Affectation des opérations de déplacements de conteneurs : approche par colonies de fourmis

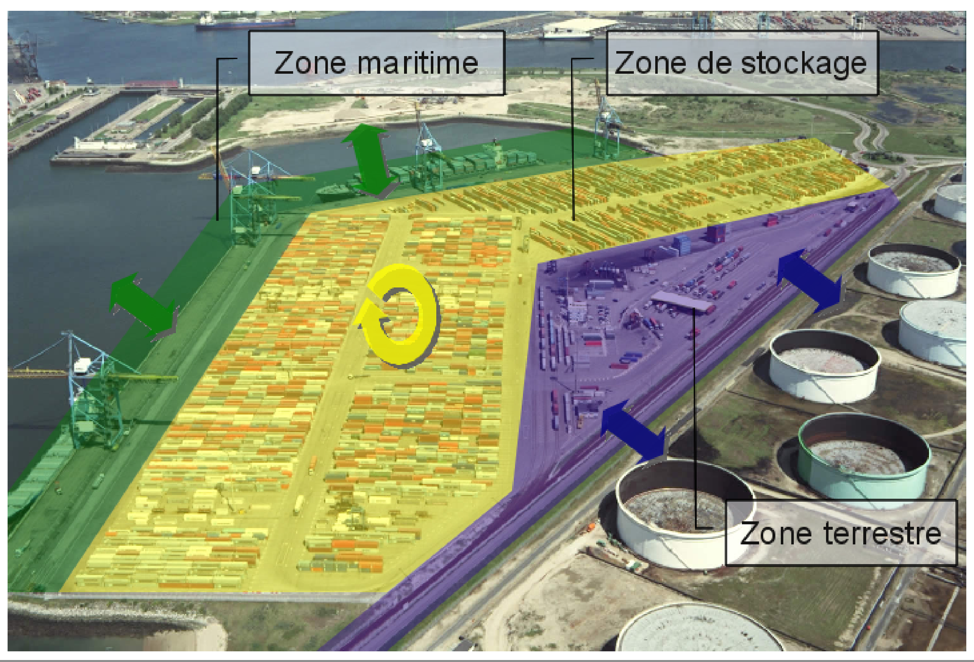
Gaëtan Lesauvage, Frédéric Guinand, Stefan Balev

Abstract

- Déterminer les listes ordonnées de déplacements de conteneurs, appelés missions, pour chaque chariot cavalier, afin de minimiser le temps de dépassement des fenêtres temporelles ainsi que la distance totale parcourue
- Problème des multiples voyageurs de commerce avec fenêtres de temps (*M-TSP-TW*)
- Prise en compte de la dynamique : arrivée de nouvelles missions, modification des contraintes, etc
- Résolution par colonie de fourmis

Contexte : terminal portuaire à conteneurs

Transfert de conteneurs entre 3 zones :



Chariots Cavaliers :

- Prise de conteneurs par le dessus
- Enjambent les travées
- Capables de (dé)charger directement des camions
- Stationnent au dépôt lorsqu'ils sont inutilisés
- Mission : déplacer un conteneur dans le terminal
- 2 phases :
  - collecte (Pickup)
  - livraison (Delivery)

Modélisation

Multiple-Traveling Salesman Problem with Time Windows

- Généralisation du problème classique de voyageur de commerce pour m voyageurs [Bek06]
- Chacune des n villes doit être visitée 1 fois par un des voyageurs
- Chaque voyageur débute et termine sa tournée au dépôt
- Soit $G = (V, A, w)$ le graphe orienté complet composé des n villes à visiter et de la fonction w_k de pondération des arcs, dépendante du voyageur k.
- Soit $T_v = \{T_v^{MIN}, T_v^{MAX}\}$ la fenêtre de temps associée à la ville v
- Objectif : minimiser la durée de dépassement des fenêtres de temps ainsi que la distance totale parcourue :
  - Dominance de Pareto
  - Priorisation
  - Linéarisation : dépassement$\cdot F_1 +$distance$\cdot F_2$

$\Longrightarrow$

Application au terminal à conteneurs

- m chariots cavaliers
- n missions
- $w_k(i,j) =$durée$_k(i,j)+$durée$_k(j,j)$
- dynamique et incertitude :
  - nouvelles missions
  - retards
  - pannes
  - ...

Méthode de résolution : approche par colonies de fourmis

- Modèle basé sur l'*Ant-System* de Dorigo et al. [Dor92]
- m fourmis, chacune avec sa propre phéromone
- Une fourmi peut coloniser un nœud du graphe s'il n'a pas été déjà colonisé par une des fourmis dans le tour courant
- La construction d'un tour consiste à choisir aléatoirement une fourmi parmi les m disponibles afin qu'elle colonise un sommet (eq. 2), tant que tous les sommets n'ont pas été visités
- Lorsque tous les sommets ont été colonisés, le tour est évalué et si le score est meilleur que le record courant :
  - les fourmis déposent de la phéromone sur les arcs de leur chemin (eq. 1)
  - le record est mis à jour
- La solution au problème sera le meilleur tour t parmi les T tours construits
- Règle de dépôt de phéromone :

$$\tau_{ij} = (1-\rho)\cdot\tau_{ij} + \rho\cdot\Delta\tau_{ij} \qquad (1)$$

$$\text{Avec } \Delta\tau_{ij} = \begin{cases} 1/L^* & \text{si } (i,j) \in \text{meilleur chemin global} \\ 0 & \text{sinon} \end{cases}$$

- Règle de transition proportionnelle :

$$p_{ij}^m = \frac{\left(\tau_{ij}^m\right)^\alpha \cdot \left(\eta_{ij}^m\right)^\beta \cdot \left(\frac{\tau_{ij}^m}{\sum_{l=1}^{M} \tau_{ij}^l}\right)^\gamma}{\sum_{k \in V_{nonvisites}} \left(\tau_{ik}^m\right)^\alpha \cdot \left(\eta_{ik}^c\right)^\beta \cdot \left(\frac{\tau_{ik}^m}{\sum_{l=1}^{M} \tau_{ik}^l}\right)^\gamma} \qquad (2)$$

Avec :

- τ_{ij}^m : niveau de phéromone de la fourmi m sur l'arc (i,j)
- $\eta_{ij}^m = \frac{1}{t^m(d_{ij})\cdot F_1 + l_{ij}^m \cdot F_2 + e_{ij}^m * F_3}$
- $t^m(d_{ij})$: temps de parcours de la distance entre les villes i et j par la fourmi m.
- l_{ij}^m : retard (lateness) de la fourmi m à l'arrivée à j en venant de la ville i (dépend de la date de départ de la ville i).
- e_{ij}^m : temps d'attente (earliness) de la fourmi m avant de pouvoir visiter la ville j en venant de i (dépend de la date de départ de la ville i).

Résultats préliminaires

Simulations réalisées avec D²CTS [Les2011]. Paramètres : $F_1 = 1$, $F_2 = 5$, $F_3 = 10$, $\alpha = 1$, $\beta = 1$, $\gamma = 1$, $\rho = 0.1$, $it_{MAX} = 10^6$

$m = 2$			
Problem	Score (**optimal**)	Itérations	Temps (Branch & Bound)
P_1	**5832** (5832)	15150	1.43s (15.83s)
P_2	**4699** (4699)	8541	0.62s (7.23s)
P_3	**4294** (4294)	206869	8.55s (6.71s)

$m = 3$			
Problem	Score (**optimal**)	Itérations	Temps (Branch & Bound)
P_1	**2742** (2742)	404550	19.59s (12m04s)
P_2	2830 (2827)	542903	27.22s (9m46s)
P_3	3106 (3103)	227260	10.27s (12m39s)

Table : Résultats pour 3 problèmes de taille $n = 10$

Réferences

[Dor92] M. Dorigo.
Optimization, Learning and Natural Algorithms.
PhD thesis, Politecnico di Milano, Italy, 1992.

[Bek06] T. Bektas.
The multiple traveling salesman problem: an overview of formulations and solution procedures.
Omega, 34(3):209–219, jun 2006.

[Les2011] Gaëtan Lesauvage, Stefan Balev, and Frédéric Guinand.
D²CTS : A dynamic and distributed container terminal simulator.
In *HMS 2011 : The 13rd International Conference on Harbor, Maritime & Multimodal Logistics Modelling and Simulation*, September 2011.