

# Dynamical Handling of Straddle Carriers Activities on a Container Terminal in Uncertain Environment

## - A Swarm Intelligence approach -

Stefan Balev, Frédéric Guinand, Gaëtan Lesauvage, Damien Olivier

**Abstract**—The CALAS project consists in a laser measure system allowing to localize precisely straddle carriers location in a box terminal. The information given by such a tool makes an optimization possible. In fact, a box terminal is an open system subject to dynamism, so many events can occur. They concern containers arrivals and departures. Within the terminal, straddle carriers are trucks which are able to carry one container at a time in order to move it through the terminal. We aim to optimize the straddle carriers handling to improve the terminal management. Moreover, missions come into the system in an unpredictable way and straddle carriers are handled by human. They can choose to follow the schedule or not. For these reasons, the exact state of the system, i.e the exact location of boxes is unknown. The optimizing system that we try to build must be fail-safe and adaptive. So, in this context, how to devote a mission to a straddle carrier ? We propose a simulation approach using a meta-heuristic based on Ant Colony to resolve the scheduling problem.

**Index Terms**—swarm intelligence, colored ant colony system, dynamic graph, multiple criteria optimization.

### I. SYSTEM DESCRIPTION

The CALAS project aims at localizing precisely handling trucks on a box terminal. It uses a laser measure system and a software which allows to deal with the data sent by lasers. This project is the result of a collaboration between *Laser Data Technology Terminal* company and the *Terminaux de Normandie* company. The goal of the CALAS project is to know the state of the terminal in real time, meaning both containers and trucks location.

A box terminal is divided into three main areas. Each part is a set of boxes rows where containers can be stacked up and these areas are linked by oriented roads. The first area is beside a channel where ships can tie to the dockside. It is an area bound to prepare the ship load or unload. The second area is used to load or unload trucks. The third part is a storing area linking the two others. Containers are moved into this area when a ship or a truck is unloaded, and containers are moved from this area when a ship or a truck is loaded. So, there are three kinds of missions:

- Preparing a ship loading / unloading
- Preparing a truck loading / unloading
- Optimizing storing area

The box terminal is an open system subject to dynamism. In fact, though a set of missions is known before starting the schedule, the main parts come into the system when the schedule has already been established. Moreover, trucks arriving time is not precisely known enough to forecast

container delivery. If the truck is late, straddle carriers must wait for it to complete the mission, meanwhile, it can not accomplish another mission. Human behavior also affects the system because straddle carriers are handled by human drivers who can choose to follow the schedule or not.

### II. VEHICLE ROUTING PROBLEMS CLASSES

Vehicle routing problems (VRP) are very common in some industrial processes, therefore, they are deeply studied. One or many vehicles must start from the depot, visit a list of customers, delivering (or picking-up) some goods, and coming back to the depot. The aim is to minimize the vehicle's routes. Many different subproblems belongs to the VRP class, such as capacitated Vehicle Routing Problem (CVRP) or Vehicle Routing Problem with Pickup and Delivery (VRPPD) by instance. Every subproblem contains a little variation of the main one, for example, it can have many depots, or vehicles must respect time windows... We distinguish static instances of these problems from the dynamics ones because the way to solve them are different.

#### A. Vehicle Routing Problem (VRP)

The Vehicle Routing Problem with Time Windows (see [1]) (VRPTW) consists in visiting a set of cities by a set of capacitated vehicles, optimizing path length. For instance, an Italian factory produces toys. It has to deliver a set of stores in order to make the goods sold. These stores are spread all over the country and goods are carried by trucks. Trucks capacity is restricted and they all start from the factory depot. Deliveries can only be done during a defined time interval. If a truck comes too early, it will have to wait. A solution to this problem should minimize the global length of the trucks runs.

The Dynamic VRPTW (DVRPTW) includes dynamism of the new orders. For the Italian factory example, if the stores can ask for deliveries when an already scheduled plan is running, then this problem belongs to DVRPTW class.

#### B. Pickup and Delivery Problem (PDP)

According to Berbeglia in [2], PDP contains 3 subclasses :

1) *Many to Many Pickup and Delivery Problems (M-MPDP)* : Here, the vehicles routes trend to pickup many objects to many locations. This kind of problem still relatively neglected because it is not frequently present in real situation.

2) *One to Many to One Pickup and Delivery Problems (1-M-1PDP)* : In this class, there are 2 different directions for the goods. They are first delivered to the customer. When the customer have done with them, he will ask for bringing back the goods to the depot. These problems must be with single or combined demands. In the first case, each customer ask for a delivery OR a pickup. With combined demands, a same customer can ask for both a delivery and a pickup.

3) *One to One Pickup and Delivery Problems (1-1PDP)* : This is the main subclass of Pickup and Delivery Problem, meaning the most frequently encountered problem in real life. It deals with picking-up one object at one location and delivering it to one destination. The main problem of this class is the Vehicle Routing Problem with Pickup and Delivery (VRPPD). In this problem, we have to compute the best routes for a fleet of vehicles in order to move objects on the graph. Every routes has to start and to end at the depot. It is not a 1-M-1PDP because here, each object  $o$  has its own pickup and delivery location.

When the problem deals with people, it is called Dial-A-Ride Problem (DARP). Some particular cases of VRPPD problems like the Stacker Crane Problem (SCP) are also common in practical life. This is a single vehicle with unit capacity problem. In an other subproblem, vehicles are allowed to temporarily drop their loads on specials locations called transshipment points to be able to better answer customers demands faster. This is called Vehicle Routing Problem with Pickup and Delivery and Transshipment.

When some request are not known in advance these static problems below may become dynamic. Those Dynamic Pickup and Delivery Problems (see [3]–[5]) (DPDP or DVRPPD) consist in optimizing vehicles routes in order to pickup a load somewhere, then to deliver it to its destination, adapting these routes to the new incoming orders without recomputing from scratch. Most of the time, DVRPPD has to handle time windows (DVRPPDTW). Indeed, to start a mission, vehicles have to wait the beginning of its time window. If it is not respected, the vehicle will have to wait for the right time and, meanwhile this vehicle becomes useless.

### C. A new problem

Our problem belongs to the Dynamic Vehicle Routing Problem with Pickup and Delivery and Time Windows (DVRPPD-TW). Several vehicles (straddle carriers) of unit capacity must accomplish missions (by moving containers within the box terminal). They also can use transshipment location to make the tasks more efficient. Straddle carriers can start from anywhere, ie. they don't have to start from

the depot. If a vehicle comes too early for picking up or delivering a container, it will have to wait the beginning of the missions time window. Furthermore, if a straddle carrier is late, meaning its time window is now closed, the mission must be aborted and a new one dealing with the same box will appears into the system.

Three interconnected problems must be solved:

- Minimize straddle carriers moves : shortest path problem
- Minimize ressources : clustering problem
- Minimize customers delays : scheduling problem

In order to realize a good scheduling, the system must integrate the shortest path concept. In the same way, scheduling shortest paths trends to reduce straddle carriers moves. Moreover, we have to define a quality of service level to satisfy customers while lowering operation costs. This is a dynamic large scale problem which requires a real time solution. We tried to develop an online algorithm based on Ant Colony Optimization (see [6], [7]) and more precisely on a colored version of this swarm algorithm (see [8]).

## III. ANT COLONY AND STRADDLE CARRIER HANDLING

Ant Colony (see [6], [7]) is a meta-heuristic which makes a solution appear thanks to the run of artificial ants into the solution space. The system is self-regulated. In fact, ants spread pheromone according to the solution quality (positive feedback) but the pheromone tracks evaporate progressively (negative feedback). The positive feedback makes the algorithm converge to a quality solution, and the negative feedback prevent it to block into a local extreme.

Ant Colony with one colony provides a sorted list of missions to accomplish (see [9]–[11]). The problem is to set a mission to a specific straddle carrier.

We have proposed to employ a solution using colored ants (see [8]). In our model, every straddle carrier represent a colony with its own color. Convergence is assured by the fact that ants are attracted by the pheromone of their own colony and repulsed by the pheromones of foreign colonies. These mechanisms will provide a sorted list of missions per every straddle carrier.

### A. Modelling

The problem can be conceptualized as such:

- let  $M$  the set of missions to schedule. Every mission has a defined time window. We say that mission  $m_i$  is prior to mission  $m_j$  if the time window of  $m_i$  starts before the one of  $m_j$
- let  $G(V, A)$  an oriented graph where  $V$  is the set of vertices and  $A$  the set of arcs.
- $\forall m \in M, \exists v \in V$
- $\forall c_i \in C$ , where  $C$  is the set of straddle carriers,  $\exists v_{k_i}$  in  $V$ .  $\forall v \in V, \exists a(v_{k_i}, v) \in A$ .
- $\exists a_{c_i}(v_i, v_j) \in A$ , if  $m_i$  is prior to  $m_j$  and if  $m_i$  and  $m_j$  can be executed by the vehicle  $c_i$ .
- $\forall v, \exists e(h_i, v_j) \in E$ , if  $m_i$  is prior to  $m_j$

In this way, a graph is set and ants colony system can be run to make the solution appears (see Fig. 1). The main

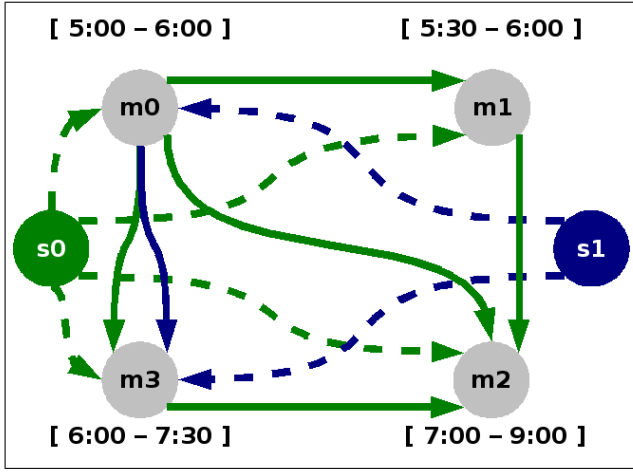


Fig. 1. Mission graph for 2 straddle carriers and 4 missions

```

1: for all colony c do
2:   for all ant of colony c do
3:     choose an unvisited destination according to the
       pheromone track
4:     move towards it according to the ant speed
5:     spread pheromone according to the destination
       quality
6:   end for
7: end for
8: evaporation

```

Fig. 2. Colored Ant System main algorithm

algorithm is described on Fig. 2.

### B. Assets and weaknesses of Ant Colony

Regarding the size of the solution space and the real time constraint, a metaheuristic is relevant for solving this problem.

The main asset of ant colony is to provide an anytime solution. It is an online algorithm which always tends to fit the problem whatever happens. Indeed, on one hand, ants reinforce the pheromone rates to get closer to the best solution. On the other hand, at the same time, evaporation process provides a feed back control of the algorithm by preventing it to get stuck into a local optimum and allowing dynamic events to be handled.

Ant colony deals with evaporation, solution evaluation, ants quantity and speed, dynamic events, etc... It makes a lot of parameters. Here is a major weakness of this metaheuristic. Solution quality closely depends on these interdependent settings. Although optimizing those parameters is a whole subject of research, we have been trying to make them self-adaptive. We use a local based method to adapt some of these parameters online.

### C. Division of labour

As there are several distinct colonies in a local and distributed system, there is no way to use a pheromone

spreading process based on a global characteristic. In fact, in this architecture, a colony can not evaluate the quality of its own solution compared with the ones of the other colonies. So, we must use the same pheromone spreading process for each colony. However, we are able to adapt the quantity of pheromone spread by ants of a colony according to the corresponding vehicle skill for a task. Indeed, in a fleet of  $n$  vehicles, we can increase the quantity of pheromone spread by a vehicle  $v_i$  ( $i \in [1..n]$ ) for tasks concerning a specific area in the box terminal and decrease the quantity spread for the tasks located into the others area. At the same time, we do the opposite for a vehicle  $v_j$  ( $\forall j \neq i$ ). In this way, we try to specialize the vehicle in a kind of tasks and we are able to regulate these quantities by taking into account both the preference and the distance criteria.

This regulation keeps the benefit of allowing this specialized vehicle to take a mission for which it is not specialized. It is really important in some cases where the number of missions is high because this regulation prevents the system from having unused vehicles in an area of the terminal and unaffected missions, close of the end of their time window, in an other area. In this way, we are getting close to reproduce the division of labour in castes of a real ant colony.

### D. Reducing resources

Always in a cost lowering purpose, we try to decrease the number of straddle carriers in the system. Our current solution to the entire problem tends to distribute the missions upon all the vehicles. So, every vehicle has almost the same activity rate. But if this rate is under a defined target, it is possible to conclude that a vehicle could be removed. Otherwise, if the rate is greater than the target, it is possible to say that a new vehicle should be added to the fleet.

The target rate must be computed by taking into account several facts. First, it has to deal with the quality of service. Indeed, the system must answer the request before the end of their time windows. Furthermore, if a vehicle is ready to serve several missions before the beginning of their time windows, it means that this vehicle is maybe superfluous, and so that the target rate must be raised. On the other hand, the target rate has to deal with other criteria like the covered distance of a vehicle per mission or the ratio between the number of vehicles and the number of missions, and it has to set these criteria against the penalties of transcended time windows. Measuring the time of inactivity of every straddle carrier may also lead the optimization. Concerning this last criterion, we must interrelate the time of inactivity with the penalties of transcended time windows.

So as for the missions arrivals into the system, the number of vehicles is subject to dynamicity. A vehicle can break down and then must be sent to the maintenance. In function of the failure seriousness, we can estimate the time needed to repair the vehicle and so make it available for routing. We take a rate of fault into account for optimizing the number of vehicles into

#### IV. SIMULATOR

The simulator has two main parts. The first one is the terminal simulation (see Fig. 3), and the second one is the Colored ant Colony Optimization System (see Fig. 4). The first part contains an implementation of the terminal structure and components. Roads and crossroads provide the network of the terminal on which straddle carriers will be able to go. Some of these roads may contain containers. Quay crane locations are represented by these specials roads, so as the trucks handling locations. This terminal is built at the very beginning of the simulation. A scenario file is read to set the terminal configuration. The second part of the simulator contains the algorithmic view of the simulation, i.e the dynamic mission graph. In this way, it shows how the missions are chosen by the vehicles. This part of the simulator uses GraphStream<sup>1</sup> toolkit which allows to handle dynamic graph easily.

The simulator uses a discrete motor which has to iterate every object of the simulation on every single step. Throughout the simulation, the scenario file is read and some dynamic events are sent back to both terminal and Ant Colony views. In this way, the system can simulate the dynamicity of the incoming missions and of the vehicles availability.

In order to have relevant tests and results, we have to define several levels of dynamicity. But how to value a such level? In [12], Allan Larsen point out two main ways to measure the degree of dynamicity.

First, the degree of dynamism (*dod*) (see [13]) is the ratio between the number of dynamical requests and the overall ones. The main weakness of this measure is that it does not take into account the arrival time of these requests into the system. Indeed, with *dod* if the requests come into the system at the beginning of the day, the system is as dynamic as if they come late in the day. Yet, the later these requests are known, the shorter is the delivery delay. This lateness impacts on the performance of the system.

For this reason, Larsen and al. in [12] defined the effective degree of dynamism (*edod*) by this formula :

$$edod = \frac{\sum_{i=1}^{\eta_d} \frac{t_i}{T}}{\eta_d + \eta_s} \quad (1)$$

Here,  $\eta_s$  and  $\eta_d$  are respectively the number of static and dynamic requests, and  $t_i$  is the time  $i$  (with  $0 < t_i < T$ ) and  $T$  is the time of the simulation end. These measure take into account the average of the incoming time of the requests into the system. The more the dynamical requests come late, the more *edod* will be high. If *edod* = 0, then the system is totally static. Else, if *edod* = 1, then the system is purely dynamic.

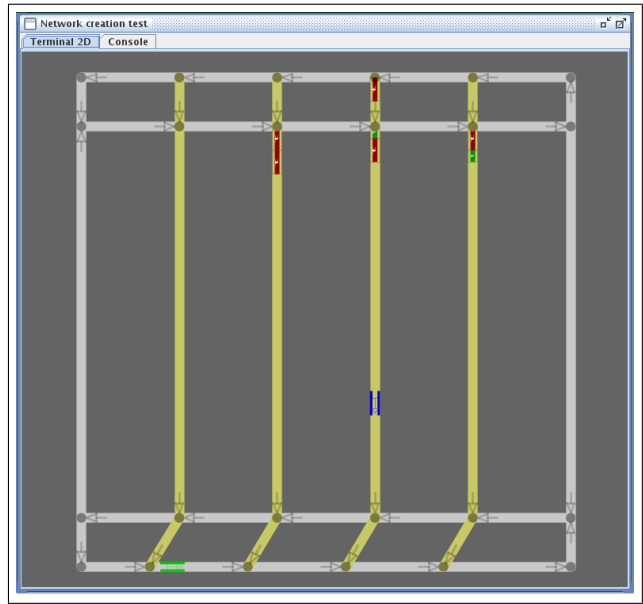


Fig. 3. Terminal view in the simulator

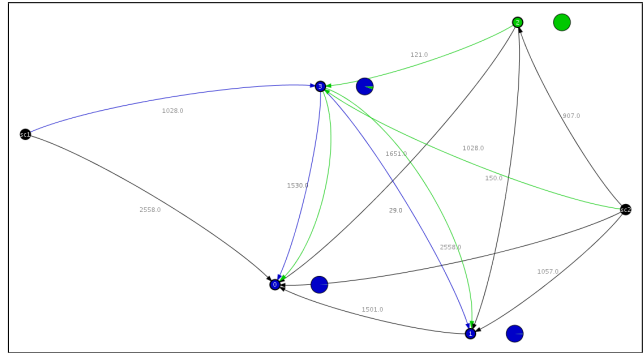


Fig. 4. Mission graph view in the simulator

Every straddle carriers on the terminal simulation received a schedule from the Colored Ant Colony System. Then they act in function of it and move to their pick-up location. Once they have picked-up their container, they move to the delivery location to achieve their mission. At the same time, the mission graph is dynamically updated and the colored ants keep colonizing it.

Simulator gives information about each mission like its length, container, straddle carrier

## V. CONCLUSION

The problem to solve belongs to the Dynamic Pickup and Delivery Problem with Time Windows class. However, it does not exactly fit. So it is an original unsolved problem. We propose to solve it using swarm intelligence methods. An Ant Colony System is being developed. It uses colored ants and a graph modeling in order to plan a schedule. Moreover, we are trying to minimizing the number of vehicles into the fleet in order to both maintain a sufficient quality of service and reduce costs. A simulator able to reflect a such system and to handle dynamic events is being developed. We are actually collecting data in order to compare the performance of our system into a

box terminal environment with the current optimization used in a terminal of the seaport of Le Havre in France.

## REFERENCES

- [1] L. Bianchi, "Notes on dynamic vehicle routing - the state of the art," *Technical Report, IDSIA*, 2000.
- [2] I. G. G. Berbeglia, J.-F. Cordeau and G. Laporte, "Static pickup and delivery problems: A classification scheme and survey," *TOP*, 2007.
- [3] S. Mitrovic-Minic, "The dynamic pickup and delivery problem with time windows," Ph.D. dissertation, Simon Fraser University, 2001.
- [4] M.-M. S. and G. Laporte, "Waiting strategies for the dynamic pickup and delivery problem with time windows," *Transportation Research Part B*, 2004.
- [5] M.-M. S., "Pickup and delivery problem with time windows : A survey," *Technical Report 1998-12, SCS Simon Fraser University*, 1998.
- [6] M. Dorigo, "Learning and natural algorithms," Ph.D. dissertation, Politecnico di Milano, Italie, 1992.
- [7] M. D. et L.M. Gambardella, "Ant colony system : A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, 1997.
- [8] F. G. D. O. C. Bertelle, A. Dutot, "Distribution of agent based simulation with colored ant algorithm," in *14th European Simulation Symposium*.
- [9] A. R. R. Montemanni, L.M. Gambardella and A. Donati, "A new algorithm for a dynamic vehicle routing problem based on ant colony system," *Technical Report, TR-23-02, IDSIA, Galleria 2, Manno, 6928, Switzerland*, 2004.
- [10] R. F. H. B. Bullnheimer and C. Strauss, "Applying the ant system to the vehicle routing problem," in *In Proceedings of the 2nd International Conference on Metaheuristics - MIC97*.
- [11] R. H. B. Bullnheimer and C. Struss, "An improved ant system algorithm for the vehicle routing problem," *Annals of Operations Research*, 1999.
- [12] L. A., "The vehicle routing problem," Ph.D. dissertation, Department of Mathematical Modelling, Technical University of Denmark, 2000.
- [13] O. B. G. M. Karsten Lund and J. M. Rygaard, "Vehicle routing problems with varying degrees of dynamism," *Technical report, IMM, The Department of Mathematical Modelling, Technical University of Denmark*, 1996.