doubler la taille du circuit. Ce circuit est noté FACT2sd. Les différents jeux sont notés "FACT2s(d)_X_Y" avec :

- X le nombre d'AGV ;
- Y le numéro du jeu.

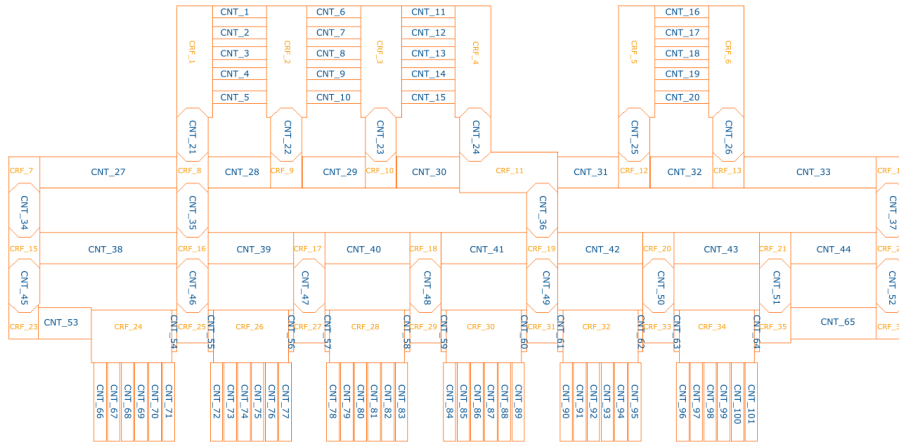


FIGURE 3 – Réseau FACT2 simplifié

Le nombre de "docking stations" est 6 pour FACT2s et 12 pour FACT2sd. Le nombre d'AGV varie de 2 à 16. Les résultats des algorithmes sont comparés avec ceux trouvés par CPLEX et

qui sont les solutions optimales. Ces résultats sont montrés dans le tableau 2. La colonne "C" représente le coût de la solution trouvée. La colonne "T" représente le temps de calcul en seconde. La colonne "G" représente la différence entre la valeur de la solution trouvée par l'algorithme et la valeur de la solution optimale trouvée par CPLEX.

Avec jusque 4 AGV, la solution optimale est trouvée dans un temps raisonnable. L'algorithme glouton trouve la solution optimale pour 5 jeux sur 7. Pour le dernier jeu avec 16 AGV, l'algorithme glouton et CPLEX ne sont pas capables de trouver une solution. Les algorithmes 0 et 1 sont beaucoup plus rapides et trouvent toujours une solution en moins d'une minute. La qualité des solutions est également plutôt bonne puisque le plus grand écart avec la solution optimale est inférieur à 10%.

Jeu d'essai	Algorithme 0			Algorithme 1		
	C	T(s)	G(%)	C	T(s)	G(%)
FACT2s_2_1	172.5	< 1	0	178.5	< 1	3.5
FACT2s_2_2	182.5	< 1	0	182.5	< 1	0
FACT2s_4_1	360	< 1	0	360	< 1	0
FACT2s_4_2	465.5	1	8.1	454	1	5.5
FACT2s_8_1	1001.5	2	9.2	954	3	4
FACT2sd_8_1	1078.5	10	9.8	1065	7	8.4
FACT2sd_16_1	2048	17	N/A	2083.5	36	N/A
FACT2sd_24_1	3675.5	51.48	N/A	3792.5	70.58	N/A

Jeu d'essai	Glouton			Cplex	
	C	T(s)	G(%)	C	T(s)
FACT2s_2_1	172.5	< 1	0	172.5	< 1
FACT2s_2_2	182.5	< 1	0	182.5	< 1
FACT2s_4_1	360	8	0	360	1
FACT2s_4_2	430.5	21	0	430.5	14
FACT2s_8_1	917	64134	0	917	31560
FACT2sd_8_1	1015	76780	3.3	982.5	16629
FACT2sd_16_1	N/A	N/A	N/A	N/A	N/A
FACT2sd_24_1	N/A	N/A	N/A	N/A	N/A

Nous pouvons remarquer que le temps de calcul de la solution augmente de manière exponentielle avec le nombre d'AGV. Ceci est dû au fait que la complexité de trouver le plus court chemin d'un AGV est de plus en plus grande avec le nombre d'AGV déjà planifiés ou en circulation sur le circuit. Il est ainsi très complexe de trouver le plus court chemin du dernier AGV lorsque l'avant dernier AGV sont déjà planifiés.

7 Conclusion

Cet article décrit différents algorithmes de résolution du problème de routage d'une flotte d'AGV. Ces algorithmes permettent de réduire considérablement le temps de calcul pour les jeux d'essai les plus grands, tout en donnant des solutions faisables suffisamment proches des solutions optimales pour les plus petites instances. Les expériences ont été menées sur des circuits avec jusque 64 AGV.

Une piste d'amélioration est d'introduire une limite de temps pour le calcul de chemin d'un AGV. Ce qui permettra de trouver des solutions pour les plus grandes instances en un temps maximal.

Références

- [1] S. Voß et R. Stahlbock D. Steenken. Container terminal operation and operations research - a classification and literature review. *OR Spectrum* 26.1, p. 3–49., 2004.
- [2] Nicolas Danloup, Benoît Trouillet, Thomas Bourdeaud’huy, and Armand Toguyéni. Optimisation de tournée de véhicule dans un environnement portuaire. In *GOL 2018 - International Conference on Logistics Operations Management*, pages 1–7, Le Havre, France, April 2018.
- [3] R. Stahlbock et S. Voß. Operations research at container terminals : a literature update. *OR Spectrum* 30.1, p. 1–52, 2008.
- [4] I. F. Vis et K. J. Roodbergen H. J. Carlo. Transport operations in container terminals : Literature overview, trends, research directions and classification scheme. *European Journal of Operational Research* 236.1, p. 1–13, 2014.
- [5] Thomas Bourdeaud’Huy Armand Toguyeni Nicolas Danloup, Benoit Trouillet. Heuristics for a vehicle routing problem in a port environment. *Green Supply Chain 2018 (GSC’2018)*, Jul 2018, Thessaloniki, Greece., 2018.
- [6] de Koster Vis, I. F. A. R. transshipment of containers at a container terminal : An overview. *European journal of operational research*, vol. 147, n 1, pages 1–16, 2003.