



HAL
open science

A rail-road PI-hub allocation problem: Active and reactive approaches

Faiza Walha, Abdelghani Bekrar, Sondès Chaabane, Taicir Moalla Loukil

► To cite this version:

Faiza Walha, Abdelghani Bekrar, Sondès Chaabane, Taicir Moalla Loukil. A rail-road PI-hub allocation problem: Active and reactive approaches. *Computers in Industry*, 2016, 81, pp.138-151. 10.1016/j.compind.2016.04.007 . hal-03427034

HAL Id: hal-03427034

<https://uphf.hal.science/hal-03427034v1>

Submitted on 18 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A rail-road PI-hub allocation problem: Active and reactive approaches

Faiza Walha^a, Abdelghani Bekrar^{b,*}, Sondes Chaabane^b, Taicir Moalla Loukil^c

^a LOGIQ, University of Sfax, Sfax, Tunisia

^b LAMIH, UMR CNRS 8201, University of Valenciennes and Hainaut-Cambrésis, UVHC, Le Mont Houy, 59313 Valenciennes Cedex, France

^c Faculty of business administration, Tabuk University, Tabuk, Saudi Arabia

ABSTRACT

This research concerns an allocation problem in the context of the physical internet aimed at improving rail-road π -hub efficiency by optimizing the distance travelled by each container to the dock, as well as the number of trucks used. To achieve this, heuristic, metaheuristic and Multi-agent-based approaches are proposed. When given thesequence of all the containers in the train, the proposed heuristic approach can assign these containers to outbound doors. Then, the Simulating Annealing (SA) method improves this allocation by minimizing the distance travelled. In addition, a multi-agent system model is proposed to generate reactive solutions which take dynamic aspects into account.

The experimental results show that the proposed SA yields an improvement of about 2.42–7.67% in relation to the solution generated by the heuristic; it provides good results within a reasonable time. Conversely, the multi-agent-based approach provides good solutions in case of perturbations or unexpected events.

Keywords:

Physical internet
Rail-road allocation
Cross-docking
Heuristic
Simulated annealing
Multi-agent system

1. Introduction

The Physical Internet (PI, denoted π) is defined as “an open global logistics system founded on physical, digital and operational interconnectivity through encapsulation, interfaces and protocols” [1]. In this structure, goods are shipped in standard size containers in the same way as data packets in the Digital Internet, where networks are connected using standard packets of data under the TCP-IP protocol. In order to exploit the Physical Internet, Montreuil et al. [2] proposed three key types of physical elements: π -containers, π -movers (π -vehicles, π -carriers, π -conveyors and π -handlers) and π -nodes (π -transits, π -switches, π -bridges, π -sorters, **π -hubs**, π -composers, π -shops, π -bridges . . .).

The mission of the **π -hub** is to transfer π -containers from the incoming π -movers to the outgoing π -movers. Ballot et al. [3] developed a new, specific “road-rail π -hub” for the purpose of transferring containers from trucks to trains (“road \rightarrow rail”), and vice-versa (“rail \rightarrow road”), as well as from one train to another (“rail \rightarrow rail”). The road-rail π -hub is plagued by three main problems:

- The road \rightarrow rail problem: some π -containers must be transferred from π -trucks to railcars using the road-rail π -sorters.
- The rail \rightarrow rail problem: some π -containers must be transferred from railcars to other railcars using the rail-rail π -sorters.
- The rail \rightarrow road problem: some π -containers must be transferred from railcars to outgoing π -trucks using the rail-road π -sorters.

In this paper, the last problem (“rail \rightarrow road”) is considered. The main performance objective of the “rail \rightarrow road” zone is to minimize the number of trucks used and the distance travelled by each container to reach the docks. However, many specific constraints are considered: 1) The position of both containers and trucks in relation to the docks tends to change over time. 2) The position of the containers in relation to the docks is important when the objective is to minimize the number of containers moving through the routing zone. 3) The size of the containers placed on each truck should not exceed the capacity of the truck. 4) All containers put on a specific truck are heading for the same destination.

This paper is organized as follows. In Section 2, a literature review is presented. Section 3 describes the problem considered. Proposed approaches to solve the allocation problem are detailed in Section 4. Computational results and experiments are presented in Section 5. Finally, a conclusion is drawn and future prospects are addressed.

* Corresponding author.

E-mail addresses: walha.faiza@gmail.com (F. Walha),

Notations

| | |
|-----------|---|
| N | Number of containers. |
| M | Number of dock positions |
| D | Number of destinations |
| P | Number of docks |
| i | Indices of containers, $i = 1, \dots, N$ |
| p | Indices of docks, $p = 1, \dots, P$ |
| l_{c_i} | Length of container i , $i = 1, \dots, N$. |
| T | Number of periods in planning horizon |
| K | Capacity/length of trucks |
| d_{ip} | Distance travelled by container i , to reach dock p . $i = 1, \dots, N$, $p = 1, \dots, P$ |
| cd | Total distance travelled defined as the sum of all distances travelled by all the containers to reach the docks |

2. Literature review

2.1. The cross-docking platform

Cross-docking is a distribution system where freight is received and prepared in order to be transferred to another location, typically by trucks, shipping containers or rail. Most of the existing research is on truck-to-truck applications [4]. In these cross-docking hubs, freight is shipped from inbound trucks to outbound trucks on the same day, or overnight without storage. The cross-docking problem is classified into three levels: strategic, tactical and operational (see Refs. [5] and [6]). Different problems are studied: cross-dock location, layout design, vehicle routing, inner transport scheduling, truck scheduling, and dock-door assignment.

In the dock-door assignment problem, the purpose is to assign destinations to outbound dock-doors of the cross-dock, with the

aim of minimizing the distance travelled for the material-handling equipment. When the number of trailers exceeds the number of docks available, the first are parked until at least one dock becomes available. The assignment can be of medium-term, short-term, or a combination of both [7]. In the medium-term, each door is assigned to an input or output destination for a specific period of time (usually 6 months). In the short-term model, each door is assigned to an input or output destination based on the current flow of goods.

In Ref. [8], the authors compared the existing literature review with industry practices. They propose a different classification of cross-docking. For the dock-door assignment problem, the authors classify the existing papers as “truck-to-door assignment problems”. McWilliams [9] developed a dynamic algorithm to solve the problem of load balancing in the cross-docking context. This consists in scheduling a set of incoming trucks, with a heterogeneous set of packets, to a set of unloading doors. The proposed method can be applied to manual and automated systems. In automated systems, freight is moved using conveyors, as with our problem, and the objective is to minimize the total transfer time. Tsui and Chang [10] formulate an assignment issue as a bi-linear programming problem, where the goal is to minimize the distance travelled by the forklifts. The same authors [11] propose a branch-and-bound algorithm to solve the dock-door assignment problem. Oh et al. [12] solve the assignment problem in a mail distribution centre. Miao et al. [13] develop a similar heuristic search and an adapted genetic algorithm to solve truck scheduling problems so as to minimize the operational cost of shipments. In Refs. [14–16] some heuristics and metaheuristics are proposed to minimize the total processing time. Golias et al. [17] developed a memetic algorithm where they demonstrate the advantages of scheduling inbound and outbound trucks simultaneously.

2.2. The classical road-rail hub

The road-rail terminal is a special transshipment node where gantry cranes tranship containers from trains to trucks, and vice versa. These containers are collected, rearranged, unloaded,

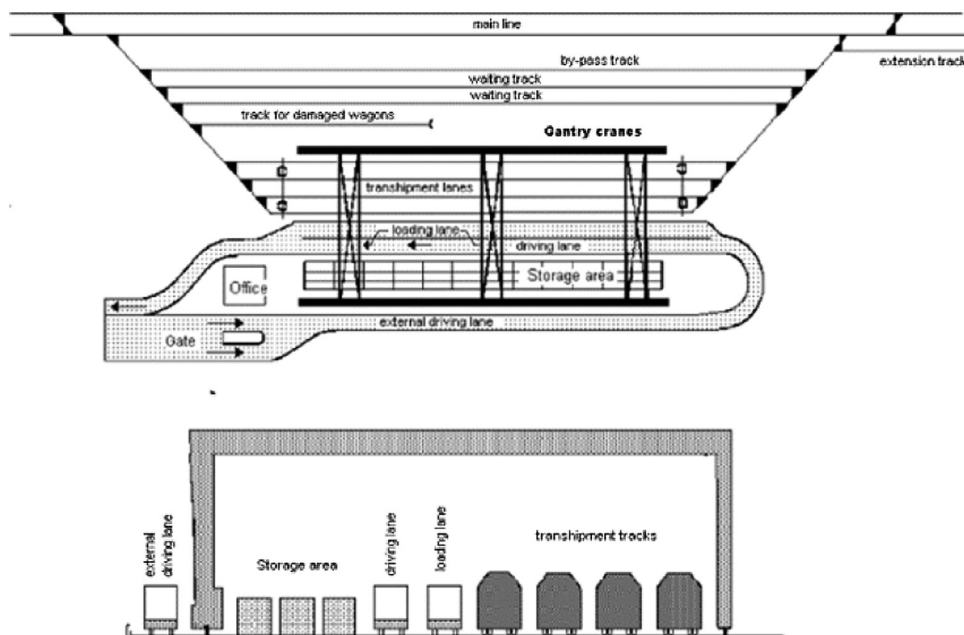


Fig. 1. Layout and cross-section of the rail-road terminal equipped with three gantry cranes [19].

intermediately stored, loaded, and/or picked up. Boysen et al. [18] grouped these terminals into three generations:

- First generation: in traditional terminals, the railcars must be detached and then pushed over a ramp when the train arrives at the yard.
- Second generation: in modern terminals, the railcars should not be detached. Only the containers are transhipped from trains to trucks, and vice-versa, using the gantry cranes. In some terminals, there are also storage areas for stacking containers. These terminals are used as part of a hub-and-spoke network.
- Third generation: the layout is similar to that of second-generation terminals. However, a fully automated sorting system is used. The shuttle cars receive containers and move them to their target position. The gantry crane is used to pick up the containers and transfer them to their outbound trains. Most of these terminals are still in the design phase.

According to Ballis and Golias [19], a typical road-rail terminal (second and third generation) includes the following elements (Fig. 1):

- Rail sidings for train/wagon storage, marshalling and inspection purposes;
- Transshipment tracks (also termed loading tracks) for loading/unloading the trains;
- Storage or buffer lanes, for intermodal transport units;
- Loading and driving lanes for the trucks;
- Gates for internal road network.

Many decisions must be made carefully during the operational phase of the terminals. Boysen et al. [20] consider five of these essential issues which have a short-term impact (operational): 1) Deciding on the storage positions of containers handled by split moves. 2) Assigning each truck to a parking position. 3) Determining the positions of outbound containers on trains. 4) Assigning cranes to move containers. 5) Determining the sequence of container moves per crane.

According to this classification, only the fourth and the fifth issues address the “rail → road” problem. However, the goal of these issues is to assign the containers to the crane and find the sequence of moves per crane, using the storage area or not.

Furthermore, the positioning of the trucks is horizontal (there are no docks, but the truck is positioned parallel to the container).

By contrast, the road-rail π -hub design proposed by Ballot et al. [3] requires the use of stackers, robots or conveyors to transfer the containers from the train to the truck. In addition, the positioning of the trucks is vertical (see Fig. 3).

2.3. PI road-rail vs. cross-docking and classical road-rail

As introduced previously, this paper focuses on the “road-rail π -hub” context, and more precisely the “rail-road” problems. In order to minimize the distance travelled by each container to reach the docks, the objective involves assigning containers to trucks, while at the same time defining the destinations of the dock/trucks. The road-rail π -hub is considered as a novelty in logistics due to the recentness of the Physical Internet concept. Since there is a gap in the literature concerning the “road-rail” π -hub problem, in Table 1 we present some differences between the classical and the PI-hub “road-rail” problem.

As the design of the PI road-rail hub is new and different from the classical road-rail terminal (see Table 1), we found that the allocation problem proposed does not conform to the existing literature. So we turned to the nearest problem, which is the cross-docking problem, and more precisely, the dock-door assignment problem.

2.4. Multi-agent system for cross-docking and physical internet problems

In order to have a reactive solution that evolves better in a dynamic industrial context, Multi-agent system (MAS) based approaches are among those that have demonstrated their efficiency in various aspects of different supply-chain management problems.

For cross-docking problems and physical internet applications, only a few papers have used the approaches classified as Decentralized (Holonetic, MAS, etc.), in contrast to classical approaches (operational research-based, etc.). In their review of cross-docking applications, Van Belle et al. [5] consider Holonic manufacturing execution systems as a research opportunity to improve the applicability of cross-docking systems. They affirm

Table 1
Classical road-rail vs. PI road-rail vs. cross-docking terminal.

| | Cross-docking terminal | Classical road-rail terminal | Road-rail π -hub |
|--|---|---|---|
| Mission | Transfer containers from one means of transport to another. Trucks are commonly used as a means of transport. | Transfer containers from train to trucks, and vice versa, and from one train to another. | Transfer containers from train to trucks, and vice versa, and from one train to another. |
| Kinds of freight | All kinds of freight | All kinds of freight: boxes, shrink-wrapped pallets . . . | Modular and standard π -containers |
| Parking position of trucks | There are specified docks for each truck, and the positioning of the trucks is vertical. | There is no specific dock for the truck, and the positioning of the trucks is horizontal. | There are specific docks for each truck, and the positioning of trucks is vertical. |
| Internal means to transport containers | Conveyors or trolleys | Gantry cranes | Conveyors, robots or stackers |
| Storage area | Containers are often transferred directly to the trucks, and sometimes they are transferred to the storage areas. | Used frequently. | Containers are often transferred directly to the trucks, and sometimes, they are transferred to the storage area. |
| Users | Restricted to suppliers and/or clients of a specific company and its partners/suppliers. | Restricted to suppliers and/or clients of a specific company and its partners/suppliers. | Conceived by default to be open to any π -certified users. |
| Sorting area | The containers could be temporarily sorted en route to the docks. | Marshalling yards. | The π -containers can be momentarily sorted on their way to the docks or to the train. |

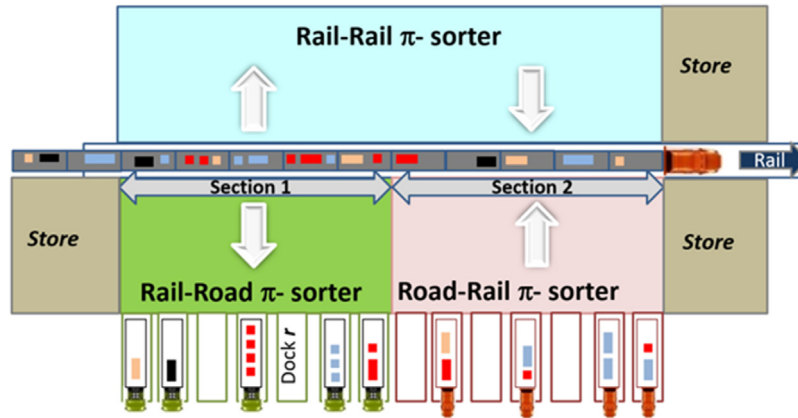


Fig. 2. Road-rail PI hub problems (Ballot et al. [3]).

that these approaches will be more robust and dynamic to tackle logistics problems in a dynamic context.

In Ref. [20], the authors introduced the supply web concept, and propose a Simulation Platform as a tool for Decision Systems that support decision making in the context of the supply web. The Simulator offers advanced agent behaviour to reproduce environments which are as close to reality as possible.

Klumpp and Sandhaus [21] proposed an agent-based decision support framework for optimizing rescheduling alternatives at the cross-dock.

In his thesis [22], Van Belle proposes applying the concepts of the Holonic Manufacturing Execution System in order to develop a Holonic Logistics Execution System (HLES). The author shows that the HLES concept is a valuable option for a cross-docking problem, based on cooperation with a vehicle-routing scheduling system and a truck-scheduling system.

In [23], Furtado simulates a transportation network, with the Physical Internet concept, and analyses its efficiency through performance indicators. The authors tried to simulate the whole network with the different “players” (customers, orders, hubs, trucks, routes, etc.). The simulation was conducted with the NetLogo software using the concept of multi-agent simulation. The authors tried to demonstrate the effectiveness of the consolidation; however, the physical internet concepts are not clearly exposed.

More recently, in Ref. [24], a cross-docking optimization case study involving a technical consumer products manufacturer is

presented. Using the hybrid simulation model, which consists of a discrete-event and an agent-based model, key cross-docking performance metrics for one of the major finished product unloading hubs were analyzed. Key parameters were then configured to optimize cross-docking performance.

As one can see in this short review, only a few papers were interested in the application of decentralized approaches for cross-docking problems, and even fewer in the physical internet. This can be explained by the novelty of these concepts.

The main contribution of this paper concerns two issues. First, an assignment problem is addressed where containers should be assigned to the “best” dock in a physical internet context. Second, a reactive approach based on a Multi-agent System is proposed to deal with perturbations or unexpected events.

In the next section, the problem considered is addressed and detailed.

3. Problem description

In their paper, Ballot et al. [3] proposed a functional design of the road-rail π -hub. This design ensures the scalability of the hub according to the amount of traffic. The proposed functional design can be implemented in many ways and with various technologies; e.g. stackers, robots or conveyors. The π -hub is very sensitive to many parameters: the number of containers to be loaded and

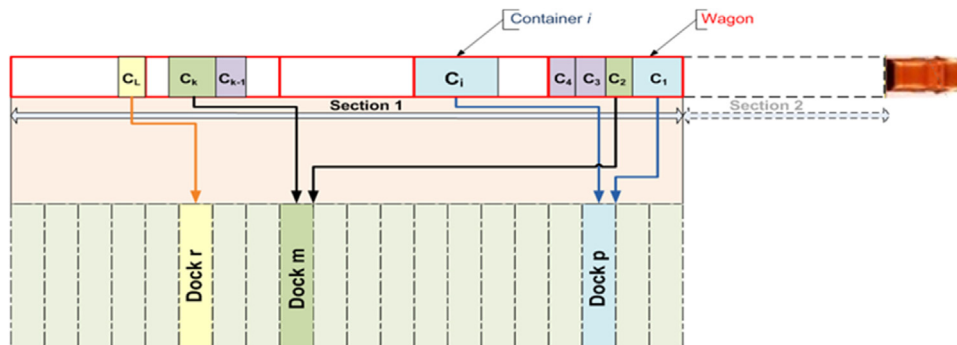


Fig. 3. Distance travelled by the container to reach a dock.

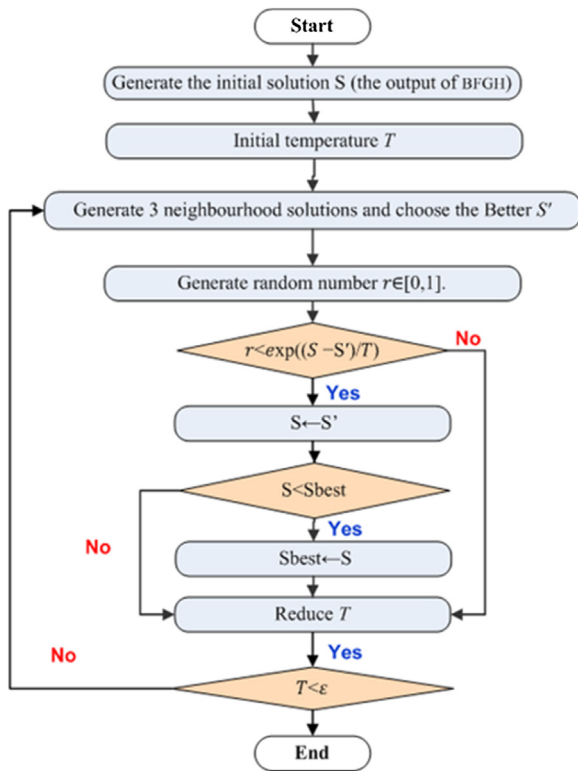


Fig. 4. The Simulated Annealing method.

unloaded on each train, the number of docks, the distance between the docks, the size of the railcars, etc.

The rail splits the π -sorters into two sections: the top section is devoted to the loading and unloading of containers from one train to another. The bottom section is composed of two other sections: the left section (Section 1) is used for unloading containers from railcars to trucks; the right section (Section 2) is used for loading containers from trucks to railcars. The distance between two trucks is 1 m or 1.25 m. Thus, each railcar can be processed by 5 or 6 trucks. We estimate that the number of trucks per section (dock) is about 30. The positioning of the trucks is vertical. The length of each truck is 13.2 m (Fig. 2).

The handling of the train is a two-fold operation: loading and unloading. These operations are repeated until the entire train is processed. The train moves forward one section, the previous railcars are moved to the second section of the π -hub, and new railcars take over the first section of the π -hub.

The train, with 30 railcars, comprises blocks of 5 railcars in the rail-road π -sorters section (see Fig. 3). So, when the unloading task is completed in the block, the train moves on, and another block of 5 railcars is ready to be unloaded. All full trucks leave the docks so empty trucks can be loaded.

Pach et al. [25] distinguished three types of problems in the road-rail π -hub:

- The trucks-to-docks assignment problem: this assignment entails assigning trucks to docks.
- The containers-to-docks/trucks assignment problem: the suitable assignment of π -containers unloaded from the train to the available trucks must be defined.
- The routing problem: the containers must be routed to their designated trucks.

This paper focuses on the second problem, which involves allocating each container to its truck (the capacity of the containers should not exceed that of the truck) and then assigning the docks to the correct destination (all the containers in the same truck have the same destination). The main objective is to minimize the distance travelled by each container to reach the docks, as well as the number of trucks used (Fig. 3). In the following section, input parameters, notations and assumptions are presented.

3.1. Input parameters and assumptions

As presented by Ballot et al. [3], the road-rail π -hub is composed of:

- The train comprising n railcars. It is split into two sections of equal size;
- The railcars (which are never detached), 18 m long (2 m between each railcar);
- The π -containers, characterized by different destinations, sizes (width and height are 2.4 m, and length is any of the following: 1.2 m, 2.4 m, 3.6 m, 4.8 m, 6 m and 12 m) and locations (the coordinates of the containers in the train);
- The π -conveyors, which are different from classical conveyors (with neither rollers nor belts), the π -containers simply clip themselves to the π -conveyors so they can be towed;
- The outbound and inbound π -bridge docks;
- The π -buffer area, where trucks wait to be assigned.
- Trains, which arrive on a scheduled basis at regular intervals. Therefore, a single train is treated at a time.
- As we are seeking maximum capacity, we assume that a train is full when it enters and when it leaves the hub.
- The railcar is 18 m in length and can be used by any combination of containers within the defined set (according to this, a full railcar carries between 2 and 15 π -containers). Since the height and the width of the containers and the trucks are the same, we only consider the length.
- A fleet of trucks with a fixed capacity K for each is available. A truck is affected to a dock to serve a destination. When the truck is full or a fixed time is reached, it leaves the dock.

4. Proposed approaches for the PI-hub allocation problem

In this paper, we propose a new heuristic called the “Best Fit Grouping Heuristic (BFGH)”. In this heuristic, we propose a method to allocate containers to trucks, taking into account two goals: minimize the number of trucks used and the distance travelled by the containers from the train to reach the dock. In a second stage, the BFGH is improved with the SA metaheuristic. Finally, we

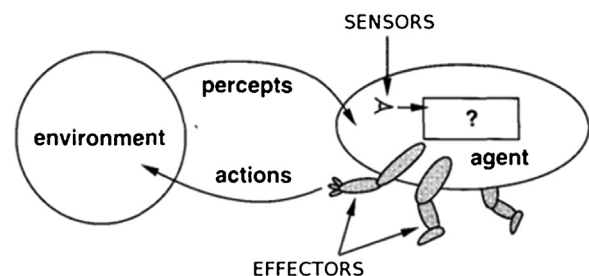


Fig. 5. Agents interact with environments through sensors and effectors [31].

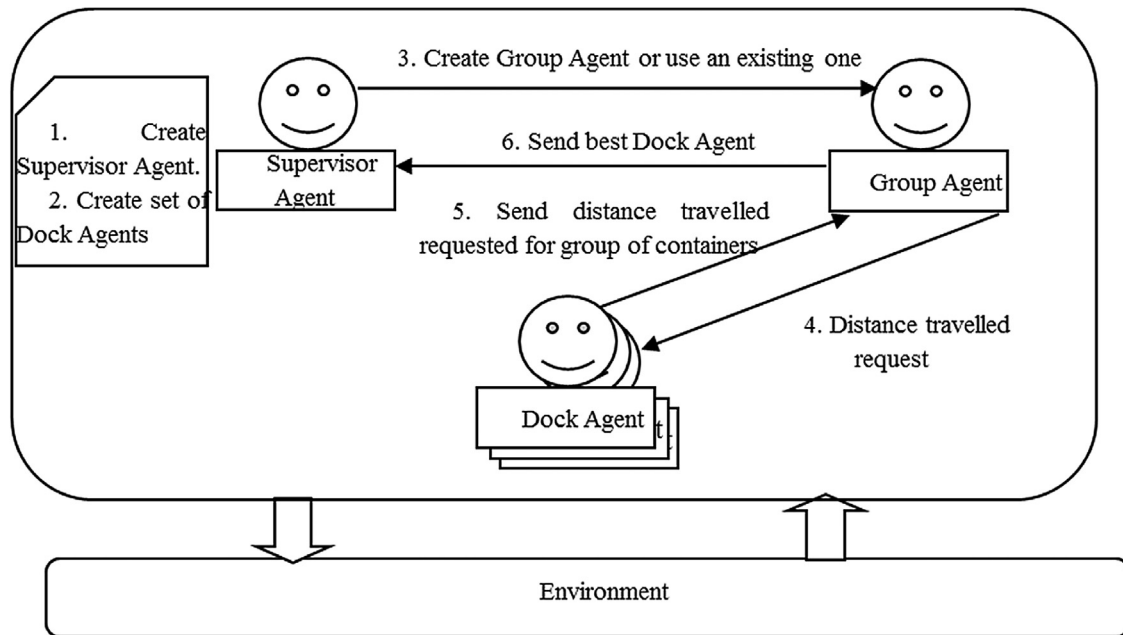


Fig. 6. Communication between types of agents.

provide a Multi-agent based approach to deal with perturbations in a realistic context.

4.1. Heuristic-based approach for the rail-road π -hub allocation problem

In order to consider the special characteristics and constraints of the problem, we propose a new heuristic called the “Best Fit Grouping Heuristic (BFGH)” [26]. The main idea of the heuristic is to take the closest containers heading for the same destination, and group them in such a way that each group does not exceed the capacity of a single truck.

The train is composed of five blocks. In each block, trucks which are not completely full (waiting from previous blocks) are filled,

and the remaining containers are assigned in four steps: the first step of the BFGH is to randomly select a new destination and determine all the containers going to this destination. In the second step, the containers are grouped (ensuring that they satisfy the capacity requirements of the truck), and checked to ensure that they are heading for the same destination. In the third step, the solution obtained from the previous step is improved by reassigning the containers to minimize the number of trucks used. Finally, in the fourth step, the groups of containers are assigned to the nearest dock, based on the average position of the containers.

In order to minimize the distance travelled by the containers, the solution obtained with the BFGH will be used as an initial solution for the Simulated Annealing (SA) metaheuristic.

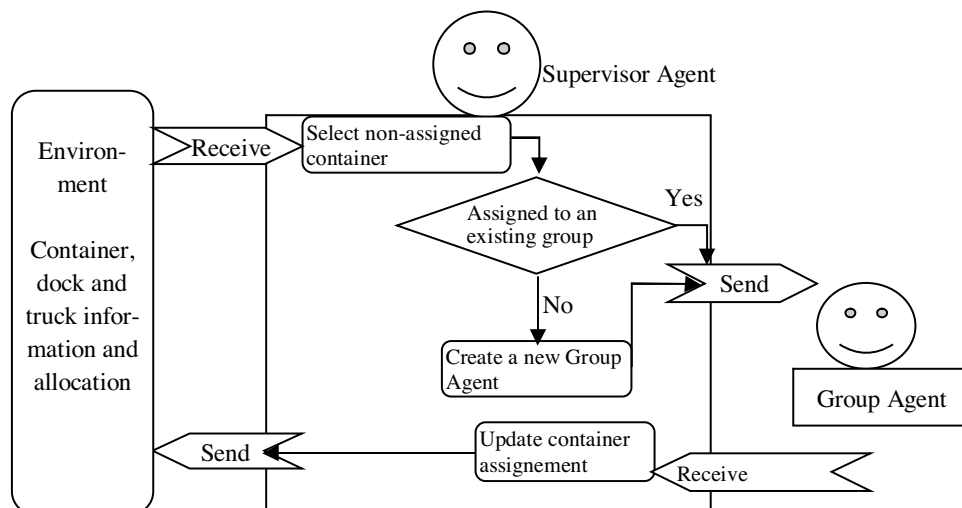


Fig. 7. Supervisor Agent architecture.

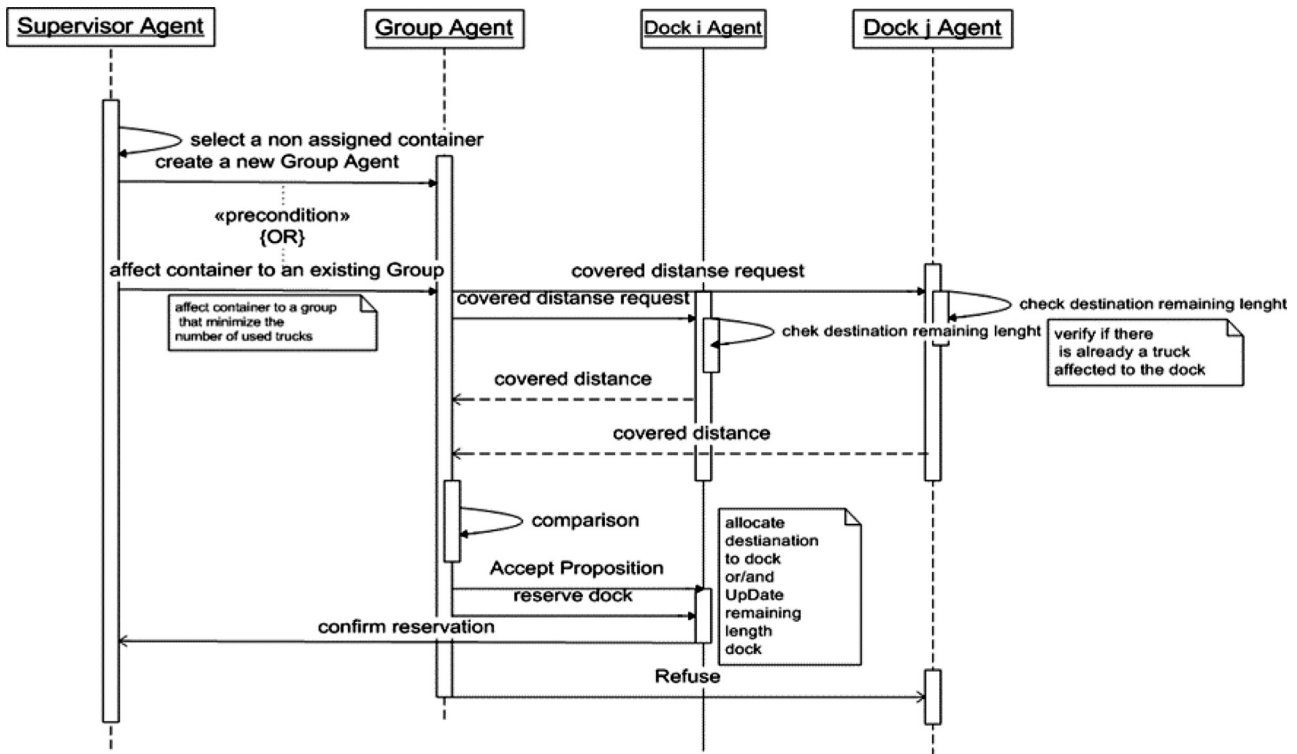


Fig. 8. Sequence diagram of Multi-Agent interaction.

4.2. Simulated annealing-based approach to the rail-road π -hub allocation problem

Simulated annealing is a local search method that is able to explore the solution space stochastically and try to avoid being trapped in local minima [27]. In this paper, we improved the classical SA by considering different neighbourhoods which make the method more like a hybrid SA-VNS (Variable Neighbourhood Search [28]). The method was also modified by introducing other improvements: using a predefined solution (not random, as usual), and retaining the best solution [29]. The SA metaheuristic is composed of three steps. In the first step, we generate the initial SA solution. Since the choice of initial solution greatly affects the results, we defined the result of the BFGH heuristic as an initial solution (step1 in Fig. 4). In the second step, to develop this solution, three neighbourhood structures were considered. The first neighbourhood consists of randomly choosing two docks and switching their trucks (including their containers). The second neighbourhood involves choosing one or two containers and switching them (by maintaining the docks in the solution obtained through BFGH). The third neighbourhood is a combination of both

previous moves: choosing random containers and their docks. The best solution obtained by these three neighbourhoods is thus maintained (step 3 in Fig. 4). In the third and final step, we decide to accept or reject the solution obtained, depending on the temperature parameter (see Fig. 4).

4.3. Multi-agent-based approach for the rail-road allocation problem

The optimization models given earlier are based on static information. The composition of the train should be known and established two hours before its arrival. The PI-hub structure is also supposed to function properly (no perturbations at the docks, all trucks are available, etc.). However, in daily life, many problems occur which disrupt the proper functioning of the PI-hub (container disposition is different from what is expected, some docks are out-of-use, etc.).

In order to take uncertainties and real-time information into account, we used the multi-agent optimization method to generate reactive solutions.

A Multi-Agent System (MAS) is a set of agents situated in a common environment, which interact and attempt to reach a set of

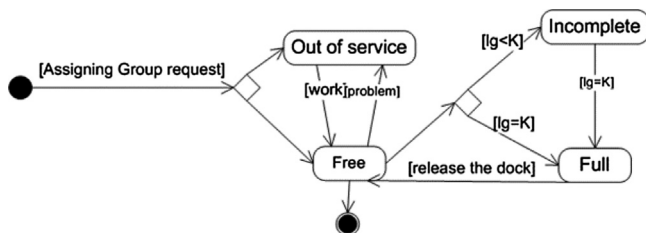


Fig. 9. Dock Agent state diagram.

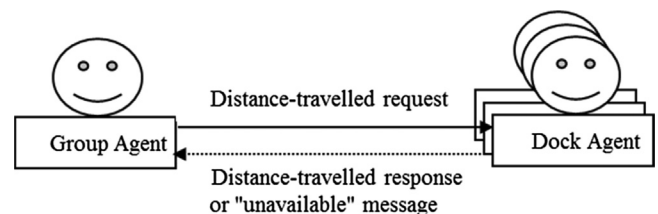


Fig. 10. Generation of distance travelled.

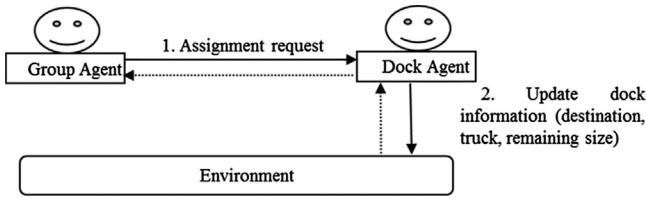


Fig. 11. Dock allocation.

goals [30]. An agent is “anything that can be viewed as perceiving its environment through sensors, and acting upon that environment through effectors” [31] (Fig. 5).

4.3.1. Agent types

In this study, we propose a Multi-Agent System model. This model is composed of three types of agents which communicate with each other and with the environment (see Fig. 6): a Supervisor Agent, a subset of Group Agents, and a subset of Dock Agents. The Supervisor Agent creates or uses an existing Group Agent, which is composed of a set of containers. Then this group sends a distance-travelled request to all the Dock Agents. The distance travelled represents the distance travelled (Euclidean distance) by the group of containers from the train to the dock. Every Dock Agent sends the distance travelled to the Group Agent, who then selects the best dock, based on the value of the distance travelled. The environment represents all the information at the terminal (for containers: size, destination, current position; for docks: coordinates, current destination; for trucks: truck size, remaining size, current destination, etc.)

In the following subsections, the role of each type of agent is explained.

4.3.1.1. Supervisor agent. The Supervisor Agent is an agent which manages the creation of Group Agents. It receives all the information about the containers: train number, block number, railcar, container position on the train, destinations, length, etc. It selects a container that is not yet assigned. Then, it checks if there are any existing groups of containers with the same destination and that respect the capacity of a truck. If it finds any such group of containers, it selects it; otherwise, it creates a new group of containers including the selected container (Fig. 7).

4.3.1.2. Group Agent. The Group Agent is composed of a set of containers with the same destination, and whose length/size does not exceed the truck’s capacity. This agent interacts with the Supervisor Agent: it accepts the request of the Supervisor Agent to add a new container to the group. Then, it sends a request message to all Dock Agents inquiring about the distance travelled cd . The distance travelled is the sum of all the distances travelled by the group of containers to reach the dock.

$$cd = \sum_{i=1}^N \sum_{p=1}^P d_{ip},$$

where d_{ip} is the distance travelled by container i to reach dock p .

When the Group Agent receives all the distances travelled from all docks, it chooses the best dock (the one with the smallest cd).

4.3.1.3. Dock agent. The Dock Agent is a reactive agent representing the dock and all the information about its contents (dock number, position, etc.). When the Dock Agent receives a request from the Group Agent, it generates the distance travelled

by the group of containers based on the information received from the environment. Then, it sends this value to the Group Agent.

Based on these definitions of the multi-agent system, we propose a new approach called Decision-making Mechanism Agents Heuristic (DMAH). Then we adapt this approach to a static case to compare it with SA and BFGH approaches.

4.3.2. Decision-making mechanism agents heuristic (DMAH)

Fig. 8 shows a simplified sequence diagram of the interactions between agents. This diagram is valid for both the DMAH approach and the reactive approach (presented later in Section 5.2). The difference is that the reactive approach takes into account perturbations (such as the dock’s “out-of-service” state).

The main idea of this approach is to assign a group of containers to the “best dock” based on the distance travelled. Whenever there is a new container assigned to an old group of containers, this group must be re-assigned to another dock.

The agents must respect the following steps during these interactions:

1. The Supervisor Agent selects a container that has not yet been assigned.
2. **If** there is an existing Group Agent with the same destination and the length/size of the group is less than the truck’s size, **then** the selected container must be assigned to this Group Agent (if we find several “group agents”, we choose the “group agent” that minimizes the remaining space in the truck.)
3. **Else**, create a new Group Agent with the container.
4. The Group Agent sends a request message to all Dock Agents asking for the distance travelled.
5. Every available Dock Agent sends the distance travelled cd of the containers to the current Group Agent. The dock is considered available if it is free, or if there are already containers assigned to this dock that have the same destination, and the capacity of the dock still allows other containers to be added to the group.
6. The Group Agent selects the best Dock Agent with the shortest cd .
7. Assign or reassign the current group of containers to the selected Dock Agent.
8. **If** all containers in the current block of the train are assigned, **then** release the dock for which the container size equals that of the truck (i.e., the truck leaves the hub when it is full).

4.4. PI-hub allocation problem under perturbations

In the real world, several perturbations can occur. These perturbations or uncertainties can be either external (outside the hub) or internal (within the hub).

External uncertainties: the train and the truck arrival times, the freight flow/content of the railcar, the number of inbound and outbound trucks, etc.

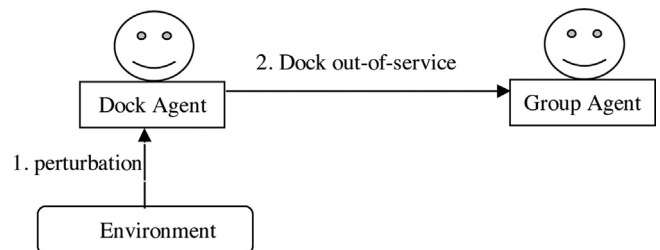


Fig. 12. Declaration of perturbations.

Table 2
Example distribution of containers in railcars.

| Railcar w | w1 | | w2 | | w3 | | w4 | | w5 | |
|---------------|-----|-----|----|------|------|------|------|------|------|------|
| Container c | c1 | c2 | c3 | c4 | c5 | c6 | c7 | c8 | c9 | c10 |
| Length l_i | 4.8 | 6 | 12 | 1.2 | 3.6 | 6 | 3.6 | 2.4 | 2.4 | 12 |
| dc | d2 | d4 | d1 | d2 | d1 | d2 | d4 | d4 | d2 | d2 |
| mid_i | 2.4 | 7.8 | 26 | 32.6 | 41.8 | 46.6 | 61.8 | 64.8 | 81.2 | 88.4 |

Internal uncertainties: processing time (loading and unloading time, transshipment time), truck departure times, availability of facilities, etc.

In this paper, the last type of uncertainty – the *availability of facilities*, and more specifically the *availability of the docks* – is considered. We feel that a related perturbation could be when some docks are out-of-use for certain periods, and how our system reacts to that.

To evaluate our approach under these perturbations, we generated scenarios where docks were out-of-service during a specific period of time (dock is out-of-service, truck assigned to the dock is out-of-service, or conveyors in front of the dock are out-of-service). In the following sub-section, the different states and reactions of the Dock Agent are presented.

4.4.1. Dock agent state

Every dock can attain four principal states: Free, Incomplete, Full and Out-of-service (Fig. 9).

4.4.1.1. *Free.* the dock is available and any container can be assigned to it.

4.4.1.2. *Incomplete.* There are some containers assigned to the dock, but the size of all the containers (lg) is less than the truck capacity (K). In this case, other containers can be assigned to the dock.

4.4.1.3. *Full.* There are some containers assigned to the dock, and the size of all the containers equals the truck capacity.

4.4.1.4. *Out-of-service.* The dock cannot be used during a specific period.

Table 3
Results of BFGH, SA and DMAH regarding the number of trucks used, and the filling rates.

| D | N | N/D | LB(T) | UB (FR) % | BFGH | | | SA | | | DMAH | | | UB vs. SA (FR) | UB vs. DMAH (FR) |
|----|-----|-------|-------|--------------|------|--------|--------------|----|--------|--------------|------|--------|--------------|-------------------|---------------------|
| | | | | | T | FR (%) | CPU time (s) | T | FR (%) | CPU time (s) | T | FR (%) | CPU time (s) | | |
| 4 | 60 | 15 | 26 | 87.41 | 26 | 87.41 | 1 | 26 | 87.41 | 100 | 26 | 87.41 | 3 | 0.00% | 0.00% |
| | 90 | 22.5 | 31 | 91.49 | 31 | 91.49 | 1 | 31 | 91.49 | 200 | 31 | 91.49 | 4 | 0.00% | 0.00% |
| | 137 | 34.25 | 41 | 94.6 | 43 | 90.27 | 1 | 43 | 90.27 | 260 | 42 | 92.42 | 7 | -4.65% | -2.30% |
| 6 | 60 | 10 | 27 | 90.23 | 28 | 87.01 | 1 | 28 | 87.01 | 146 | 29 | 84.01 | 5 | -3.57% | -6.89% |
| | 90 | 15 | 30 | 95.75 | 30 | 95.75 | 2 | 30 | 95.75 | 240 | 30 | 95.75 | 7 | 0.00% | 0.00% |
| | 132 | 22 | 41 | 94.67 | 41 | 94.67 | 1 | 41 | 94.67 | 220 | 42 | 92.42 | 8 | 0.00% | -2.38% |
| 10 | 60 | 6 | 28 | 77.59 | 28 | 77.59 | 2 | 28 | 77.59 | 170 | 28 | 77.59 | 5 | 0.00% | 0.00% |
| | 89 | 8.9 | 30 | 86.96 | 30 | 86.96 | 1 | 30 | 86.96 | 75 | 30 | 86.96 | 8 | 0.00% | 0.00% |
| | 132 | 13.2 | 43 | 88.79 | 44 | 86.77 | 1 | 44 | 86.77 | 327 | 43 | 88.79 | 10 | -2.28% | 0.00% |
| 15 | 60 | 4 | 29 | 78.99 | 29 | 78.99 | 1 | 29 | 78.99 | 103 | 29 | 78.99 | 7 | 0.00% | 0.00% |
| | 90 | 6 | 32 | 77.55 | 32 | 77.55 | 1 | 32 | 77.55 | 92 | 32 | 77.55 | 8 | 0.00% | 0.00% |

4.4.2. Dock agent reactions

Every Dock Agent reacts according to incoming messages from other agents and from the environment. There are three principal actions: generating the distance travelled, allocating the dock, and declaring perturbations.

- Generation of distance travelled: when the Group Agent sends a request message to all the Dock Agents asking for the distance travelled, each one should generate the distance travelled, if they are available (Fig. 10)
- Dock allocation: after comparing the different distances travelled, the Group Agent chooses the best Dock Agent and sends an assignment request to the dock to allocate it. When the dock agent receives this request message, it updates the dock information (destination, truck, remaining size, etc.)(Fig. 11).
- Declaration of perturbations: when the environment reports a perturbation the Dock Agent informs the Group Agent that the dock in question is out-of-service (Fig. 12).

5. Experimental simulation

In order to validate all the proposed approaches, the heuristic and the metaheuristic were implemented in Java. The Multi-Agent System-based approach, both with and without perturbations, were implemented in Java Agent Development framework (JADE), using a PC with an Intel(R) Core(TM) i3 processor, CPU 2.53 GHz, and 4Gb of RAM.

These approaches were tested on randomly-generated instances, as follows:

- The train, with 30 railcars, comprises blocks of 5 railcars.
- The number of containers N was chosen from the set {60, 90, 132 or 137},
- The number of destinations D was chosen among the values {4, 6, 10, 15},
- For each container, its length was generated uniformly from the different possible lengths {1.2, 2.4, 3.6, 4.8, 6, and 12}. The sum of the lengths of the containers in a railcar should be less than the length of the railcar.
- For each container, a destination was generated uniformly from the set {1, 2, . . . , D}.

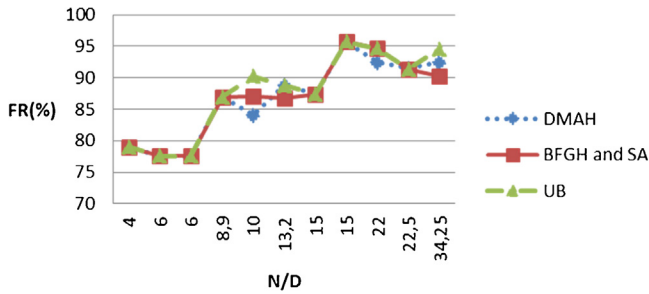


Fig. 13. Comparison of proposed approaches, and UB (Filling Rate).

Table 2 presents an example of the distribution of containers in railcars. In this example, in order to explain the method, we considered only one block $b1$ of the train. The block includes 5 wagons and each railcar is composed of a set of containers of different lengths lc , destinations dc (we have three possible destinations) and their position mid_i (according to the container axis).

The principal goals were to maximize the filling rates (FR) of the trucks, minimize the number of trucks used (T) and minimize the distance travelled (CD) by the containers from the train to the trucks. The minimization of T and the maximization of FR are related. However, when two solutions use the same number of trucks, FR could be different. For this reason, we used both criteria to distinguish the performance of two different solutions.

A lower and an upper bound were computed to evaluate the performance of the proposed approaches on T and FR.

The lower bound of the number of trucks used $LB(T)$ was computed as follows:

$$LB(T) = \left\lceil \frac{\sum_i lc_i}{K} \right\rceil$$

The upper bound of the filling rate $UB(FR)$ was computed as follows:

$$UB(FR) = \frac{\sum_i lc_i}{LB(T) \times K}$$

Where lc_i is the length of container i , K is the truck capacity, D is the number of destinations.

In the following subsections, we compare the three static approaches first (BFGH, SA and DMAH), and then the dynamic approach based on the scenario presented in Section 5.2.1.

Table 4 Results of BFGH, SA and DMAH for distance travelled with static case.

| D | N | N/D | BFGH CD (m) | SA CD(m) | DMAH CD (m) | BFGH VS. SA (CD) | SA VS. DMAH (CD) |
|----|-----|-------|----------------|-------------|----------------|---------------------|---------------------|
| 4 | 60 | 15 | 1860.79 | 1779.06 | 1714.9261 | -4.39% | -3.60% |
| | 90 | 22.5 | 2947.47 | 2753.85 | 2740.5112 | -6.57% | -0.48% |
| | 137 | 34.25 | 4224.26 | 4121.68 | 4386.062 | -2.43% | 6.41% |
| 6 | 60 | 10 | 2076.49 | 1945.12 | 1931.1873 | -6.33% | -0.72% |
| | 90 | 15 | 3059.92 | 2960.45 | 3026.9705 | -3.25% | 2.25% |
| | 132 | 22 | 4561.28 | 4400.18 | 4506.5195 | -3.53% | 2.42% |
| 10 | 60 | 6 | 2052.74 | 1909.89 | 1892.0896 | -6.96% | -0.93% |
| | 89 | 8.9 | 3202.02 | 3000.50 | 3046.7253 | -6.29% | 1.54% |
| | 132 | 13.2 | 4544.67 | 4400.78 | 4475.211 | -3.17% | 1.69% |
| 15 | 60 | 4 | 1929.58 | 1781.49 | 1761.5582 | -7.67% | -1.12% |
| | 90 | 6 | 3187.53 | 3023.25 | 3038.097 | -5.15% | 0.49% |

Table 5 Perturbation instances.

| Number of perturbations | Number of blocks "out of service" |
|-------------------------|-----------------------------------|
| 3 | 2 |
| | 3 |
| | 4 |
| | 5 |
| | |
| 5 | 2 |
| | 3 |
| | 4 |
| | 5 |
| | |
| 7 | 2 |
| | 3 |
| | 4 |
| | 5 |
| | |

5.1. The static scenario

Table 3 and Fig. 13 present the results obtained with the three static methods: BFGH, SA and DMAH.

In the SA heuristic, the BFGH is used as an initial solution. So, SA results should be similar to or better than those of the BFGH. Concerning the filling rate results, SA improves the distance travelled (see Table 4). Column "UB vs. SA (FR)" is defined by $SA(FR) - UB(FR)/UB(FR)$. It presents the effectiveness of the filling rate (FR) obtained with SA compared to the upper Bound $UB(FR)$. According to this column, the results obtained with SA are very close to the upper bound UB. The difference in the filling rate is about -0.95% (average). Column "UB vs. DMAH (FR)" is defined by $DMAH(FR) - UB(FR)/UB(FR)$. It focuses on the effectiveness of the filling rate (FR) obtained with DMAH compared to the upper bound $UB(FR)$. The difference between DMAH and UB is about -1.05%. According to these results, BFGH and SA both give results almost better than DMAH for the static case.

The BFGH heuristic takes a few seconds to reach the solution. According to column "CPU Time", the SA takes more CPU-time to converge, but it remains very reasonable (177 s on average).

Fig. 13 shows the variations in the different approaches proposed compared to the upper bound. The N/D ratio represents the average number of containers per destination. This curve shows that the filling rate percentage increases when the N/D ratio is high, and vice versa.

In Table 4, Column "BFGH vs. SA (CD)" presents the improvement (for distance travelled CD) of the metaheuristic SA results compared with the BFGH heuristic. It is defined by $SA(CD) - BFGH(CD)/BFGH(CD)$. According to this column, the

Table 6 Example of docks out-of-service.

| Docks out of service | Chosen blocks for duration |
|----------------------|----------------------------|
| k3 | b1 |
| | b2 |
| | b3 |
| k15 | b3 |
| | b4 |
| | b5 |
| k17 | b4 |
| | b5 |
| | b6 |

Table 7

Results of BFGH, SA and DMAH on the number of used trucks, and the filling rates in the dynamic case.

| D | N | N/D | LB (T) | UB (FR) % | BFGH | | | SA | | | DMAH | | | UB vs. SA (FR) | UB vs. DMAH (FR) |
|----|-----|-------|--------|-----------|-------|--------|--------------|-------|--------|--------------|------|--------|--------------|----------------|------------------|
| | | | | | T % | FR (%) | CPU time (s) | T % | FR (%) | CPU time (s) | T % | FR (%) | CPU time (s) | | |
| 4 | 60 | 15 | 26 | 87.41 | 26.16 | 86.87 | 1 | 26.16 | 86.87 | 130 | 26 | 87.413 | 3 | -0.62% | 0.00% |
| | 90 | 22.5 | 31 | 91.49 | 31.16 | 91.03 | 1 | 31.16 | 91.03 | 250 | 31 | 91.496 | 4 | -0.51% | 0.00% |
| | 137 | 34.25 | 42 | 92.42 | 42.16 | 92.07 | 3 | 42.16 | 92.07 | 260 | 42 | 92.424 | 7 | -0.38% | 0.00% |
| 6 | 60 | 10 | 27 | 90.23 | 28.37 | 85.91 | 1 | 28.37 | 85.91 | 150 | 29 | 84.013 | 5 | -4.79% | -6.90% |
| | 90 | 15 | 30 | 95.75 | 30.41 | 94.21 | 2 | 30.41 | 94.21 | 245 | 30 | 95.758 | 7 | -1.61% | 0.00% |
| | 132 | 22 | 41 | 94.67 | 41.16 | 94.11 | 2 | 41.16 | 94.11 | 220 | 42 | 92.424 | 8 | -0.60% | -2.38% |
| 10 | 60 | 6 | 28 | 77.59 | 28.41 | 76.62 | 2 | 28.41 | 76.62 | 180 | 28 | 77.597 | 5 | -1.26% | 0.00% |
| | 89 | 8.9 | 30 | 86.96 | 30.58 | 85.26 | 1 | 30.58 | 85.26 | 75 | 30 | 86.970 | 8 | -1.97% | 0.00% |
| | 132 | 13.2 | 43 | 88.79 | 44.41 | 85.84 | 3 | 44.41 | 85.84 | 377 | 43 | 88.795 | 10 | -3.32% | 0.00% |
| 15 | 60 | 4 | 29 | 78.99 | 30.08 | 75.50 | 1 | 30.08 | 75.50 | 113 | 29 | 78.997 | 7 | -4.42% | 0.00% |
| | 90 | 6 | 32 | 77.55 | 32.85 | 74.47 | 2 | 32.85 | 74.47 | 98 | 32 | 77.557 | 8 | -3.98% | 0.00% |

results obtained with SA are better than those obtained with BFGH. The reduction in the distance travelled is about 5.17%. We observed a maximum improvement of 7.67% in the distance travelled. This improvement is very important for the PI-hub since, for a good synchronization, containers should spend less time in the routing zone. By positioning the trucks in the best docks beside containers, the latter travel shorter distances and take less time to reach the destination docks.

Column 'SA vs. DMAH (CD)' presents the gap in terms of distance travelled CD between the SA and the DMAH heuristic. The DMAH heuristic gives a better solution than SA (reduction in distance travelled between 0.72% and 3.6%) when the number of containers decreases. However, when the number of containers is significant, the SA solution is better than DMAH (between 0.49% and 6.41%).

5.2. Dynamic scenario

With perturbations, static approaches become limited. Our dynamic approach DMAH and the static ones (SA and BFGH), adapted to take perturbations into account, were implemented on a dynamic scenario presented below.

5.2.1. Example of dynamic scenario

Many perturbations may occur in the PI-hub: dock breakdown, dock maintenance, perturbations on the conveyors in front of the dock, or any other perturbation which renders the dock unusable. To overcome these difficulties, the system must reallocate the group of containers to another available dock.

In order to test the approaches under perturbations, the following scenario is proposed: random docks could be out of service for a defined duration. The duration of this perturbation corresponds to the period in which a block of the train is treated. So, containers in this block could not be affected to this dock out of service.

So, for a dynamic scenario, the instances generated presented previously in the static context were used and perturbations were generated randomly as follows:

- Docks that will be out of service are chosen from the set of docks $\{k_1, k_2, \dots, k_{30}\}$.
- The number of perturbations (docks out of service) is chosen from the set $\{3, 5, 7\}$.
- Block b of the train is chosen from the set of blocks $\{b_1, b_2, \dots, b_6\}$ to obtain the duration for each perturbation.
- The number of blocks "out of service" is chosen from the set $\{2, 3, 4, 5\}$.

Table 8

Results of BFGH, SA and DMAH for the distance travelled with the dynamic approach.

| D | N | N/D | BFGH CD (m) | SA CD(m) | DMAH CD (m) | BFGH vs. SA (CD) | SA vs. DMAH (CD) |
|----|-----|-------|-------------|------------|-------------|------------------|------------------|
| 4 | 60 | 15 | 1874.90112 | 1791.20097 | 1733.55856 | -4.67% | -3.33% |
| | 90 | 22.5 | 2907.072 | 2770.34033 | 2778.26298 | -4.94% | 0.27% |
| | 137 | 34.25 | 4212.13513 | 4127.8968 | 4411.14364 | -2.04% | 6.33% |
| 6 | 60 | 10 | 2080.87078 | 1956.72716 | 1954.81094 | -6.33% | -0.10% |
| | 90 | 15 | 3089.22458 | 2976.69217 | 3129.1051 | -3.78% | 4.82% |
| | 132 | 22 | 4541.18887 | 4377.28209 | 4628.43311 | -3.75% | 5.39% |
| 10 | 60 | 6 | 2048.38853 | 1918.69493 | 1916.48987 | -6.75% | -0.17% |
| | 89 | 8.9 | 3179.42101 | 2993.51053 | 3142.16381 | -6.22% | 4.66% |
| | 132 | 13.2 | 4540.31788 | 4383.6473 | 4637.84384 | -3.50% | 4.30% |
| 15 | 60 | 4 | 1944.06971 | 1820.31553 | 1815.60592 | -6.23% | -0.90% |
| | 90 | 6 | 3163.45143 | 3021.19207 | 3236.22545 | -4.71% | 3.61% |

Based on this, different instances are generated by combining the number of perturbations and the durations (number of blocks) [Table 5](#).

The perturbations are generated on the scenario in the static instance. An example is given in the [Appendix A](#).

[Table 6](#) presents an example of an instance where the number of docks out of service is 3 and the “Number of out of service blocks” is 3. The dock *k15* is out-of-service during the unloading of containers in block 3, 4 and 5.

In this example, during the fourth block ‘b4’ of railcars, two Dock Agents (*k15* and *k17*) send a message to the Group Agent declaring that these docks are unavailable. Thus, the Group Agent has to assign all the containers in this group to another available dock.

5.2.2. Results for dynamic scenarios

The DMAH heuristic is based on the multi-agent system approach, it is thus a reactive approach and can handle perturbations better. Based on this reasoning, we evaluated the multi-agent approach (SMA-SC1) with the scenario presented in [Section 5.2.1](#).

In this dynamic scenario, the obtained results of BFGH, SA and DMAH concerning the number of used trucks and the filling rates are presented in [Table 7](#). It focuses on the effectiveness of the filling rate (FR) of DMAH and BFGH/SA results compared to the upper bound UB (FR). The difference between DMAH and UB is about 0,84% and the difference between BFGH/SA and UB is about 2,13%. According to these results, DMAH gives better results than BFGH and SA for the dynamic scenario.

In [Table 8](#), we compare the performance of the proposed approaches on the travelled distance (CD) criterion. Column “BFGH vs. SA (CD)” presents the improvement (for CD) of the metaheuristic SA compared with the BFGH heuristic. It is defined by $SA(CD) - BFGH(CD)/BFGH(CD)$. According to this column, the results obtained with SA were better than those obtained with BFGH. The reduction in the distance travelled is about 4.8%. We observed a maximum improvement of 6.75% in the distance travelled.

The gap (in terms of CD) between SA and DMAH is presented in the column “SA vs. DMAH (CD)”. The DMAH heuristic gives a better solution than SA (reduction in distance covered between 0.1% and 3.33%) when the number of containers decreases. However, the SA solution is better than DMAH when the number of containers

increases (between 0.27% and 6.33%). This is due to the fact that SA is proposed especially to improve the distance travelled while DMAH is more generic. However, we note that the CPU-time of DMAH is very short (less than 10 s) compared to that of SA (190 s in average), which means that in case of real-time implantation, it is preferable to take the reactive solution.

6. Conclusion and future research

The allocation problem in the PI-rail-road hub is considered in the context of the physical internet. This problem concerns the allocation of containers to the destination docks. Three static approaches are proposed to deal with the problem: a heuristic, a simulated annealing metaheuristic (SA), and a multi-agent system approach (DMAH).

Afterwards, a dynamic approach is proposed to take perturbations into account. The objective of the proposed methods is to minimize the distance travelled by containers when moved from trains to trucks (or docks) based on the target destination. The experiments were conducted according to two criteria: filling rate of trucks and travelled distances. The results were compared to lower and upper bounds. Heuristics and metaheuristic were implemented, tested and compared using randomly-generated instances.

The results show that for the static and dynamic case scenarios, the difference in the filling rate between SA and an upper bound is about 0.95% and 2.13% respectively. The difference between DMAH and the upper bound is about 1.05% and 0.84%, respectively. Concerning the distance travelled, SA is better than DMAH when the number of containers is high and DMAH is better when the number of containers is low. DMAH is faster than the SA metaheuristic, which is suitable for real-time implementation. Note that the experimentations were conducted on one dynamic scenario with 12 different instances, which make difficult for drawing conclusions. However, this study viewed as proof of concept allows showing the advantages and the limits of each approach on the tested instances.

Future studies will focus on three issues. Firstly, we are working to improve the metaheuristic by taking more uncertainties into account. Secondly, other perturbations and scenarios will be defined and integrated to test the approaches in a real-world context. And thirdly, we will address the coordination of the allocation problem with the routing problem in the PI-hub.

Appendix A.

Detailed example of the dynamic scenario (case with $D=4$ and $N=60$).

In the following static example we apply the perturbations given in [Table 5](#).

| Container | Wagon | Block | Train | Container length | Destination | mid_i |
|-----------|-------|-------|-------|------------------|-------------|---------|
| c1 | w1 | b1 | r1 | 4.8 | d2 | 2.4 |
| c2 | w1 | b1 | r1 | 6.0 | d4 | 7.8 |
| c3 | w2 | b1 | r1 | 12.0 | d1 | 26.0 |
| c4 | w2 | b1 | r1 | 1.2 | d2 | 32.6 |
| c5 | w3 | b1 | r1 | 3.6 | d1 | 41.8 |
| c6 | w3 | b1 | r1 | 6.0 | d2 | 46.6 |
| c7 | w4 | b1 | r1 | 3.6 | d4 | 61.8 |
| c8 | w4 | b1 | r1 | 2.4 | d4 | 64.8 |
| c9 | w5 | b1 | r1 | 2.4 | d2 | 81.2 |
| c10 | w5 | b1 | r1 | 12.0 | d2 | 88.4 |
| c11 | w1 | b2 | r1 | 12.0 | d2 | 6.0 |
| c12 | w1 | b2 | r1 | 3.6 | d4 | 13.8 |
| c13 | w2 | b2 | r1 | 6.0 | d4 | 23.0 |
| c14 | w2 | b2 | r1 | 1.2 | d1 | 26.6 |
| c15 | w3 | b2 | r1 | 4.8 | d2 | 42.4 |

(Continued)

| Container | Wagon | Block | Train | Container length | Destination | mid _i |
|-----------|-------|-------|-------|------------------|-------------|------------------|
| c16 | w3 | b2 | r1 | 6.0 | d4 | 47.8 |
| c17 | w4 | b2 | r1 | 12.0 | d1 | 66.0 |
| c18 | w4 | b2 | r1 | 1.2 | d3 | 72.6 |
| c19 | w5 | b2 | r1 | 6.0 | d1 | 83.0 |
| c20 | w5 | b2 | r1 | 6.0 | d2 | 89.0 |
| c21 | w1 | b3 | r1 | 12.0 | d1 | 6.0 |
| c22 | w1 | b3 | r1 | 2.4 | d3 | 13.2 |
| c23 | w2 | b3 | r1 | 6.0 | d3 | 23.0 |
| c24 | w2 | b3 | r1 | 4.8 | d3 | 28.4 |
| c25 | w3 | b3 | r1 | 12.0 | d1 | 46.0 |
| c26 | w3 | b3 | r1 | 4.8 | d4 | 54.4 |
| c27 | w4 | b3 | r1 | 4.8 | d3 | 62.4 |
| c28 | w4 | b3 | r1 | 3.6 | d1 | 66.6 |
| c29 | w5 | b3 | r1 | 2.4 | d2 | 81.2 |
| c30 | w5 | b3 | r1 | 12.0 | d4 | 88.4 |
| c31 | w1 | b4 | r1 | 2.4 | d1 | 1.2 |
| c32 | w1 | b4 | r1 | 6.0 | d1 | 5.4 |
| c33 | w2 | b4 | r1 | 12.0 | d3 | 26.0 |
| c34 | w2 | b4 | r1 | 2.4 | d3 | 33.2 |
| c35 | w3 | b4 | r1 | 4.8 | d3 | 42.4 |
| c36 | w3 | b4 | r1 | 2.4 | d4 | 46.0 |
| c37 | w4 | b4 | r1 | 12.0 | d1 | 66.0 |
| c38 | w4 | b4 | r1 | 2.4 | d1 | 73.2 |
| c39 | w5 | b4 | r1 | 2.4 | d2 | 81.2 |
| c40 | w5 | b4 | r1 | 1.2 | d4 | 83.0 |
| c41 | w1 | b5 | r1 | 3.6 | d4 | 1.8 |
| c42 | w1 | b5 | r1 | 4.8 | d4 | 6.0 |
| c43 | w2 | b5 | r1 | 4.8 | d1 | 22.4 |
| c44 | w2 | b5 | r1 | 6.0 | d3 | 27.8 |
| c45 | w3 | b5 | r1 | 4.8 | d1 | 42.4 |
| c46 | w3 | b5 | r1 | 4.8 | d3 | 47.2 |
| c47 | w4 | b5 | r1 | 6.0 | d2 | 63.0 |
| c48 | w4 | b5 | r1 | 1.2 | d1 | 66.6 |
| c49 | w5 | b5 | r1 | 6.0 | d1 | 83.0 |
| c50 | w5 | b5 | r1 | 4.8 | d1 | 88.4 |
| c51 | w1 | b6 | r1 | 1.2 | d3 | 0.6 |
| c52 | w1 | b6 | r1 | 1.2 | d1 | 1.8 |
| c53 | w2 | b6 | r1 | 2.4 | d2 | 21.2 |
| c54 | w2 | b6 | r1 | 3.6 | d3 | 24.2 |
| c55 | w3 | b6 | r1 | 1.2 | d3 | 40.6 |
| c56 | w3 | b6 | r1 | 4.8 | d2 | 43.6 |
| c57 | w4 | b6 | r1 | 4.8 | d1 | 62.4 |
| c58 | w4 | b6 | r1 | 6.0 | d1 | 67.8 |
| c59 | w5 | b6 | r1 | 1.2 | d1 | 80.6 |
| c60 | w5 | b6 | r1 | 1.2 | d1 | 81.8 |

Detailed results obtained on the example with 12 perturbations given in Table 5.

| D | N | Nb of OS docks | Nb of OS blocks | BFGH | SA | DMAH | BFGH vs. | SA vs. DMAH | AVG | AVG |
|---|----|----------------|-----------------|-----------|-----------|-----------|----------|-------------|--------|--------|
| | | | | CD (m) | CD(m) | CD (m) | SA (CD) | (CD) | | |
| 4 | 60 | 3 | 2 | 1860.7881 | 1782.1289 | 1715.5366 | -4.41% | -3.88% | -4.67% | -3.33% |
| | | | 3 | 1860.1732 | 1781.7366 | 1733.4471 | -4.40% | -2.79% | | |
| | | | 4 | 1862.5211 | 1779.5824 | 1737.8168 | -4.66% | -2.40% | | |
| | | 5 | 5 | 1861.195 | 1779.4724 | 1728.2701 | -4.59% | -2.96% | | |
| | | | 2 | 1860.8442 | 1778.1355 | 1731.5243 | -4.65% | -2.69% | | |
| | | | 3 | 1839.0349 | 1757.1461 | 1727.4929 | -4.66% | -1.72% | | |
| | | 7 | 4 | 1927.8969 | 1818.9706 | 1727.2078 | -5.99% | -5.31% | | |
| | | | 5 | 1867.4341 | 1796.7286 | 1731.0463 | -3.94% | -3.79% | | |
| | | | 2 | 1862.3485 | 1790.2634 | 1735.737 | -4.03% | -3.14% | | |
| | | | 3 | 1867.6166 | 1788.766 | 1738.8024 | -4.41% | -2.87% | | |
| | | | 4 | 1936.9957 | 1839.6942 | 1734.5525 | -5.29% | -6.06% | | |
| | | | 5 | 1891.9651 | 1801.7869 | 1761.2689 | -5.00% | -2.30% | | |

References

- [1] R.D. Meller, B. Montreuil, C. Thivierge, Z. Montreuil, Functional design of physical internet facilities: a road-based transit center, *Progress in Material Handling Research* (2012).
- [2] B. Montreuil, R.D. Meller, E. Ballot, Towards a Physical Internet: The Impact on Logistics Facilities and Material Handling Systems Design and Innovation, *Progress in Material Handling Research* (2010).
- [3] E. Ballot, B. Montreuil, C. Thivierge, Functional design of physical internet facilities: a road-rail hub, *Progress in Material Handling Research: MHR*. (2012).
- [4] M. Napolitano, Making the Move to Crossdocking: A Practical Guide to Planning, Designing, and Implementing a Cross Dock Operation, *Warehousing Education and Research Council*, Oak Brook (2000).
- [5] J. Van Belle, P. Vaclkeners, D. Cattrysse, Cross-docking: state of the art, *Omega* 40 (2012) 827–846.
- [6] K. Stephan, N. Boysen, Cross-docking, *J. Manage. Control* 22 (2011) 129–137.
- [7] N. Boysen, M. Flidner, Cross dock scheduling: classification, literature review and research agenda, *Omega* (2010) 413–422.
- [8] A.L. Ladier, G. Alpan, Cross-docking operations: current research versus industry practice, *Omega* 62 (2016) 145–162.

- [9] D.L. McWilliams, A dynamic load-balancing scheme for the parcel hub-scheduling problem, *Comput. Ind. Eng.* 57 (2009) 958–962.
- [10] L.Y. Tsui, C.H. Chang, A microcomputer based tool for assigning dock doors in freight yards, *Comput. Ind. Eng.* 19 (1990) 309–312.
- [11] L.Y. Tsui, C.H. Chang, An optimal solution to a dock door assignment problem, *Comput. Ind. Eng.* 23 (1992) 283–286.
- [12] Y. Oh, H. Hwang, C.N. Cha, S. Lee, A dock-door assignment problem for the Korean mail distribution center, *Comput. Ind. Eng.* 51 (2006) 288–296.
- [13] Z. Miao, A. Lim, H. Ma, Truck dock assignment problem with operational time constraint within crossdocks, *Eur. J. Oper. Res.* 192 (2009) 105–115.
- [14] Y. Kuo, Optimizing truck sequencing and truck dock assignment in a cross docking system, *Expert Syst. Appl.* 40 (2013) 5532–5541.
- [15] T.W. Liao, P.J. Egbelu, P.C. Chang, Simultaneous dock assignment and sequencing of inbound trucks under a fixed outbound truck schedule in multi-door crossdocking operations, *Int. J. Prod. Econ.* 141 (2013) 212–229.
- [16] A. Bellanger, S. Hanafi, C. Wilbaut, Three-stage hybrid-flowshop model for cross-docking, *Comput. Oper. Res.* 40 (2013) 1109–1121.
- [17] M.M. Goliás, G.K.D. Saharidis, S. Ivey, H.E. Haralambides, Advances in truck scheduling at a cross dock facility, *Int. J. Inf. Syst. Supply Chain Manage.* 6 (3) (2013) 40–62.
- [18] N. Boysen, M. Fliedner, M. Kellner, Determining fixed crane areas in rail–rail transshipment yards, *Transp. Res.* 46 (2010) 1005–1016.
- [19] A. Ballis, J. Goliás, Comparative evaluation of existing and innovative rail–road freight transport terminals, *Transp. Res. Part A* 36 (2002) 593–611.
- [20] N. Boysen, M. Fliedner, F. Jaehn, E. Pesch, A survey on container processing in railway yards, *Transp. Sci.* 47 (August) (2013) 312–329.
- [21] M. Klumpp, G. Sandhaus, Agent-based dynamic Cross-docking, Working Paper. Published in *Textsammlung zum deutsch-chinesischen Wissenschaftsdialog*. MA Akad.-Verl., pp. 45–63 (2013).
- [22] J. Van Belle, A holonic logistics execution system for cross-docking, PhD Thesis, Faculty of Engineering Science, KU Leuven, 2013 (October).
- [23] P. Furtado, Simulation and analysis of a physical internet network through multi-Agent systems, World Conference on Transport Research, Rio de Janeiro – Brazil, 2013.
- [24] E.S. Suh, Cross-docking assessment and optimization using multi-agent co-simulation: a case study, *Flexible Serv. Manuf. J.* 27 (2015) 115–133.
- [25] Pach C., Sallez Y., Berger T., Bonte T., Trentesaux D. et Montreuil B., Routing Management in Physical Internet Crossdocking Hubs: Study of Grouping Strategies for Truck Loading, *Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World*, vol. 438, pp. 483–490 (2014).
- [26] F. Walha, A. Bekrar, S. Chaabane, T. Loukil, A rail-road PI-hub allocation problems: model and heuristic, *International Physical Internet Conference: IPIC, Québec City- Canada*, 2014.
- [27] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, *Science* 220 (1983) 671–680.
- [28] N. Mladenovic, P. Hansen, Variable neighborhood search, *Comput. Oper. Res.* 24 (1997) 1097–1100.
- [29] F. Walha, S. Chaabane, A. Bekrar, T.M. Loukil, A Simulated annealing metaheuristic for a rail-road PI-hub allocation problem, *Stud. Comput. Intell.* 594 (2015) 307–314.
- [30] J.P. Jamont, M. Occello, A. Lagrèze, A multiagent approach to manage communication in wireless instrumentation systems, *Measurement* 43 (2010) 489–503.
- [31] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach.*: edition: Pearson – International edition (2010).



Dr. Abdelghani Bekrar received his engineering degree in computer science (1999) from INI (Algerian National high school), Master degree from ECN (French National high school), and his Ph.D. (2007) from the University of Technology of Troyes (UTT), France. *Abdelghani Bekrar* is currently an associate Professor at the University of Valenciennes. His research interests include Metaheuristic design and implementation and Hard optimization for engineer applications and supply chain management. He has published more than 20 papers and conference presentations and he is involved in several research projects.



Dr. Sondes Chaabane received her engineering degree in computer science engineering (1999) from ENSI, Tunisia and her Ph.D. (2004) from INSA of Lyon, France. She pursued her postdoctoral research (2005) at Catholic University of Mons (FUCAM), Belgium. She is currently full assistant professor at the University of Valenciennes and Hainaut-Cambrésis (UVHC, France) and researcher in LAMIH Laboratory. She is in charge of modeling and Simulation of production systems, operation research, planning and scheduling, performance assessment, and maintenance courses at the ENSIAME Engineering School at this university. Her research activities are based on manufacturing and hospital management problems such as scheduling under uncertainty, optimization and simulation process, strain situations modeling and piloting. Her scientific research includes: human and material resources sizing, planning and scheduling, logistics, hospital management.



Prof. Taicir Loukil has received her State doctorate from the Faculty of Economics and Management of Sfax, Tunisia in 2001. She worked as a chief of the department development and studies at the "Office des ports Aériens de Tunisie" prior to joining the University of Sfax. Her research activities include combinatorial optimization, multicriteria optimization, scheduling and logistics problems. She acted as a guest editor of a special issue on "Developments in Multiple Objective Programming and Goal Programming" at *International Transactions in Operations Research (ITOR)*. **Taicir Loukil** has authored or co-authored more than 50 scientific papers published in specialized reviews. She has supervised twenty PhD thesis and more than 30 master thesis. In 2003 she received the best paper award for the conference of the Administrative Sciences Association of Canada.



Faiza Walha is Currently Ph.D. student in Computer Science in the LOGIQ laboratory of the Higher Institute of Industrial Management of Sfax, Tunisia. She is working on active and reactive approaches to solve an allocation problem in a real environment, using the multi-agent systems. Her research interests include Multi-Agents Systems, operation research, optimization problems and algorithms, heuristic and meta-heuristic methods. Tel.: +216 95 66 97 26.