Ant Colony Optimization for Solving the Container Stacking Problem:

Jalel Euchi
University of Gafsa

Adnan Yassine
Université du Havre

International Journal of Applied Logistics
Volume 6 • Issue 2 • July-December 2016

Ant Colony Optimization for Solving the Container Stacking Problem:
Case of Le Havre (France) Seaport Terminal

Jalel Euchi, LOGIQ Laboratory, Sfax University, Sfax, Tunisia
Riadh Moussi, Kairouan University, Kairouan, Tunisia
Fatma Ndiaye, University of Le Havre, Le Havre, France
Adnan Yassine, University of Le Havre, Le Havre, France

ABSTRACT

In this paper, the authors study the Container Stacking Problem (*CSP*) which is one of the most important problems in marine terminal. An optimization model is developed in order to determine the optimal storage strategy for various container-handling schedules. The objective of the model is to minimize the distance between vessel berthing location and the storage positions of containers. The *CSP* is solved by an efficient ant colony algorithm based on *MAX-MIN* ant system variant. The performance of the algorithm proposed is verified by a comparison with *ILOG CPLEX* for small-sized instances. In addition, numerical results for real-sized instances proved the efficiency of the algorithm.

KEYWORDS
Ant Colony Algorithm, Container Staking Problem (CSP), Port Container Terminal

1. INTRODUCTION

The idea to deliver goods in boxes or "containers" was originally developed by Malcolm Mclean in the 1950 which represents the first apparition of the principle of containerization. The development of containerization in the North Atlantic in 1965 and its gradual spread to the container has subsequently become a standardized box whose standards were set in 1974 by the *ISO* (International Standards Organization). The apparition of this principle was followed by the emergence of multimodal transport which consists of distributing goods via multiple types of transportation: ship, road, rail, etc.., and by the computerization and automation which allows the acceleration of the movement of goods and reducing costs traditionally associated.

Over the last four decades, the use of containers for international maritime transport has dramatically increased. An efficient container-handling at seaport terminal is very important for reducing transportation costs and maintaining shipping schedules.

Recently, given the competition among ports, the improvement of customer service became a serious problem for seaport terminal. One of the performance measures of customer service is the berthing time or throughout time. This time is based on the time for loading and unloading containers. In order to reduce the loading time, it is necessary to select the best storage locations for containers. This operation is a very important decision problem in a container terminal which is known by *CSP*. According to Kap et al. (2000) "the *CSP* consists in determining optimal positions or exact locations of containers from the empty slots in order to make efficient loading onto a ship, truck or train".

DOI: 10.4018/IJAL.2016070104

In this paper, we propose a new and original mathematical model for the *CSP* problem which is based on: port restrictions and practical problem (e.g. case study section). The originality of the proposed model is gained by the presentation of several physical and logical constraints of port: First, in our work, we choose the best stack in the terminal to store a container unlike to other works that they choose the best block and the best side in it. Second, for each stack, containers are stored in the decreasing order of their departure time. This constraint is very important for avoiding the unproductive movements which are very expensive. At third, containers are stored by respecting the constraint of size compatibility. The implementation phase consists in solving the proposed mathematical model. We know that the *CSP* is an *NP-hard* problem. Generally, two different strategies can be investigated to resolve this type of problems: the exact algorithms and the meta-heuristics methods.

To the best of our knowledge, no exact solutions are proposed to solve these variant of problems. Several metaheuristics procedures are proposed to solve the *CSP* problems, either metaheuristics optimization via memory as the Tabu search (Erhan and Peter (2006), etc. or the evolutionary algorithms metaheuristics as the Genetic algorithm (Peter and Erhan (2001)), Hybrid Genetic algorithm and Simulated Annealing (Moussi et al. (2012)).

For the reason of the complexity of *CSP,* we propose an ant colony algorithm based on *MAX-MIN* ant system variant. To our knowledge ant algorithm was not previously applied to resolve *CSP* or any similar problem in seaport terminal.

ACO algorithms have been inspired by colonies of real ants, which deposit a chemical substance (called pheromone) on the ground. This substance influences the choices they make: the larger amount of pheromone is on a particular path; the larger probability is that an ant selects the path. Artificial ants in *ACO* algorithms behave in similar way see Marco (1992).

Our contribution is twofold. First, we propose a new mathematical formulation for the CSP problem, our work is based on communication with port agents as well as observation of actual behavior in port. The choice of the mathematical model was based on the recommendations of the port handling staff to facilitate storage and transfer of the container in the terminal. Second, the contribution of the paper is the development of an efficient ant colony with high-quality solution produced. The problem becomes more complex, which implies that the choice of a good metaheuristics can provide a good result, with a major contribution, which the paper presents a case study of Le Havre seaport (France).

The rest of this paper is organized as follows: a brief overview and related literature is presented in Section 2. The mathematical model representing the concrete problem is formulated in Section 3. The solution methodology based on Ant algorithm is developed in Section 4. The results of numerical simulations and a comparative study are presented in Section 5. Section 6, presents our computational case study in Le Havre port (France), and finally Section 7 is devoted to conclusion.

2. LITERATURE REVIEW

Various aspects of the operational problems in container terminal are developed in literature. Steenken et al. (2004) have described and classified the main logistic processes and operations in container terminals. Stahlbock and Voß (2008) have presented the current state of the art in container terminal operations and operation research.

The storage of containers is a critical resource in container terminals. The loading sequence of containers affects significantly the productivity of port operations. However, an optimal allocation of containers allows a sequence of optimal loading. Recently, Moussi et al. (2015) represent a hybrid ant colony and simulated annealing algorithm to solve the container stacking problem at seaport terminal. Also in the paper of Ndiaye (2014) consider a modern container terminal which uses straddle carriers instead of internal trucks. They propose mathematical model which takes into account operational constraints and they minimize the total distance traveled by straddle carriers between the quays and the container yard.

In this section, we present some works treated the *CSP* and other similar problems in port container terminal. Among the first studies that have dealt with the *CSP*, McDowell et al. (1985) proposed a mathematical model to determine a pattern of efficient storage. Sculli and Hui (1988) have developed a study by simulation. The objective of the simulations was to measure the performance of storage Containers considering the size of the storage area, the stack height, container types and different storage policies.

Dayama, Niraj Ramesh, et al. (2014) consider an optimization problem of sequencing the operations of cranes that are used for internal movement of containers in maritime ports. They apply a MIP model to evaluate their appropriateness for various possible situations.

Taleb-Ibrahimi et al. (1993) applied a strategic and tactic approach to implement a plan to find a minimum storage space required for export containers (export container is container sent from a client to the storage and waiting delivery to a ship). At the strategic level, they assured the choice of technologies to be applied and that the terminals to be treated, and at the operational level, they proposed a strategy for minimizing the amount of handling work in the storage area.

Kap and Hong (1999) proposed a mathematical model to allocate storage space for import containers using the segregation strategy in order to minimize the number of unproductive moves.

Peter and Erhan (2001) developed a mathematical optimization model for the *CSP*. They proposed a heuristic method with a *GA*. Erhan and Peter (2006) treated the same problem with an improved algorithm. Indeed, they developed a *GA*, a Tabu Search algorithm (*TS*) and a hybrid algorithm between *TS* and *GA*.

Chuqian et al. (2003) treated the Storage Space Allocation Problem (*SSAP*) (The *SSAP* is defined as the temporary storage of containers in storage blocks*)* by using a rolling-horizon approach. For each planning horizon, the problem is decomposed into two levels: At the first level, they defined for each period the number of containers to be placed in each storage block. At the second level, they found out the number of containers stored in each block at each period associated with each vessel. The objective of the work of Chuqian et al. (2003) is to minimize the total distance to transport containers between their storage blocks and the vessel berthing locations.

There are other problems developed by some academic researchers to treat the *CSP* problem which is the storage of inbound containers or outbound containers. For the inbound containers problem type, Kap and Ki (2007) discussed a method of determining the optimal price schedule for storing inbound containers. Kap and Kang (2003) formulated a basic model as a mixed-integer linear program and they suggested two heuristics algorithms to solve the outbound containers problem. Recently, Lu and Zhiqiang (2010) treated the outbound containers problem. The objective of the problem is to minimize the rehandling operations by cranes in order to maintain the stability of the ship. The problem is decomposed into two stages. In the first stage, the numbers of locations in each yard block are determined by a mixed integer programming model. In the second stage, the exact storage location for each container is determined by a hybrid sequence stacking algorithm.

Mohammad et al. (2009) solved an extended *SSAP* in a container terminal by an efficient *GA*. The objective of the *SSAP* developed is to minimize the storage and retrieval time of containers. Changkyu and Junyong (2009) focused on the planar storage location assignment problem (*PSLAP*). The *PSLAP* can be defined as the assignment of containers in the storage area in order to minimize the number of obstructive moves. The authors proposed a GA to solve this variant of problem.

3. MATHEMATICAL MODEL

One of the major problems of a terminal is how to store containers in an optimal way. The goal of this work is to minimize the distance between vessel berthing location and the storage positions of containers and to determine an optimal storage strategy. In our case, the rehandling operations are not accepted, i.e., when a container is stored, it is not moved from its position until its departure time.

The Containers Stacking problem (*CSP*) can be formally described in the following way. Let $R = (K, P)$ be a graph where $K = \{1, ..., N\}$ is the containers set and $P = \{1, ..., N_p\}$ is the stacks set. The distance matrix cost is denoted by d_{pk}. It represents the shortest way between the position of the stack p and the ship of container k. The number of empty position of each stack is denoted by c_p.

We propose a new mathematical model that reflects reality and takes into account most of the constraints imposed by port authorities. This model treats the following hypotheses:

1. We don't mix on the same block and in the same period the loading and the unloading containers.
2. Before the beginning of each period, we know the state of the storage area. For each stack, we know: the number of container stored, the departure time of each container and the type of the stack (dimension of containers in the stack).
3. For each stack, containers must be stored in the decreasing order of their departure time.
4. Containers are stored by respecting the constraint of dimension compatibility. All containers stored in the same stack have the same dimension.

The *CSP* is formulated as a new and original mathematical programming model. This model is applied on each period in order to determine an optimal storage strategy based on the following assumptions.

3.1. Notation and Model Formulation

3.1.1. Parameters

i : is the index of the empty position in a stack p.

N *:* Represents the number of containers.

N_p *:* Represents the number of stacks.

M *:* is a big number

c_p *:* Represents the number of empty position for stack p.

r_p *:* Represents the type of stack p.

R_k *:* Represents the type of container k.

T_k *:* Represents the departure time of container k.

d_{pk} *:* Represents the shortest way between the position of stack p and the ship of container *k*.

$$t_p = \begin{cases} T_k & \textit{if} \text{ container } k \text{ is stored in the top of stack } p \\ M & \textit{if} \text{ the stack is empty} \end{cases}$$

3.2. Decisions Variables

$$\lambda_{pik} = \begin{cases} 1 & \textit{if} \text{ container } k \text{ is stored in a position } i \text{ of stack } p \\ 0 & \text{otherwise.} \end{cases}$$

The mathematical model proposed as follows:

$$Min \sum_{p=1}^{N_p} \sum_{i=1}^{c_p} \sum_{k=1}^{N} \lambda_{pik} d_{pk} \tag{1}$$

$$\sum_{p=1}^{N_p} \sum_{i=1}^{c_p} \lambda_{pik} = 1; \quad k = 1, \ldots, N \tag{2}$$

$$\sum_{k=1}^{N} \lambda_{ipk} \leq 1; \; p = 1, \ldots, N_p; \; i = 1, \ldots, c_p \tag{3}$$

$$\sum_{k=1}^{N} \lambda_{pik} \geq \sum_{k'=1}^{N} \lambda_{pi+1k}; p = 1, \ldots, N_p; \; i = 1, \ldots, c_p \tag{4}$$

$$\left(r_p - R_k\right) \lambda_{pik} = 0; \quad k = 1, \ldots, N; \; p = 1, \ldots, N_p; \; i = 1, \ldots, c_p \tag{5}$$

$$T_k \leq M(1 - \lambda_{pik}) + \lambda_{pik} t_p; k = 1, \ldots, N; \; p = 1, \ldots, N_p; \; i = 1, \ldots, c_p \tag{6}$$

$$M\left(1 - \lambda_{pik}\right) + \lambda_{pik} T_k \geq \lambda_{pi+1k'} T_{k'}; k = 1, \ldots, N; \; k' = 1, \ldots, N; p = 1, \ldots, N_p, ; i = 1, \ldots, c_p \tag{7}$$

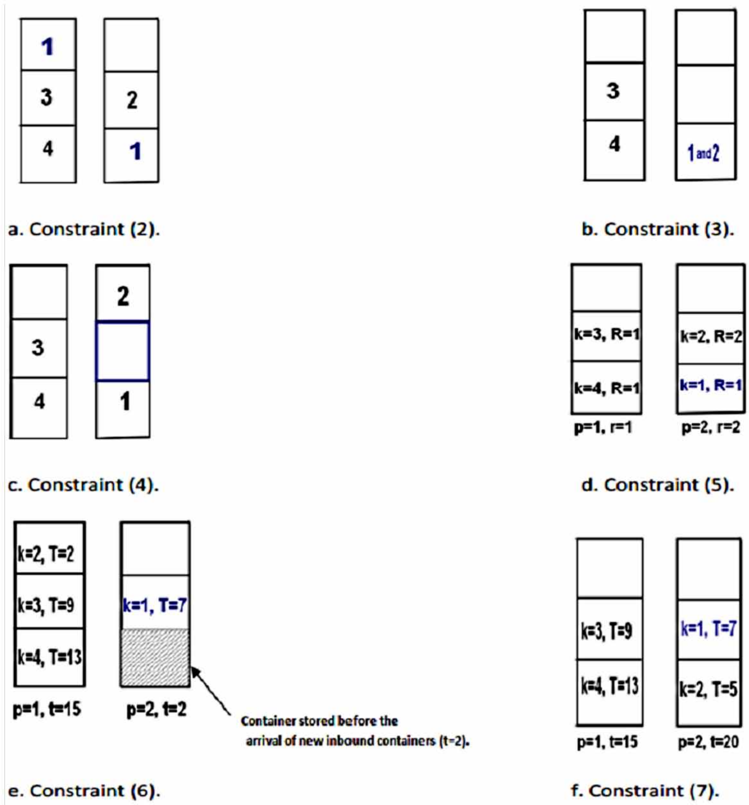
The objective function (1) of this model is to minimize the distance between the quay and the storage position of each container. Constraints (2) ensure that, each container is stored in one storage position. Constraints (3) indicate that each position in each stack receives only one container. Constraint (4) ensures that, an empty intermediate positions between containers stored in the same stack are not accepted. Constraints (5) specify that each container is stored in a stack with the same type. Constraints (6) and (7) apply that containers must be stored in a stack in the decreasing order of their departure time from the yard. Constraint (6) takes into account containers stored before the beginning of the new period. In constraint (7), Containers will be stored in the same stack; they must be stored in the decreasing order of their departure time.

Each constraint is more explained in Figure 1. Each example (from "a" to "f"), shows two stacks with a maximum height equal to three containers and four containers stored. Each figure represents a case that cannot be accepted by one of the constraint developed before.

Each point of "a" through "f" in Figure 1 is not validating by each constraint, for example:

1. constraint (2): In this representation, we note that the container 1 is stored in two stacks in the same time which is not possible if we respect the constraint 2 of the model.
2. constraint (3): In this representation, we note that the container 1 and 2 are stored in the same position in a stack which is not possible if we respect the constraint 3 of the model.
3. constraint (4): In this representation, we observe that the container 1 and 2 are stored in the same stack, leaving an empty position between the two containers which is not possible if we respect the constraint 4 of the mathematical model.

Figure 1. Explanation of constraints



4. SOLUTION METHODOLOGY

Ant algorithms have been inspired by colonies of real ants, which deposit a chemical substance (called pheromone) on the ground. This substance influences the choices they make: the larger amount of pheromone is on a particular path; the larger probability is that an ant selects the path. Artificial ants in ACO algorithms behave in similar way see Marco (1992). Traditionally, ACO have been applied to combinatorial optimization problems and they have achieved widespread success in solving different problems: scheduling, routing, assignment, see Marco and Thomas (2002), Euchi et al. (2015).

Generally, an ant solution is constructed by a series of probabilistic decision making where each decision represents a solution biased by adding a new component to the solution until a complete solution is derived. To construct the solution, the sequence of decisions can be viewed as a path of decision graph, the artificial ants look for the decision graph with the good solutions.

Our choice for the metaheuristic is based on the idea that the *ACO* has not been applied before to solve *CSP* or any similar problem in port. The preference of our solution procedure is based on the success of *ACO* to solve a wide range of challenging problems. Below the implementation of each part of the *ACO* algorithm to solve the *CSP* is described. The proposed solution is illustrated in three steps: Solution representation, solution construction and pheromone trail update. The details of each step of the *ACO* metaheuristic are detailed below:

4.1. Solution Representation

Solution representation of *CSP* problem is one of the key elements for valuable implementation of *ACO*. In our algorithm the solution representation is based on the assignments of containers to stacks. The solution is represented as a matrix with two rows as a string of length equal to the number of containers. For each column, the corresponding value represents the index number of container and the index number of stack, simultaneously.

To illustrate the encoding solution, we propose an example treating 5 containers and 3 stacks. Then the representation is described in Figure 2.

Container 1 is stored in Stack 2.
Container 2 is stored in Stack 2.
Container 3 is stored in Stack 1.
Container 4 is stored in Stack 3.
Container 5 is stored in Stack 1.

4.2. Solution Construction

The construction solution step consists of generating a set of solution $S_k = \left\{S_1, \ldots, S_{nbAnts}\right\}$ where all the containers are stored. In this section, we describe how to obtain an effective storage planner. This construction is based on the initial assignment, update of assignment and update of pheromone.

4.2.1. Initial Assignments

The different steps are more explained based on the following example: Table 1 and Table 2 illustrated an example of 5 containers which will be stored in 5 stacks.

We prepare a list of initial assignment that each container k is assigned to all stacks that can accommodate, taking into account all problem constraints. An assignment $\left(a_i\right)$ is represented as a table with one column and two rows as the first row represents the container and the second is the stack. Consider the previous example, the initial assignments of this example is shown in Table 3. For example, container 1 can be stored in stack 1 or stack 2.

For the proposed algorithm, each ant builds a solution. At the beginning each ant randomly selects an assignment among the initial assignments. After choosing an initial assignment all other assignments are chosen according to the following probability from within the set of *Candidates:*

$$P_{S_k}\left(a_i\right) = \frac{\left[\tau_{s_k}\left(a_i\right)\right]^{\alpha} \cdot \left[\eta_{s_k}\left(a_i\right)\right]^{\beta}}{\sum_{a_{j \in Condidats}} \left[\tau_{s_k}\left(a_i\right)\right]^{\alpha} \cdot \left[\eta_{s_k}\left(a_i\right)\right]^{\beta}}$$

This probability depends on:

Figure 2. Solution representation

Containers	1	2	3	4	5
Stacks	2	2	1	3	1

Table 1. Example of containers

k	1	2	3	4	5
R_k	1	2	1	3	3
T_k	8	10	7	11	6

Table 2. Example of stacks

p	1	2	3	4	5
t_p	10	11	∞	16	17
r_p	1	1	3	2	2
c_p	2	2	3	1	1

Table 3. Example of initial assignments

Assignments	1	2	3	4	5	6	7	8
Containers	K=1, R=1, T=8	K=3, R=1, T=7	K=1, R=1, T=8	K=3, R=1, T=7	K=4, R=3, T=11	K=5, R=3, T=6	K=2, R=2, T=10	K=2, R=2, T=10
Stacks	P=1, r=1, t=10	P=1, r=1, t=10	P=2, r=1, t=11	P=2, r=1, t=11	P=3, r=3, t=∞	P=3, r=3, t=∞	P=4, r=2, t=16	P=5, r=2, t=17

- The amount of pheromone: ants depose pheromone for each *assignment*$\left(a_i\right)$. The amount of pheromone for each a_i is $\tau_{s_k}\left(a_i\right)$. This amount represents the desirability to choose a_i on the construction of the solution, where S_k is the solution constructed by an ant k.
- The heuristic factor: is defined as the ratio based on the distance factor $\eta_{S_k}(a_i) = \dfrac{1}{d_{pk}}$
- α and β are two parameters that determine the relative importance of the two parameters of the probability presented before.

4.2.2. Update of Assignments

After each addition of an assignment to a solution S_k an update procedure is applied to the assignment list that includes:

1. Delete all assignments containing the container added to S_k .
2. Update the data of the stack of the assignment chosen. Date of stack becomes the date of the container and the capacity becomes the previous capacity-1.
3. If the capacity of the stack chosen becomes equal to 0 then we delete all the assignment of this stack.

Consider the previous example, we suppose that an ant choose the assignment that container 1 is stored in stack 1. Following the procedure of update assignments (see Table 4):

1. We delete all the assignment of container 1.
2. We update the data of stack 1. The date of stack 1 decreases from 10 to 8 and the capacity decreases also from 2 to 1.

4.3. Update of Pheromone

The implementation of ACO follows the $MAX - MIN$ Ant System which is proposed by Thomasand Holger (2000). The principle of $MAX - MIN$ Ant System that it impose lower $\left(\tau_{\min}\right)$ and upper $\left(\tau_{\max}\right)$ bounds on pheromone trails (with 0 < $\tau_{\min}$ < $\tau_{\max}$), and pheromone trails are set to $\tau_{\max}$ at the beginning of the search. Once each ant has constructed a solution, pheromone trails are updated according to:

• Evaporation rate: this is done by multiplying the quantity of pheromone by a pheromone persistence rate (1− ρ) such that $0 \le \rho \le 1$.

Table 4. Update of assignment

Assignments	~~1~~	2	~~3~~	4	5	6	7	8
Containers	~~K=1, R=1, T=8~~	K=3, R=1, T=7	~~K=1, R=1, T=8~~	K=3, R=1, T=7	K=4, R=3, T=11	K=5, R=3, T=6	K=2, R=2, T=10	K=2, R=2, T=10
Stacks	~~P=1, r=1, t=10~~	P=1, r=1, t=8	~~P=2, r=1, t=11~~	P=2, r=1, t=11	P=3, r=3, t=∞	P=3, r=3, t=∞	P=4, r=2, t=16	P=5, r=2, t=17

• The best ant of the cycle deposits pheromone. More precisely, let $S_k \in \left\{S_1,\dots,S_{nbAnts}\right\}$ is the solution constructed during the cycle by ant *k*, and S_{best} be the best solution built since the beginning of the run. The quantity of pheromone laid by ant k is inversely proportional to the gap of profit between S_k and S_{best} i.e., it is equal to $\dfrac{1}{(1+S_k-S_{brst})}$.

The pseudo code of the proposed hybrid metaheuristic is detailed in Algorithm 1:

5. EXPERIMENTAL RESULTS

In this section we present the performance of our *ACO* algorithm over the test problem randomly generated based on real-life terminal operations presented in the paper of Moussi et al. (2012). The *ACO* implementation is compared to *ILOG CPLEX* with small scales problems and with Hybrid Genetic Simulated Annealing Algorithm (*HGSAA*) of Moussi et al. (2012).

5.1. Benchmarks Description and Implementation

We consider two sets of instances to evaluate the performance of *ACO* metaheuristics problems. The first set is composed of the thirty-one small size instances compared with *ILOG CPLEX* and *HGSAA*. The second set is composed of the twenty-one hard instances compared with *HGSAA* metaheuristics.

Algorithm 1. ACO algorithm

begin
Initialize the pheromone trails $\tau_{\max}$ and the initial assignments.
Repeat
For each ant **do**
Solution construction using the pheromone trail;
Choose randomly the first assignment $a_1 \in 1 \dots n$
$S_k \leftarrow \left\{a_1\right\}$
Update list of assignment
Candidates $\leftarrow \left\{a_i \in 1 \dots n\right\}$
While Candidates ≠ Ø **do**
Choose $a_i \in$ Candidates with probability

$$P_{S_k}(a_i) = \frac{\left[\tau_{s_k}(a_i)\right]^{\alpha} \cdot \left[\eta_{s_k}(a_i)\right]^{\beta}}{\sum_{a_{j \in Condidats}} \left[\tau_{s_k}(a_i)\right]^{\alpha} \cdot \left[\eta_{s_k}(a_i)\right]^{\beta}}$$

$S_k \leftarrow S_k \cup \left\{a_i\right\}$
Update list of *assignment*
End while
Update the pheromone trails $\left\{S_1,\dots,S_{nbAnts}\right\}$
If a pheromone trail is lower than $\tau_{\min}$ **then** set it to $\tau_{\min}$
If a pheromone trail is greater than $\tau_{\max}$ **then** set it to $\tau_{\max}$
Until *Itermax*
end

In Tables 5 and 6, we present the description of the thirty-one small instances and the twenty one hard instances. For each instances, N_p = number of stacks, N = number of containers, and P_d = percentage of empty position in the terminal which is calculated by: $\frac{\sum of\ empty\ position}{3 \times N_p} \times 100$.

Table 5. Specifications of thirty-one benchmark problems

Instances	N_p	N	P_d
Instance 1	20	10	80%
Instance 2	35	25	67.61%
Instance 3	40	20	68,33%
Instance 4	40	25	73.33%
Instance 5	45	30	57.03%
Instance 6	45	35	66.66%
Instance 7	50	30	60%
Instance 8	50	35	66.66%
Instance 9	50	40	71.33%
Instance 10	55	40	66.66%
Instance 11	60	40	65%
Instance 12	60	45	64.44%
Instance 13	65	40	64.61%
Instance 14	70	45	67.14%
Instance 15	70	50	67.14%
Instance 16	70	55	73.33%
Instance 17	70	60	61.42%
Instance 18	75	60	67.11%
Instance 19	80	60	65%
Instance 20	80	65	68,33%
Instance 21	85	60	68,62%
Instance 22	90	65	69.62%
Instance 23	90	70	64.44%
Instance 24	95	70	70.17%
Instance 25	95	75	66.31%
Instance 26	95	80	61.4%
Instance 27	100	80	64%
Instance 28	100	85	63%
Instance 29	105	95	69.2%
Instance 30	110	95	66.96%
Instance 31	110	90	65.75%

Table 6. Specifications of twenty-one benchmark problems

Instances	N_p	N	P_d
Instance 1	300	100	64.22%
Instance 2	300	200	66.66%
Instance 3	300	300	66.66%
Instance 4	400	100	67.08%
Instance 5	400	200	66.5%
Instance 6	400	300	50.16%
Instance 7	400	400	66.33%
Instance 8	400	500	66.33%
Instance 9	500	100	66.53%
Instance 10	500	200	67.13%
Instance 11	500	300	65.93%
Instance 12	500	400	67.2%
Instance 13	600	100	66.66%
Instance 14	600	200	66.72%
Instance 15	600	300	67.94%
Instance 16	600	400	67.77%
Instance 17	700	100	67.42%
Instance 18	700	200	67.52%
Instance 19	700	300	67.14%
Instance 20	700	400	66.61%
Instance 21	700	500	66.09%

A comprehensive set of tests problems randomly generated based on real-life terminal operations. For each container and each stack, we have three types [20 feet, 40 feet, 45 feet], a departure time is uniformly distributed in [10, 80] for the small instances, [200, 1000] for the hard instances for each container and we calculate the distance between its ship and the different stacks. For each stack, the capacity is fixed from [1, 2, 3] and a departure time of the container situated on the top of each stack is uniformly distributed in [40, 100] for the small instances, [500, 2000] for the hard instances else if the capacity of the stack is equal to 3, then, its departure time is equal to 10000.

To authenticate the performance of the proposed metaheuristic, a number of experiments are set up. Our algorithm employs a set of parameters. In Table 7 we propose the parameters of the *ACO* algorithm to solve the *CSP*. To run the algorithm, it is necessary that the *ACO* metaheuristic provides work to set the parameters as follows:

This comparison is carried out in order to verify the quality of the *ACO* proposed. For a given instance, the results obtained vary with the values of *ACO* parameters.

The algorithm described here has been implemented in Scilab. Experiments are performed on a *PC* including two Intel® Xeon ® T3500, 2.67 GHz processors and 4 GO RAM.

Table 7. Parameters calibration of our experimental environment

Parameter type	Description	Value
Nbants	number of ants	10
$\alpha = 0.3$	Coefficient of pheromone quantity	0.3
$\beta = 0.2$	Coefficient of visibility	0.2
$\rho = 0.1$	Evaporation rate	0.1
$\tau_{\min}$	Lower bounds on pheromone trails	1
$\tau_{\max}$	Upper bounds on pheromone trails	10
Itermax	Number of iteration	20

5.2. Evaluation Method

In order to establish the effectiveness of our proposed approach we first compare the results obtained with *ILOG CPLEX* then the results produced by our algorithms have been compared with those produced by the *HGSAA* algorithm of Moussi et al. (2012). We run the algorithm to find the best results within a maximum number of 10 runs.

We begin the presentation of the results by examining, in Table 8, the efficiency of the procedure of *ACO*. This table shows the efficiency of the *ACO* compared with exact methods and the *HGSAA*. The average percentage deviation of *ACO*/*CPLEX* is equal to 1.23%. The percentage deviation is calculated by this formula: $\frac{\text{metaheuristics } \ result - optimum}{optimum} \times 100$. To have an idea of the efficiency of the *ACO* and *HGSAA*, at the last line, we calculate the average of percentage deviation which is defined as: $\frac{\sum percentage\ deviation}{number\ of\ ins\text{tan}ces}$.

In Table 9 we describe the efficiency of the *ACO* algorithm over other metaheuristics presented in the literature. It gives the comparison results between the *ACO* and the other methods proposed by Moussi et al. (2012). The results prove the strong performance of *ACO* procedure. Over the 21 instances, all best solutions were produced with our algorithm.

We clearly see that *ACO* gives the best results for all the instances generated. Moreover, the average percentage deviation decreases from 11.24% with the HGSAA to 1.23% with *ACO*. We compare also *ACO* and *HGSAA* between each other to solve some hard instances. However, it is clear that for hard instances *ACO* algorithm achieves better solution compared with *HGSAA*.

6. CASE STUDY

This work was supported by Laser Data Technology Terminal Company (LDTT) and Le Havre port in France by treating TERMINAUX DE NORMANDIE. It is the main containers operator in LE HAVRE on 2 terminals: Normandie Terminal and Ocean terminal. Figure 3 represent the picture of the Normandie Terminal in Le Havre port. It is also involved in conventional vessels at public berths and as operational subcontractor in the following activities:

Table 8. Computational results for Small scale problem

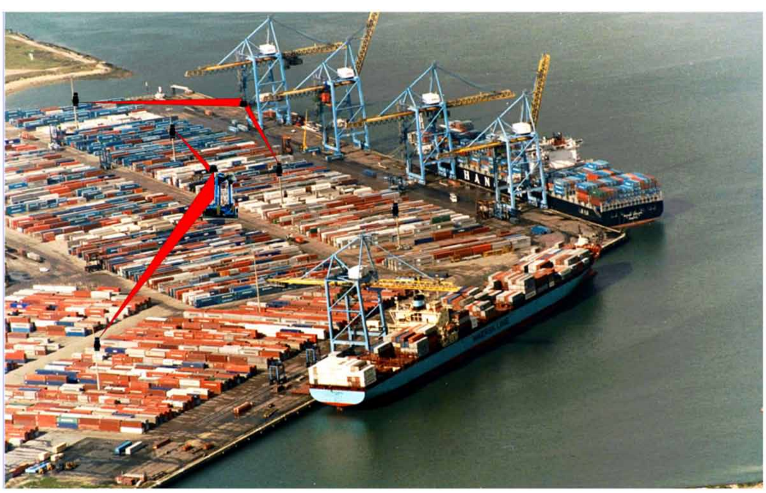
Instances	ILOG CPLEX[1]	ACO[2]	percentage deviation[3]	HGSAA[4]	percentage deviation[5]
Instance 1	4150	4150	**0,00%**	4150	**0,00%**
Instance 2	15350	15600	1.62%	16450	7,16%
Instance 3	10500	10500	**0,00%**	10850	3,33%
Instance 4	17650	17650	**0,00%**	18300	3,68%
Instance 5	22750	23000	1.09%	23550	3,15%
Instance 6	21250	23500	10.58%	24300	14,35%
Instance 7	21700	21700	**0,00%**	22200	2,30%
Instance 8	31500	31650	0.47%	32700	3,80%
Instance 9	27750	28250	1.8%	31150	12,25%
Instance 10	31100	31700	1.92%	33800	8,68%
Instance 11	32200	32200	**0,00%**	36050	11,95%
Instance 12	43250	43400	0.34%	45450	5,08%
Instance 13	33850	34650	2.63%	38050	12,40%
Instance 14	44200	44200	**0,00%**	48650	10,06%
Instance 15	53150	53400	0.47%	58400	9,87%
Instance 16	56050	56750	1.24%	63600	13,47%
Instance 17	72850	73250	0.54%	78950	8,37%
Instance 18	66200	66800	0.9%	74000	11,68%
Instance 19	60850	61500	1.06%	71950	18,24%
Instance 20	84400	84850	5.33%	93750	11,07%
Instance 21	60950	61150	0.32%	69920	14,71%
Instance 22	64400	65800	2.17%	77750	20,72%
Instance 23	87150	87450	0.34%	98750	13,31%
Instance 24	78300	80250	2.49%	96100	22,73%
Instance 25	91750	92400	0.7%	111300	21,30%
Instance 26	112000	112000	**0,00%**	125450	12,00%
Instance 27	122600	122600	**0,00%**	140200	14,35%
Instance 28	111300	111850	0.49%	128250	15.22%
Instance 29	132750	133450	0.52%	160500	15,90%
Instance 30	145600	145800	0.13%	162650	11,71%
Instance 31	124100	125700	1.28%	143650	15,74%
The average percentage deviation			1.23%		11,24%

[1] optimal results given by ILOG CPLEX.
[2] results given by our Ant Colony Optimization.
[3] percentage deviation between ACO and optimal solution.
[4] results given by Hybrid Genetic Simulated Annealing Algorithm HGSAA (Moussi et al. (2012)).
[5] percentage deviation between HGSAA and optimal solution.

Table 9. A comparison of ACO, HGSAA for hard scale problem

Instances	HGSAA	ACO
Instance 1	340750	156750
Instance 2	857750	581400
Instance 3	1469100	1253200
Instance 4	420950	157100
Instance 5	1059800	583650
Instance 6	1754050	1148450
Instance 7	2635000	2163750
Instance 8	3654300	3285750
Instance 9	515100	167600
Instance 10	1275900	538850
Instance 11	2109550	1244450
Instance 12	3039200	2153300
Instance 13	569750	155750
Instance 14	1451450	558300
Instance 15	2490050	1242000
Instance 16	3474850	2097100
Instance 17	653650	159000
Instance 18	1759800	167200
Instance 19	2817600	1218250
Instance 20	3990550	2068600
Instance 21	5314600	3281900

Figure 3. Terminal of Normandy



- General cargo
- Handling of cars carriers

The objective of this project is to ameliorate the port performance by treating the Normandy Terminal. The resolution method is based on:

- Port restrictions represent physical and logical requirements,
- Practical problem which is gained by visiting Le Havre port and communicating with the industry (LDTT).

The description of the general data of the Normandy terminal is presented in Table 10 as follows:

Containers are stored in the storage area by forming a number of stacks. Port authorities set stacks height based on the handling equipment used. The stacks of containers are arranged in rows aligned, called sides. A set of sides form a block in the storage area (see Figure 4). Containers must be picked up and dropped off at the right times, corresponding to the arrival and departure times of their transport modes.

At Le Havre port the planning of containers storage is prepared manually based mainly on the experience of its workers, its managers and their local expertise. The strong growth in the number of containers through the terminal has led to an increased need for efficient methods to manage the various handling operations. This will improve the handling of containers and strengthen the competitiveness of the port. The problem of storage containers is a bottleneck and is a very complex problem.

The main objective of this work is to address the problem of storage containers through new mathematical formulations and efficient algorithms to achieve two main goals: to optimize the transfer of containers minimizing the distances between vessels and the programmed positions of the various storage containers and ensure a storage plan which takes into account many constraints on the terminal.

To validate our models and verify our results, we developed a simulation platform. This platform reproduces the actual dimensions of the terminal of Normandy Quai Bougainville, Grand Port Maritime du Havre in France (Figure 4). We have proposed a different storage containers missions and we implemented a 2D handling operations.

Table 10. Data information of Normandy terminal

Safe working load under spreader Telescopic 20'/40'/45'	40 T
Safe working load under heavy, lift beam	50 T
Outreach over vessel	47.5 meters
Back reach over apron	25.0 meters
Clearance under spreader	33.0 meters
Rail gauge	21.5 meters
Stacking area	400 000 square meters
Stacking capacity (3 heights)	15 500 TEU
Width of apron including hatch cover landing space	50 meters
Fendering system	DOLPHIN
Bollards (Strength/Spacing)	100 T/24 meters
Foot bridge for container inspection	Yes

Figure 4. Example of block in port

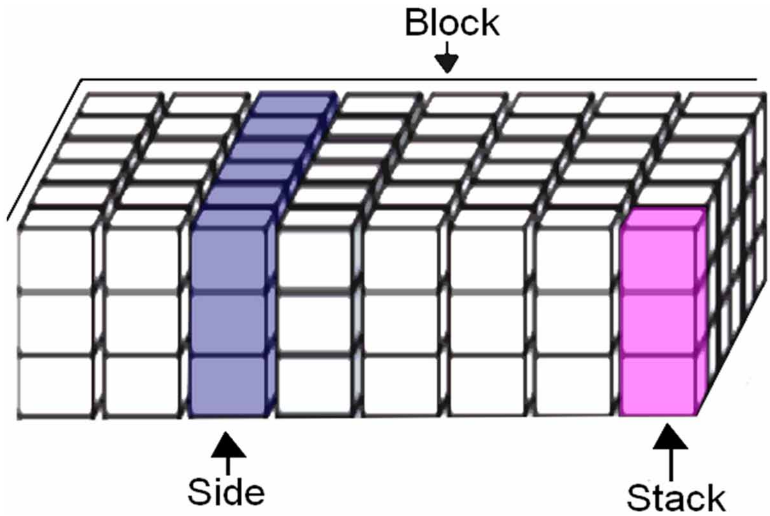


Figure 5 represents Quay Bougainville, Grand Port Maritime du Havre in France. In this terminal there are two quays and around thirteen storage blocks with a stacking capacity (3 heights) equal to 3600 stacks. In our simulator the heights of each stack are represented by different colors (Table 11).

In Figure 6 we represent ten real instances taken from the Le Havre port and resolved by our algorithm and the other methods proposed by Moussi et al. (2012). The results prove the strong performance of *ACO* procedure.

In conclusion, we can say that the results of our approaches are encouraging, and they can open a wide field of applications on other port problems not covered in this work.

7. CONCLUSION

In this paper, we propose a new model to solve *CSP* which is designed to minimize the unloading time of containers and to determine an optimal storage strategy. The originality of this model is the presentation of several physical and logical real constraints of port.

The *CSP*, very important in port logistics, is an NP-Hard problem. This requires the use of meta-heuristics methods to find an approximated optimal solution for real instances where it is difficult and sometimes impossible, to determine the optimal solution for large size problems by exact methods.

The treated model was solved in previous works by hybrid algorithm (HGSAA) (Moussi et al. 2012), but the inconvenient of these algorithms is that their results are far from the optimal values with an average percentage deviation equal to 11.24% for the HGSAA. The idea is to improve the results provided by HGSAA. For this motivation, we propose a new approach based on an implementation with ACO. By comparing the results obtained by ACO with the exact results supplied by ILOG CPLEX on the problems of small dimensions, we noticed that, the results are more effective than those supplied by HGSAA and we obtain a deviation equal to 1.23%. Furthermore, applied to the real problems, of large sizes, we notice that ACO is faster and more effective than HGSAA. The results obtained by this new approach are encouraging and improve in a considerable way the previous results obtained until now.

The perspective of this work is to improve the proposed model and to use the obtained results to solve another problem which is the transfer of containers in port container terminal.

Figure 5. Terminal of Normandy Quai Bougainville, Grand Port Maritime du Havre in France

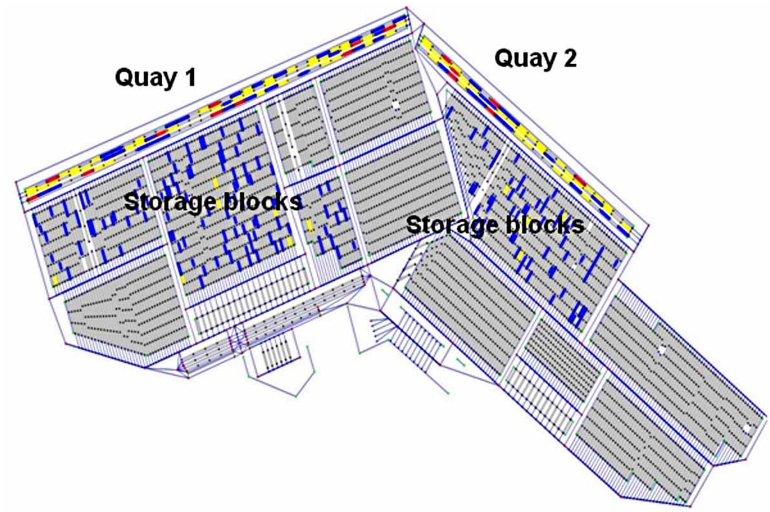
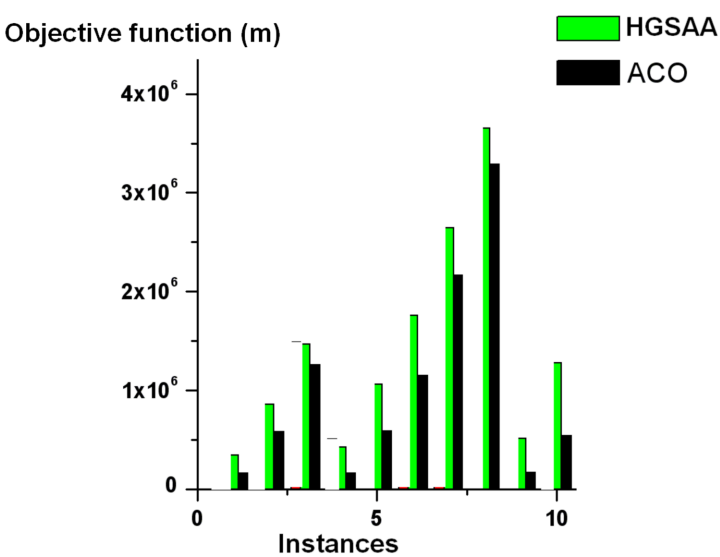


Table 11. Data information of the simulator

Stacks heights	Color in our simulator
No stacks	White
Stacks not used	Gray
One container	Yellow
Two containers	Blue
Three containers	Red

Figure 6. Terminal of Normandy Quai Bougainville, Grand Port Maritime du Havre in France



REFERENCES

Changkyu, P., & Junyong, S. (2009). Mathematical modeling and solving procedure of the planar storage location assignment problem. *Computers & Industrial Engineering*, *57*(3), 1062–1071. doi:10.1016/j.cie.2009.04.010

Chuqian, Z., Jiyin, L., Yat-wah, W., & Katta, M. (2003). Storage space allocation in container terminals. *Transportation Research*, *37*(10), 883–903. doi:10.1016/S0191-2615(02)00089-9

Dayama, N. R., Krishnamoorthy, M., Ernst, A., Narayanan, V., & Rangaraj, N. (2014). Approaches for solving the container stacking problem with route distance minimization and stack rearrangement considerations. *Computers & Operations Research*, *52*, 68–83. doi:10.1016/j.cor.2014.06.018

Erhan, K., & Peter, P. (2006). Mathematical modeling of container transfers and storage locations at seaport terminals. *OR-Spektrum*, *28*(4), 519–537. doi:10.1007/s00291-006-0048-1

Euchi, J., Yassine, A., & Chabchoub, H. (2015). The dynamic vehicle routing problem: Solution with hybrid metaheuristic approach. *Swarm and Evolutionary Computation*, *21*, 41–53. doi:10.1016/j.swevo.2014.12.003

Kap, H. K., & Hong, B. K. (1999). Segregating space allocation models for container inventories inport container terminals. *International Journal of Production Economics*, *59*(1-3), 415–423. doi:10.1016/S0925-5273(98)00028-0

Kap, H. K., & Kang, T. P. (2003). A note on a dynamic space-allocation method for outbound containers. *European Journal of Operational Research*, *148*(1), 92–101. doi:10.1016/S0377-2217(02)00333-8

Kap, H. K., & Ki, Y. K. (2007). Optimal price schedules for storage of inbound containers. *Transportation Research*, *41*(8), 892–905. doi:10.1016/j.trb.2007.02.001

Kap, H. K., Young, M. P., & Kwang-Ryul, R. (2000). Deriving decision rules to locate export containers in container yards. *European Journal of Operational Research*, *124*(1), 89–101. doi:10.1016/S0377-2217(99)00116-2

Lu, C., & Zhiqiang, L. (2010). The storage location assignment problem for outbound containers in a maritime terminal. *International Journal of Production Economics*, *48*, 991–1011.

Marco, D. (1992). Optimization, learning and natural algorithms [PhD thesis]. Politecnico diMilano, Italy.

Marco, D., & Thomas, S. (2002). *The Ant Colony Optimization Metaheuristic: algorithm, applications and advances. Handbook of metaheuristics* (pp. 251–285). Kluwer Academic Publishers.

McDowell, E., Martin, G., Cho, D., & West, T. (1985). *A Study of Maritime Container Handling*. Oregon State University, Sea Grant College.

Mohammad, B., Nima, S., & Nikbakhsh, J. (2009). A genetic algorithm to solve the storage space allocation problem in a container terminal. *Computers & Industrial Engineering*, *56*(1), 44–52. doi:10.1016/j.cie.2008.03.012

Moussi, R., Euchi, J., Yassine, A., & Ndiaye, N. F. (2015). A hybrid ant colony and simulated annealing algorithm to solve the container stacking problem at seaport terminal. *International Journal of Operational Research*, *24*(4), 399–422. doi:10.1504/IJOR.2015.072723

Moussi R, Ndiaye F, & Yassine. A. (2012). Hybrid Genetic Simulated Annealing Algorithm (HGSAA) to Solve Storage Container Problem in Port. In *Intelligent Information and Data base Systems, LNCS* (Vol. 7197, pp. 301-310).

Ndiaye, N. F., Yassine, A., & Diarrassouba, I. (2014). A hybrid ant colony and genetic algorithm to solve the container stacking problem at seaport terminal. *Proceedings of the2014International Conference on Advanced Logistics and Transport*. doi:10.1109/ICAdLT.2014.6866319

Peter, P., & Erhan, K. (2001). An approach to determine storage locations of containers at seaport terminals. *Computers & Operations Research*, *28*(10), 983–995. doi:10.1016/S0305-0548(00)00020-4

Sculli, D., & Hui, C. F. (1988). Three dimensional stacking of containers. *Omega*, *16*(6), 585–59. doi:10.1016/0305-0483(88)90032-1

Stahlbock, R., & Voß, S. (2008). Operations research at container terminals: A literature update. *OR-Spektrum*, *30*(1), 1–52. doi:10.1007/s00291-007-0100-9

Steenken, D., Voß, S., & Stahlbock, R. (2004). Container terminal operation and operations research-a classification and literature review. *OR-Spektrum*, *26*, 3–49. doi:10.1007/s00291-003-0157-z

Taleb-Ibrahimi, M., de Castilho, B., & Daganzo, C. F. (1993). Storage space vs handling work in container terminals. *Transportation Research Part B: Methodological*, *27*(1), 13–32. doi:10.1016/0191-2615(93)90009-Y

Thomas, S., & Holger, H. (2000). MAX-MIN Ant System. *Future Generation Computer Systems*, *16*(8), 889–914. doi:10.1016/S0167-739X(00)00043-1

Jalel Euchi is a Doctor at the Department of Quantitative Methods, University of Sfax, Tunisia also he prepared his thesis in the form of co-supervised at the University of Havre in France. He received his Bachelor of Operational Research degree and Master degree in Operational Research and Production Management from the Department of Quantitative Methods of the University of Sfax, Tunisia. His primary research interests are in complex vehicle routing problems, heuristics and meta-heuristics algorithms to solve the NP-Hard problems, computational operations research, logistics research and supply chains. He participated to different international conferences and he was the corresponding author in several published papers.

Riadh Moussi is a Computer Assistant Professor at the Faculty of Science and Technology of Sidi Bouzid. He has completed his PhD in 2012 at the Department of Applied mathematics, University of Le Havre, France. He holds a Master in Mathematics and Computer Science Applied to Complex Systems at the University of Le Havre, France and a fourth year on Computer Science Applied to the Management of the University of Sfax, Tunisia. Currently, he has a Post-Doctoral in Laboratory of Computer Signals and Image, University of the Littoral Opal Coast, France. His primary research interests are in port problems, heuristics and meta-heuristics algorithms to solve the NP-hard problems, computational operations research, logistics research and supply chains. He participated to different international conferences and he was the corresponding author in several published papers.

Ndèye Fatma Ndiaye is graduated with a mathematics PhD of the Laboratory of Le Havre University. She holds a Master in Mathematics and Computer Science Applied to Complex Systems at the University of Le Havre in 2011.

Adnan Yassine is a Full Professor at the University of Le Havre to Normandy in France and a member of the Laboratory of Applied Mathematics of Le Havre (LMAH). He teaches the optimisation and the logistics to the Superior Institute of Logistic Studies (ISEL). He is the author and co-author of numerous international publications in numerical and combinatorial optimisation among which some concerning the scheduling and logistics problems. He obtained his PhD of Applied Mathematics at the University of Grenoble, France in 1989 and Habilitation to Supervise Research (HDR) to the University of Nancy, France in 1998.