Dynamical Handling of Straddle Carriers Activities on a Container Terminal in an Uncertain Environment - Swarm Intelligence Approach -

G. Lesauvage

Unité de Formation et de Recherche des Sciences et Techniques

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

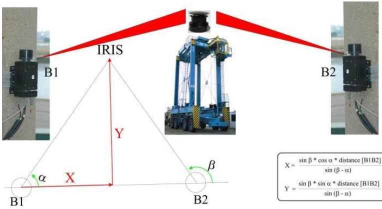
november 6th, 2009

Outline

Outline

The CALAS project

- Laser measure system and software
- 2 main companies :
  - Laser Data Technology Terminal
  - *Terminaux de Normandie*
- 2 main laboratories :
  - LMAH
  - LITIS



$$X = \frac{\sin\beta * \cos\alpha * \text{distance } [B1B2]}{\sin(\beta - \alpha)}$$

$$Y = \frac{\sin\beta * \sin\alpha * \text{distance } [B1B2]}{\sin(\beta - \alpha)}$$

Objective of the CALAS project :

To know the state of the terminal, in real time, for both containers and trucks location.

Terminal description

- Container terminal
- 3 main areas :
  - Ship handling
  - Stock area
  - Truck/Train handling

- Stock area contains many long rows of stacked containers

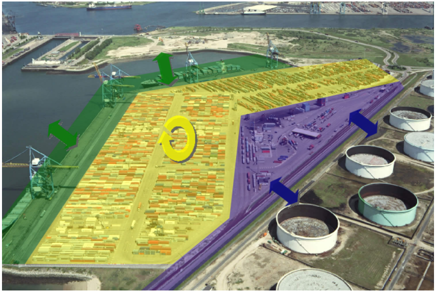
Handling trucks

- Straddle carriers are handling trucks which can carry on one container at a time
- They can treadle stacks of containers and take/drop a container from/on the peak of a stack
- One mission consist in moving a container from a place inside the terminal to another one

Container Terminal Activities

3 kinds of missions :

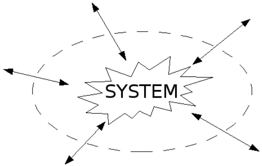
- Preparing a ship (un)loading
- Preparing a truck (un)loading
- Optimizing stock area



System dynamic

Open system means uncertain environment.
Incoming and outgoing flows :

- do not only depend on the system
- affect the system and lead it into a new state



Uncertain Environnement

There are several unpredictable events such as :

- Incoming missions
- Trucks and ships arriving time
- Road/rail network disconnections
- Straddle carriers failures
- Human behavior

Outline

VRPTW[1]

- **V**ehicle **R**outing **P**roblem
- **T**ime **W**indows

Goal

Optimizing the delivery routes of each truck

example of VRPTW : the toys factory

- The factory produces toys and its vehicles deliver a set of stores
- Stores are spread all over the country and goods are carried by trucks
- Every truck has a restricted capacity and starts from the factory depot
- Deliveries must occur during a time interval and if a truck comes too early, it will have to wait

DVRPTW[6]

- Dynamic

Goal

Optimizing the new routes of each truck without recomputing from scratch

Dynamic toys factory

- Toy factory problem
- While a schedule is running, stores are still allowed to ask for a delivery

(D)PDP

- DVRP where the goods have to be picked-up before being delivered

Goal

Optimizing both pickup and delivery routes

Pickup and Delivery Problem example : mail delivery problem

- A mail company employs a set of postmen
- They have to pickup mails from the mail boxes of the company
- Then, they must deliver them to their recipients as soon as possible

DSCPDPTW

- Pickup and Delivery Problem class
- Unit capacity
- Time windows : depends on both straddle carriers and boats/trucks/trains
- Dynamic :
  - A plan can change at anytime : vehicles may start from elsewhere than the depot
  - Straddle Carriers failures : the number of ressources can change
  - Trucks/trains/boats arriving time : we cannot be sure they will respect their time window
  - ...

A computed solution can be aborted at anytime !

DSCPPDTW (2)

- 2 problems :
  - Minimize straddle carriers moves : shortest path problem
  - Minimize customers delays : scheduling problem

Problem dependencies

appropriate scheduling	$\Rightarrow$	shortest path concept
scheduling shortest paths	$\Rightarrow$	reducing straddle carriers moves

Outline

Ant Colony Optimization[3]

- ACO is a meta-heuristic
- ACO makes a solution appear thanks to the run of artificial ants into the solution space
- ACO is adapted to the dynamic nature of this problem :
    - Positive feedback : ants spread pheromone according to solution quality
    - Negative feedback : pheromone track evaporates progressively

Scheduling with Ant Colony

Ant Colony with **one colony** provides a sorted list of missions to accomplish.

Scheduling with Ant Colony

Ant Colony with **one colony** provides a sorted list of missions to accomplish.

Problem

How to set a mission to a specific straddle carrier ?

Scheduling with Ant Colony

Ant Colony with **one colony** provides a sorted list of missions to accomplish.

Problem

How to set a mission to a specific straddle carrier ?

Colored ants[2] :

- every straddle carrier represents a colony with its own color
- ants are attracted by pheromones of their own colony
- ants are repulsed by pheromones of foreign colonies

Scheduling with Ant Colony

Ant Colony with **one colony** provides a sorted list of missions to accomplish.

Problem

How to set a mission to a specific straddle carrier ?

Colored ants[2] :

- every straddle carrier represents a colony with its own color
- ants are attracted by pheromones of their own colony
- ants are repulsed by pheromones of foreign colonies

Ant Colony with **many colonies** provides a sorted list of missions per straddle carrier.

Missions graph

The directed graph can be conceptualized as follows :

- Vertices :
  - 1 mission = 1 node
  - 1 straddle carrier = 1 colored node connected to all compatible missions
- Colored Arcs :
  - Compatibility between 2 missions for a straddle carrier

Mission compatibility

We say that mission m_a is **compatible** with mission m_b if the time window of m_a starts before the one of m_b

Example of a mission graph construction (1)

Example

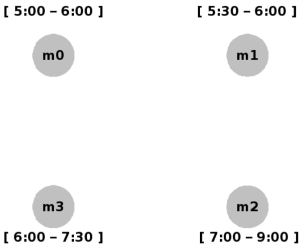
- Missions :

Name	Start	End
m0	5:00	6:00
m1	5:30	6:00
m2	7:00	9:00
m3	6:00	7:30

- Straddle Carriers :

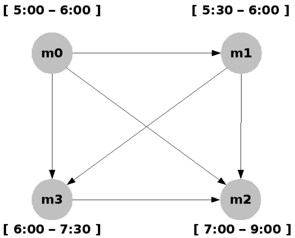
Name	Color	Compatiblility
s0	green	m0, m1, m2, m3
s1	blue	m0,m3

Example of a mission graph construction (2)



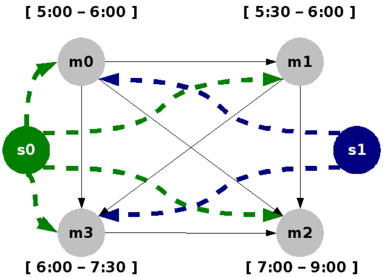
1 mission $\iff$ 1 vertex

Example of a mission graph construction (3)



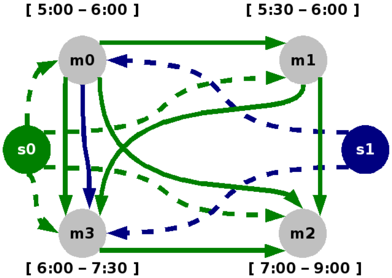
1 arc between two compatible missions

Example of a mission graph construction (4)



Adding nodes modeling straddle carriers and connecting them to every compatible vertices

Example of a mission graph construction (5)



Adding or Coloring edges between nodes according to their connectivity with the vehicles

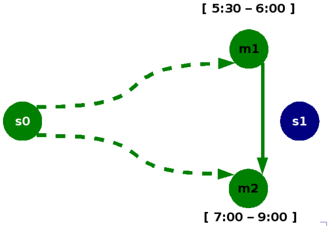
Algorithm description

Main algorithm

begin
| for each colony c do
| | for each ant a of c do
| | | choose an unvisited destination according to the pheromone track
| | | move towards it according to the speed of a
| | | spread pheromone according to the destination quality
| | end for
| end for
| evaporation
end

Solution

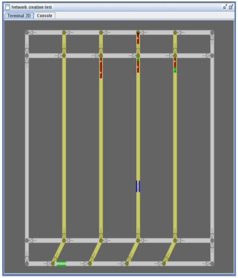
- The solution is the coloring of the nodes.
- When a straddle carrier is free, it chooses the mission of its color which has the highest pheromone rate.
- The chosen missions are removed from the graph and the algorithm continues running.
- The new missions vertices are dynamically added to the graph so they are taken into account by the ant algorithm

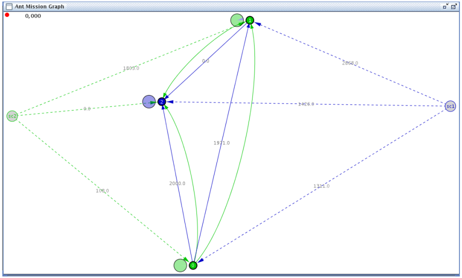


Outline

2 parallel views of the system

- Terminal implementation
- ACO modeling[4]





ACO $\Longleftrightarrow$ Terminal

Effects of ACO results must appear on the terminal and the terminal state must affect the ACO setting (mission graph)

Dynamism handling

A scenario file is read all along the execution of the simulation. It contains dynamic events.

Measure of dynamism

According to A. Larsen[5], we can measure how dynamic is a scenario by these two formulas :

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

η_s : number of static requests ;
η_d : number of dynamical requests.

Outline

Preliminary results

- Test the relevance of both our modeling and our algorithm on simulated data
- Function of the measures of dynamism

	Static	Half Dynamic	Dynamic
dod	0	0.5	1
edod	0	0.25	1
End time	22693	22276	22693
Number of overrun tw	3	5	7
Overrun time penalty	6467	8477	12485

The exceeded time windows and the time penalties evolve in the same way that *dod* and *edod*.

Outline

Conclusion

- A highly dynamic problem
- Dynamic Pickup and Delivery Problem with Time Windows
- Swarm intelligence with :
  - Graph modeling
  - Ant Colony System
  - Colored Ants
- A simulator is being developped and will allow to measure the solution relevance
- Priliminary results confirm that our modelling is able to handle dynamics

Conclusion

- A highly dynamic problem
- Dynamic Pickup and Delivery Problem with Time Windows
- Swarm intelligence with :
  - Graph modeling
  - Ant Colony System
  - Colored Ants
- A simulator is being developped and will allow to measure the solution relevance
- Priliminary results confirm that our modelling is able to handle dynamics

Thank you for your 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 algorihm.
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.