

Gestion dynamique des activités des chariots cavaliers sur un terminal portuaire à conteneurs en environnement incertain - approche par intelligence collective -

G. Lesauvage



Unité de Formation et de Recherche des Sciences et Techniques



Laboratoire d'Informatique et du Traitement de l'Information et des Systèmes

Plan

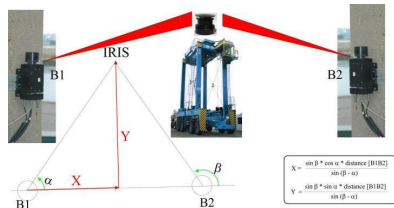
- 1 Description du système
- 2 Problèmes de tournées de véhicules : état de l'art
- 3 Ant Colony et gestion des chariots cavaliers
- 4 Simulateur
- 5 Résultats préliminaires
- 6 Conclusion

Plan

- 1 Description du système
- 2 Problèmes de tournées de véhicules : état de l'art
- 3 Ant Colony et gestion des chariots cavaliers
- 4 Simulateur
- 5 Résultats préliminaires
- 6 Conclusion

Le projet CALAS

- Système de mesure laser
- 2 principales entreprises :
 - *Laser Data Technology Terminal*
 - Terminaux de Normandie
- 2 principaux laboratoires :
 - LMAH
 - LITIS



Objectifs du projet CALAS :

Connaître l'état du terminal en temps réel, c'est-à-dire à la fois la position des conteneurs et celle des engins de manutention.

Description d'un terminal

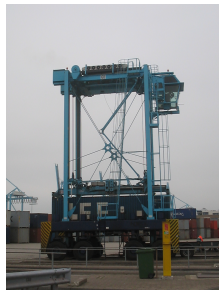
- Plateforme de transit de conteneurs
- 3 principales zones :
 - Zone maritime
 - Zone de stockage
 - Zone terrestre



- La zone de stockage est composée de longues travées de conteneurs empilés sur plusieurs niveaux

Engins de manutention

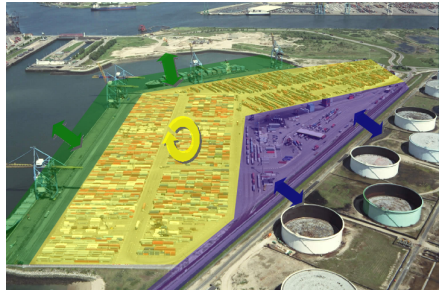
- Les chariots cavaliers sont des engins de manutention qui peuvent transporter un conteneur à la fois
- Ils sont capable d'enjamber une travée de conteneurs et de prendre (déposer) un conteneur sur le sommet d'une pile
- Une mission consiste à déplacer un conteneur d'un point A à un point B sur le terminal



Activités des chariots cavaliers

3 types de missions :

- Préparer le (dé)chargement d'un **navire**
- Préparer le (dé)chargement d'un **camion** ou d'un **train**
- Optimiser la zone de **stockage**

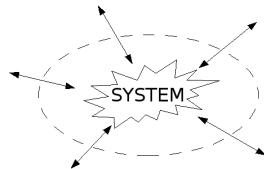


Système ouvert

Un système ouvert implique un environnement incertain.

Les flots entrants et sortants :

- ne dépendent pas uniquement du système lui-même
- affectent le système et le conduisent dans un nouvel état



Environnement incertain

Il existe plusieurs événements imprévisibles tels que :

- L'arrivée des missions
- L'heure d'arrivée des camions, des trains, et des navires
- Les déconnexions au sein du réseau routier (et ferroviaire)
- Les pannes des engins de manutention
- Le comportement humain

Plan

- 1 Description du système
- 2 Problèmes de tournées de véhicules : état de l'art
- 3 Ant Colony et gestion des chariots cavaliers
- 4 Simulateur
- 5 Résultats préliminaires
- 6 Conclusion

VRPTW[1]

- **V**ehicle **R**outing **P**roblem

- **T**ime **W**indows

Objectif

Optimiser les routes de livraisons de chaque véhicule

exemple de VRPTW : la fabrique de jouets

- Une usine fabrique des jouets et des véhicules doivent les livrer à un ensemble de points de vente
- Les boutiques sont réparties sur tout le territoire et les marchandises sont transportées par camion
- Chaque camion a une capacité limitée et débute sa tournée au dépôt
- les livraisons doivent intervenir au cours d'un intervalle défini et si un camion arrive trop tôt, il devra attendre

DVRPTW[6]

- Version dynamique du VRP

Objectif

Optimiser les nouvelles routes de chaque véhicule sans tout recalculer de zéro

La fabrique dynamique de jouets

- problème statique de la fabrique de jouets
- même si une tournée est déjà en cours les boutiques peuvent demander de nouvelles livraisons

(D)PDP

- DVRP où les marchandises doivent d'abord être collectées avant d'être livrées

Objectif

Optimiser à la fois les routes de collecte et de livraison

Problème du postier

- Une entreprise postale emploie un ensemble de postiers
- Ils doivent collecter le courrier dans les boîtes de la compagnie
- Puis, ils doivent livrer les clients aussi vite que possible

DSCPDPTW

- Pickup and Delivery Problem
- Capacité unitaire
- Fenêtre de temps : dépend à la fois des chariots cavaliers et des camions/trains/navires
- Dynamique :
 - Un plan peut changer à chaque instant : les véhicules commencent leur tournée de n'importe où (pas forcément du dépôt)
 - Pannes des chariots cavaliers : le nombre de ressources peut changer
 - Heure d'arrivée des camions/trains/navires : il est impossible d'être sûr qu'ils vont respecter leur fenêtre de temps
 - ...

Une solution calculée doit pouvoir être annulée à chaque instant !

DSCPPDTW (2)

- 2 problèmes :
 - Minimiser les déplacements des chariots cavaliers : problème de plus court chemin
 - Minimiser le temps d'attente des clients : problème d'ordonnancement

Dépendances des problèmes

ordonnancement approprié	⇒	concept du plus court chemin
planifier des plus court chemins	⇒	optimiser l'ordonnancement

Plan

- 1 Description du système
- 2 Problèmes de tournées de véhicules : état de l'art
- 3 Ant Colony et gestion des chariots cavaliers**
- 4 Simulateur
- 5 Résultats préliminaires
- 6 Conclusion

Ant Colony Optimization[3]

- ACO est une méta-heuristique
- ACO fait émerger une solution grâce aux parcours locaux d'insectes artificiels dans l'espace de recherche
- ACO est adapté à la nature dynamique du problème :
 - Rétro-action positive : les fourmis déposent de la phéromone en fonction de la qualité de la solution
 - Rétro-action négative : les pistes de phéromone s'évaporent progressivement

Ordonnancement grâce à Ant Colony

Ant Colony avec **une colonie** fourni une liste ordonnée de missions à accomplir

Ordonnancement grâce à Ant Colony

Ant Colony avec **une colonie** fourni une liste ordonnée de missions à accomplir

Problème

Comment affecter une mission à un chariot cavalier en particulier ?

Ordonnancement grâce à Ant Colony

Ant Colony avec **une colonie** fourni une liste ordonnée de missions à accomplir

Problème

Comment affecter une mission à un chariot cavalier en particulier ?

Fourmis colorées[2] :

- chaque chariot cavalier représente une colonie avec sa propre couleur
- les fourmis sont attirées par la phéromone de leur propre colonie
- les fourmis sont repoussées par la phéromone des colonies étrangères

Ordonnancement grâce à Ant Colony

Ant Colony avec **une colonie** fourni une liste ordonnée de missions à accomplir

Problème

Comment affecter une mission à un chariot cavalier en particulier ?

Fourmis colorées[2] :

- chaque chariot cavalier représente une colonie avec sa propre couleur
- les fourmis sont attirées par la phéromone de leur propre colonie
- les fourmis sont repoussées par la phéromone des colonies étrangères

Ant Colony avec **plusieurs colonies** fournit une liste ordonnée de missions par chariot cavalier

Graphe de missions

Le graphe orienté peut être modélisé de la façon suivante :

- Sommets :
 - 1 mission = 1 sommet
 - 1 chariot cavalier = 1 noeud coloré connecté à toutes les missions compatibles
- Arcs colorés :
 - Compatibilité entre 2 missions pour un chariot cavalier

Compatibilité des missions

Une mission m_a est dite **compatible** avec une mission m_b si la fenêtre de temps de m_a commence avant celle de m_b

Exemple de construction d'un graphe de missions (1)

Exemple

- Missions :

Nom	Début	Fin
m0	5:00	6:00
m1	5:30	6:00
m2	7:00	9:00
m3	6:00	7:30

- Chariots cavaliers :

Nom	Couleur	Compatibilité
s0	vert	m0, m1, m2, m3
s1	bleu	m0, m3

Exemple de construction d'un graphe de missions (2)

[5:00 - 6:00]

m0

[5:30 - 6:00]

m1

[6:00 - 7:30]

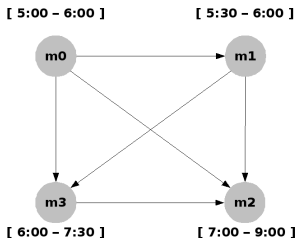
m3

[7:00 - 9:00]

m2

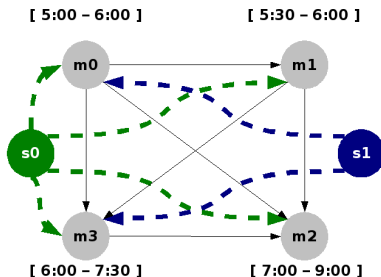
1 mission \iff 1 sommet

Exemple de construction d'un graphe de missions (3)



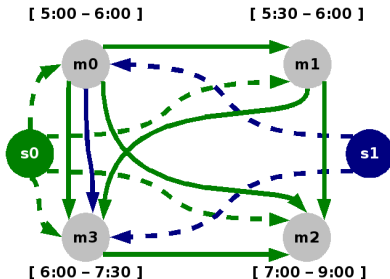
1 arc entre 2 missions compatibles

Exemple de construction d'un graphe de missions (4)



Ajout de sommets représentant les chariots cavaliers et connection avec les missions compatibles

Exemple de construction d'un graphe de missions (5)



Ajout ou coloration d'arcs entre les noeuds selon leur connectivité avec les véhicules

Description de l'algorithme

Algorithme principal

début

| pour chaque colonie c faire

| | pour chaque fourmi a de c faire

| | | choisir une destination non visitée en fonction de la trace de phéromone

| | | se déplacer vers elle en fonction de la vitesse de a

| | | déposer de la phéromone en fonction de la qualité de la destination

| | fin pour

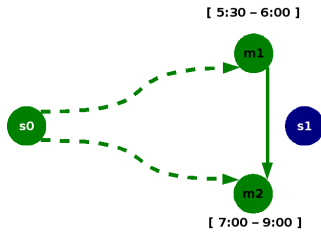
| fin pour

| évaporation

fin

Solution

- La solution correspond à la coloration des noeuds.
- Lorsqu'un chariot cavalier est disponible, il choisit la mission de sa couleur qui a le taux de pheromone le plus élevé.
- Les missions choisies sont supprimées du graphe et l'algorithme continue de fonctionner.
- Les noeuds des nouvelles missions sont dynamiquement ajoutés sur le graphe afin d'être pris en compte par l'algorithme.

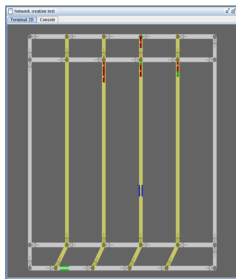


Plan

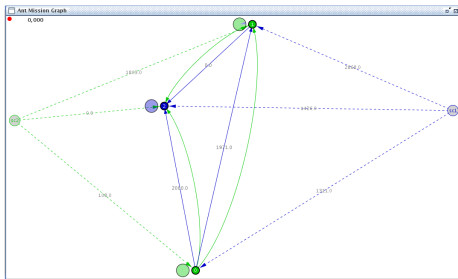
- 1 Description du système
- 2 Problèmes de tournées de véhicules : état de l'art
- 3 Ant Colony et gestion des chariots cavaliers
- 4 Simulateur**
- 5 Résultats préliminaires
- 6 Conclusion

2 vues parallèles du système

- Implémentation du terminal



- Modelisation d'ACO[4]



ACO \iff Terminal

Les effets d'ACO doivent être visibles sur le terminal et l'état du terminal doit affecter le graphe de missions d'ACO

Gestion de la dynamique

Un fichier scénario est lu tout au long de l'exécution de la simulation. Il contient des événements dynamiques.

Mesure de la dynamicité

Selon A. Larsen[5], il est possible de quantifier la dynamicité d'un scénario grâce à ces deux formules :

- Degree of Dynamism (dod) = $\frac{\eta_d}{\eta_s + \eta_d}$

- Effective Degree of Dynamism (edod) = $\frac{\sum_{i=1}^{\eta_d} t_i}{\eta_s + \eta_d}$

η_s : nombre de requêtes statiques ;

η_d : nombre de requêtes dynamiques.

Plan

- 1 Description du système
- 2 Problèmes de tournées de véhicules : état de l'art
- 3 Ant Colony et gestion des chariots cavaliers
- 4 Simulateur
- 5 Résultats préliminaires**
- 6 Conclusion

Résultats préliminaires

- Tester la pertinence de notre modélisation et de notre algorithme sur des données simulées
- En faisant varier la dynamicité

	Statique	Semi-dynamique	Dynamique
<i>dod</i>	0	0.5	1
<i>edod</i>	0	0.25	1
Fin de simulation	22693	22276	22693
Fenêtres de temps dépassées	3	5	7
Pénalité de dépassement	6467	8477	12485

Le dépassement des fenêtres de temps ainsi que le temps de pénalité évoluent de la même façon que *dod* et *edod*.

Plan

- 1 Description du système
- 2 Problèmes de tournées de véhicules : état de l'art
- 3 Ant Colony et gestion des chariots cavaliers
- 4 Simulateur
- 5 Résultats préliminaires
- 6 Conclusion**

Conclusion

- Un problème hautement dynamique
- Dynamic Pickup and Delivery Problem with Time Windows
- Intelligence collective :
 - Modélisation sous forme de graphe
 - Ant Colony System
 - Fourmis colorées
- Un simulateur est en cours de développement et permettra de mesurer la performance de notre solution
- Les résultats préliminaires confirment que notre modèle est capable de gérer la dynamique

Conclusion

- Un problème hautement dynamique
- Dynamic Pickup and Delivery Problem with Time Windows
- Intelligence collective :
 - Modélisation sous forme de graphe
 - Ant Colony System
 - Fourmis colorées
- Un simulateur est en cours de développement et permettra de mesurer la performance de notre solution
- Les résultats préliminaires confirment que notre modèle est capable de gérer la dynamique

Merci de votre attention



G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte.
Static pickup and delivery problems: A classification scheme and survey.
TOP, 2007.



C. Bertelle, A. Dutot, F. Guinand, and D. Olivier.
Distribution of agent based simulation with colored ant algorithm.
In *14th European Simulation Symposium*, 2002.



M. Dorigo.
Learning and Natural Algorithms.
PhD thesis, Politecnico di Milano, Italie, 1992.



Antoine Dutot, Frédéric Guinand, Damien Olivier, and Yoann Pigné.
Graphstream: A tool for bridging the gap between complex systems and dynamic graphs.
In *EPNACS: Emergent Properties in Natural and Artificial Complex Systems*, 2007.



A. Larsen.
The Vehicle Routing Problem.
PhD thesis, Department of Mathematical Modelling, Technical University of Denmark, 2000.



Snezana Mitrovic-Minic.
The dynamic pickup and delivery problem with time windows.
PhD thesis, Simon Fraser University, 2001.