

# Routage des chariots cavaliers sur un terminal portuaire à conteneurs

Gaëtan Lesauvage, Stefan Balev, Frédéric Guinand

LITIS, Université du Havre

25 rue Ph. Lebon, BP 540

76058 Le Havre Cedex, France

gaetanlesauvage@gmail.com, {stefan.balev, frederic.guinand}@univ-lehavre.fr

**Mots-clés :** *problème de plus court chemins, graphe dynamique, terminal à conteneurs, optimisation multi-critères*

## 1 Introduction

Un terminal portuaire à conteneurs est un système complexe ouvert composé de plusieurs entités en interaction. Divers engins de manutention permettent de déplacer les conteneurs au sein du terminal afin de répondre le plus efficacement possible aux demandes des navires, trains ou camions en attente de chargement ou de déchargement.

Dans le souci d'améliorer la qualité de service fourni aux clients et de réduire le coût d'exploitation du terminal, l'itude des déplacements des véhicules s'avère d'une grande importance.

## 2 Problème

La gestion d'un terminal portuaire à conteneurs relève de plusieurs problèmes d'optimisation. Le but étant de minimiser les temps d'attente des clients tout en minimisant les coûts d'exploitation du terminal, un soin tout particulier est apporté à l'affectation des missions aux chariots cavaliers. De plus, afin de réduire le temps d'exécution de ces missions, il est également nécessaire de disposer les conteneurs de façon efficace. Les chariots cavaliers ont pour mission de déplacer des conteneurs à l'intérieur du terminal. Chaque mission comporte deux fenêtres de temps : l'une pour la collecte du conteneur et l'autre pour la livraison. Le non respect de ces fenêtres de collecte et de livraison entraîne un coût supplémentaire au terminal. Les routes des chariots cavaliers doivent donc prendre en compte ces fenêtres de temps.

Sur un terminal portuaire à conteneurs, les itinéraires possibles sont relativement réduits mais les chariots ne peuvent se croiser que sur les routes. En effet, ils enjambent les conteneurs stockés dans les travées et ne peuvent donc pas se croiser à l'intérieur de ces travées. Ces blocages peuvent causer des retards importants pour la réalisation d'une mission. Notre but est donc de définir les itinéraires des chariots cavaliers prenant en compte les temps d'attente en entrée de travées afin à la fois de minimiser la durée totale de parcours (déplacement + blocage), le coût de déplacement (distance) et la date d'arrivée du chariot (respect de la fenêtre de temps).

### 3 Graphe routier

Le graphe routier du terminal est donc un graphe partiellement FIFO, c'est-à-dire que les arcs modélisant les traversées sont FIFO alors que les arcs modélisant les routes ne sont pas FIFO. De plus la durée de parcours d'un arc dépend à la fois de la distance à parcourir et du temps. En effet, la durée de parcours d'un arc dépend directement de la vitesse du chariot (chariot plein ou vide) et également du trafic sur l'arc au moment de la traversée. On définit donc la durée de parcours de l'arc  $(i, j)$  par cette formule :

$$duree(i, j, t, v) = (distance(i, j) / vitesse(v, i, j, t)) + attente(i, j, t) \quad (1)$$

#### 3.1 Complexité

Le problème du plus court chemin en temps est connu pour être résolu en temps polynomial pour des graphes FIFO[?]. En revanche, le problème de plus court chemin en coût est NP-difficile[?]. Le problème ici étudié est un problème de plus court chemin en coût sur un graphe FIFO.

#### 3.2 Etat de l'art

Shortest Path Problem with Time Windows and Shortest Path Problem with waiting costs. Shortest path in delay time dependent networks.

(Dynamic algorithms : incremental, decremental, fully dynamic) FIFO Networks à non FIFO Networks Waiting models : Unrestricted waiting, Forbidden waiting, Source Waiting Minimum cost walk problem Problem complexity

### 4 Algorithme de routage

Optimisation locale des routes des chariots en tenant compte des routes des autres véhicules présentement établies. Comment rendre cette optimisation globale ? Lors de la planification d'une route, il faudrait également recalculer les routes des autres véhicules en prenant comme origine leur position actuelle. Puis recommencer dans un autre ordre.  $\Rightarrow O(n!)$ . Proposition d'un algorithme permettant la résolution du problème dans un temps acceptable ? Proposition d'un algorithme méta-heuristique ?

### 5 Résultats et perspectives