

Article

Multi-Objective Sustainable Truck Scheduling in a Rail–Road Physical Internet Cross-Docking Hub Considering Energy Consumption

Tarik Chargui ^{1,2,*} , Abdelghani Bekrar ¹, Mohamed Reghioui ² and Damien Trentesaux ¹ 

¹ LAMIH, UMR CNRS 8201, Université Polytechnique Hauts-de-France, Le Mont Houy, 59313 Valenciennes, France; abdelghani.bekrar@uphf.fr (A.B.); damien.trentesaux@uphf.fr (D.T.)

² RSAID, ENSATe, University of Abdelmalek Essaadi, Tétouan 93000, Morocco; m.reghioui@gmail.com

* Correspondence: tarik.chargui@gmail.com

Received: 15 May 2019; Accepted: 29 May 2019; Published: 3 June 2019



Abstract: In the context of supply chain sustainability, Physical Internet (PI or π) was presented as an innovative concept to create a global sustainable logistics system. One of the main components of the Physical Internet paradigm consists in encapsulating products in modular and standardized PI-containers able to move via PI-nodes (such as PI-hubs) using collaborative routing protocols. This study focuses on optimizing operations occurring in a Rail–Road PI-Hub cross-docking terminal. The problem consists of scheduling outbound trucks at the docks and the routing of PI-containers in the PI-sorter zone of the Rail–Road PI-Hub cross-docking terminal. The first objective is to minimize the energy consumption of the PI-conveyors used to transfer PI-containers from the train to the outbound trucks. The second objective is to minimize the cost of using outbound trucks for different destinations. The problem is formulated as a Multi-Objective Mixed-Integer Programming model (MO-MIP) and solved with CPLEX solver using Lexicographic Goal Programming. Then, two multi-objective hybrid meta-heuristics are proposed to enhance the computational time as CPLEX was time consuming, especially for large size instances: Multi-Objective Variable Neighborhood Search hybridized with Simulated Annealing (MO-VNSSA) and with a Tabu Search (MO-VNSTS). The two meta-heuristics are tested on 32 instances (27 small instances and 5 large instances). CPLEX found the optimal solutions for only 23 instances. Results show that the proposed MO-VNSSA and MO-VNSTS are able to find optimal and near optimal solutions within a reasonable computational time. The two meta-heuristics found optimal solutions for the first objective in all the instances. For the second objective, MO-VNSSA and MO-VNSTS found optimal solutions for 7 instances. In order to evaluate the results for the second objective, a one way analysis of variance ANOVA was performed.

Keywords: Physical Internet; cross-docking; Rail–Road; sustainability; truck scheduling; energy consumption; Multi-Objective Programming; Lexicographic Goal Programming; hybrid meta-heuristics

1. Introduction

Nowadays, global optimization of the supply chain is becoming the main goal of many industrial companies, especially the logistics distribution ones. The objective is to globally reduce the economical cost and to increase the productivity while taking into consideration the social and environmental aspects. Recently, the efficiency and reactivity of the supply chains has become a big challenge for distribution companies to satisfy retailers and customer demands in terms of cost, quality and delivery time [1,2]. Therefore, with the increase of environmental constraints, supply chain sustainability has emerged as a major approach for logistics firms to enhance their economical, social and environmental

sustainability [3–5]. Since supply chains are composed of different elements (suppliers, production plants, distribution centers, retailers and customers), a robust coordination is necessary to ensure the flexibility of the products flow. In addition to the linkage between all those components of the supply chain, the structure and the configuration of the supply chain has also a major impact on the supply chain sustainability [6,7].

Cross-docking is a type of distribution centers and it is considered as one of the most efficient distribution techniques used in supply chain management [8,9]. It consists on transferring products between inbound and outbound trucks, trains or other vehicles. The process of cross-docking starts with the unloading of the products from the inbound vehicles; then, the products may be either stored temporarily in the cross-dock facility or directly transferred and loaded into the outgoing vehicles.

In order to overcome the sustainability issue and to consider all its aspects (economical, environmental and social), Physical Internet (PI or π) has been presented as a new paradigm based on a metaphor from the Digital Internet by encapsulating products in standardized modular containers called PI-containers which are handled and moved using standard collaboration protocols inspired from TCP-IP protocols of the Digital Internet [10]. The idea of the PI is to build a worldwide interconnected logistics network through PI-nodes (PI-transits, PI-hubs, etc.) enabling the external and internal routing of PI-containers between the PI-nodes using PI-movers (PI-trucks, PI-wagons, etc.) [11]. The literature of the optimization problems related to the PI concept has considerably grown over the last few years [12–14]. Most of the papers in the PI literature focus on the global interconnected supply chain rather than PI-nodes, especially cross-docking PI-hubs.

In this paper, the focus is on the cross-docking terminals. There are many types of PI-nodes that can be identified in the PI network. For example:

- Road–Road PI-hubs: Cross-docking terminals used to efficiently transfer the PI-containers between inbound and outbound trucks through PI-sorters. They are composed of PI-sorters connected to the PI-docks using maneuvering areas that arrange the PI-containers after being unloaded from inbound trucks to be routed in the PI-sorters and then grouped and loaded into the outgoing trucks [15].
- Road–Rail PI-hubs: Use the same mechanism as the Road–Road PI-hubs. However, the PI-containers are transferred between trucks and trains and between trains and other trains using PI-sorters for routing and maneuvering areas to arrange containers. The interested reader can refer to [16] for a detailed functional design of the Road–Rail PI-hubs.
- Water–Road PI-hubs: Are designed to transfer the PI-containers between boats and trucks in a port terminal [10].

We center our attention in this paper on the Rail–Road cross-docking PI-hubs. The objective is to find a truck schedule that minimizes the total cost which includes the energy cost of the routing of PI-containers through the PI-conveyors and, at the same time, the cost of using outbound trucks for each different destination. The problem is formulated as a Multi-Objective Mixed-Integer Programming model (MO-MIP). The problem is then solved using two hybrid multi-objective meta-heuristics: Multi-Objective Variable Neighborhood Search hybridized with Simulated Annealing (MO-VNSSA) and with a Tabu Search (MO-VNSTS). The remainder of this manuscript is categorized as follows: In Section 2, we review the literature related to the Physical Internet, cross-dock truck scheduling, sustainability, and the solving methods for multi-objective problems, especially those related to the truck scheduling in cross-docks. Section 3 presents the description and the layout of the studied Rail–Road cross-docking PI-hub. The mathematical formulation of the problem is detailed in Section 4. Section 5 introduces the proposed solving methods. Numerical results are presented and analyzed in Section 6. Finally, a conclusion and possible directions for future works are presented.

2. Literature Review

This section presents the works in the literature that are related to the classical and PI cross-dock truck scheduling problems with the different solving approaches used in the literature. Then, the works addressing the sustainability issue are detailed for the global supply chain and especially in cross-docking terminals. Finally, the multi-objective solving approaches used in the literature are reviewed.

2.1. Classical Cross-Docking Terminals

Optimizing the cross-docking operations was widely addressed in the literature [8]. Several literature reviews and classifications were conducted concerning cross-docking optimization problems. The reader can refer to [8,9,17] for detailed classifications and reviews. Many papers addressed the cross-dock scheduling with single receiving and shipping dock with different approaches. For instance, Yu and Egbelu [18] formulated the problem mathematically as a Mixed-Integer Programming model (MIP) to minimize the makespan. The problem was then solved using a heuristic. Later, many authors suggested different approaches to solve the single door cross-dock truck scheduling problem using Genetic Algorithm [19], Hybrid Particle Swarm Optimization [20] and other various meta-heuristics [21,22]. Other researches addressed the problem considering different uncertainties in various situations (taking into account the breakdown of the trucks [23], the availability of cross-dock resources such as handling systems and dock doors [24], unknown trucks' arrival time [25], etc.).

In multiple-door cross-dock scheduling problem, which consists on assigning trucks to the inbound and outbound docks on a time horizon, various mathematical models were developed [8,26]. In a recent study, Gelareh et al. [27] proposed eight mathematical formulations for the cross-docking assignment problem in addition of three existing models in the literature. Then, a comparative analysis was performed on the proposed models on benchmark instances from the literature. For solving approaches, many meta-heuristics were suggested: Population based meta-heuristics (Differential Evolution [28,29], Diploid Differential Evolution [30], Genetic algorithm [25], Particle Swarm Optimization [31], etc.) and single solution based meta-heuristics (Tabu search [32], Variable Neighborhood Search [33], Simulated Annealing [33,34], etc.).

In the multi-objective context, Golias et al. [35] proposed a formulation of the inbound truck scheduling as a bi-objective and also as bi-level mixed integer programming model to minimize the total service time and the delayed completion which are conflicting objectives. Then, the two formulations were compared. A genetic algorithm and k-th best algorithm were proposed because of the complexity of the problem. Authors in [23] proposed a bi-objective mathematical formulation for cross-dock truck scheduling to minimize the total tardiness and weighted completion time while considering trucks' breakdown. The problem was then solved using three multi-objective meta-heuristics: Non-dominated Sorting Genetic Algorithm, Multi-Objective Simulated Annealing and Multi-objective Differential Evolution. Mohtashami et al. [36] developed a mathematical model in which the objective is to minimize the makespan, transportation cost and the number of truck trips. They proposed two population based meta-heuristics: Non-dominated Sorting Genetic Algorithm (NSGA-II) and the Multi-Objective Particle Swarm Optimization (MOPSO). The results show that the NSGA-II outperforms MOPSO. In [22], authors addressed the multi-criteria cross-dock scheduling through a unified objective function minimizing both earliness and tardiness of trucks. The authors proposed three population based meta-heuristic: Genetic Algorithm, Particle Swarm Optimization and Differential Evolution. The results showed that Differential Evolution is the most robust algorithm since it is less sensitive to the size of the problem. Other studies considered uncertainties with multi-objective optimization and proposed bi-objective bi-level approach to solve the cross-dock truck scheduling with unknown truck arrival time [37].

2.2. Physical Internet Cross-Docking Hubs

PI Cross-Docking Hubs are a cornerstone in the Physical Internet network. Several functional designs were suggested for different PI Cross-Docking Hubs. Meller et al. [15] presented a functional design for a Road-based transit center in which the PI-containers are transferred between inbound and outbound trucks. Another functional design was proposed by Ballot et al. [16] for Road–Rail PI-hubs with various key performance indicators from the customer's and PI-hub operator's perspective. As mentioned in the introduction, our focus is on the PI-hubs, and more precisely, the Road–Rail PI-hub which is considered as a PI-node in the Physical Internet network. As illustrated in Figure 1 Road–Rail PI-hubs are composed mainly of PI-sorters, maneuvering area, and storage areas.

Many researches were conducted to study various control and optimization problems related the Road–Rail PI-hubs. The PI-sorters, which are used for the routing of the PI-containers, occupies a large area of Road–Rail PI-hub. Thus, several researchers addressed the PI-containers routing problem with different approaches. Most authors focused on intelligent reactive approaches, usually exploiting multi-agent systems. One of the first papers dealing with this issue belongs to Pach et al. [38] in which a multi-agent model is developed for the routing process and grouping of PI-containers into the outgoing trucks in a Rail–Road cross-docking PI-hub. Different grouping strategies of the PI-containers were proposed to minimize the evacuation time and the tardiness of trucks loading the grouped PI-containers. One of the cornerstones of the Physical Internet paradigm is using standardized autonomous modular PI-containers that are easy to route in the PI-hubs and over the PI transportation network. In this context, an activeness concept of the PI-containers was introduced by Salles et al. [39] considering PI-container as autonomous and active products able to make decisions such as choosing the optimal routing path and to optimize the handling and moving operations especially in the cross-docking PI-hubs. Hybrid control architectures were addressed in several papers, such as in [40] by evaluating the performance of the Rail–Road PI-hub using simulation while considering both internal and external disruptions. An Optimized and Reactive hybrid Control Architecture (ORCA) was suggested by Vo et al. [41] through a simulation study for the PI-containers routing in a perturbed environment using different predictive and reactive strategies. Authors in [42] proposed a simulated annealing based meta-heuristic to allocate the PI-containers to the trucks while minimizing the distance. In order to consider uncertainties, a reactive approach based on multi agent systems is suggested to handle possible perturbations in the Rail–Road PI-hub. Several papers proposed mathematical formulations of the Rail–Road PI-hub problems such as PI-containers routing and truck scheduling and allocation of PI-containers to the outgoing trucks or trains' wagons. For instance, in [43], authors studied the truck assignment and proposed a mathematical formulation of the PI-containers allocation and outbound trucks scheduling in the Rail–Road section of the PI-hub. The objective is to minimize the distance traveled by the PI-containers in the PI-sorters to reach the outgoing trucks. Then, in order to find a solution within a reasonable time, a four steps heuristic is proposed. A mathematical model was suggested in [44] for the truck assignment in the Road–Rail section of the PI-hub. The model was then solved using a Tabu Search meta-heuristic.

2.3. Sustainability in Supply Chains and Cross-Docks

In recent years there has been a growing interest in supply chain sustainability from all its three dimensions: economical, social and environmental. Many studies showed the importance of the supply chain practices and the coordination between its different components (supplier, production plants distribution centers and customers) on the global sustainability of the entire supply chain [6,7,45]. The sustainability issue is usually addressed from a single aspect and must integrate and combine all the three dimensions [45]. Addressing the sustainability issue by integrating all its dimensions is becoming a challenging research direction. In order to integrate the sustainability into the planning decisions of the supply chain, Mota et al. [46] proposed a multi-objective mathematical model that takes into account the three aspects of the sustainability by suggesting assessment indicators for all the three aspects. Indeed, the economical dimension is considering the total cost which includes

the fixed costs, raw material costs, transportation costs and human resources. The environmental aspect is assessed through the environmental impact of different supply chain activities such as production, transportation. The social pillar is defined by a Social Benefit Indicator. The proposed model was then validated on real case study. Kong et al. [47] addressed the just in time concept in supply chain management and incorporated the environmental and economical impact into the batch scheduling problem through a mathematical model by minimizing multiple objectives: the earliness and tardiness, resources waste and environmental emission. The problem was then solved using a polynomial time algorithm. The food industry has an important effect on the sustainability of the supply chains especially from the environmental aspect. In this context, Demartini et al. [48] analyzed the sustainability in soft drink supply chains to determine the sustainable best practices and the key performance indicators that are the most related to the sustainability issues. Guo et al. [49] proposed a hybrid genetic algorithm approach to examine how the supply chain sustainability can be affected by the coordination between the production and transportation by considering several features such as departure times and transportation modes. The results showed that the coordination between the production and green transportation has an important impact on the global supply chain sustainability.

In the cross-docking context, the sustainability issue and especially energy consumption is often ignored and has not been largely addressed in the literature. To our best knowledge, few works have addressed sustainability and energy consumption in the cross-dock scheduling problem. For instance, Dulebenets [30] examined the sustainable truck scheduling and presented a mathematical formulation for the scheduling of inbound and outbound trucks in a cross-dock facility. Given a set of inbound and outbound trucks with their three attributed costs (handling, waiting and delayed departure costs), the objective is to find a schedule for the inbound and outbound trucks while minimizing the total cost. The mixed integer mathematical model was solved in CPLEX. Then, a Diploid Evolutionary Algorithm was proposed to solve the problem. The proposed algorithm outperforms the typical Evolutionary Algorithm from the cross-dock truck scheduling literature. As another research related to energy consumption, we can refer to the work of Shahram fard and Vahdani [50] where the authors address the trucks assignment and scheduling in a cross-docking center. They proposed a bi-objective mathematical model to schedule the inbound and outbound trucks taking into account the energy consumption of the forklifts. The problem is then solved using two multi-objective meta-heuristics. In this paper, we try to fill the gap in the literature by addressing the sustainability issue in cross-docking terminals in the Physical Internet context.

2.4. Multi-Objective Optimization Techniques

Multi-objective optimization problems are complicated to solve compared to single-objective problems. In the cross-docking context, various objectives can be considered. Some performance indicators concern inbound and outbound trucks such as loading and unloading time, makespan and preemption cost. While other objective functions are more related to the internal handling of products such as travel distance, total product stay time and inventory level. More detailed performance indicators can be found in the literature review of Ladier and Alpan [8]. Optimizing all these objectives can be challenging especially in case of conflicting objectives, thus any improvement in a criterion will generate a loss in the other criteria. Multi-objective problems can be solved using different approaches based on aggregation techniques, for example: the weighted sum, goal programming, lexicographic method, weighted min-max and other Pareto based approaches such as Non-dominated Sorting Genetic Algorithm and Multi-Objective Particle Swarm Optimization [51]. As reported by Duarte et al. [52], in the context of multi-objective optimization, population based meta-heuristics are widely used than single solution based meta-heuristics (also called trajectory based meta-heuristics). Indeed, single solution based meta-heuristics such as Tabu Search (TS) [53], Greedy Randomized Adaptive Search Procedure (GRASP) [54], Simulated Annealing (SA) [55], Variable Neighborhood Search [56], Guided Local Search (GLS) [57], etc, are not used as much as population based meta-heuristics (Genetic Algorithm (GA) [58], Particle Swarm Optimization (PSO) [59], Ant

Colony Optimization (ACO) [60], etc.) to solve multi-objective problems [52]. For instance, Genetic Algorithms and Particle Swarm Optimization were used to solve various multi-objective optimization problems [61,62]. Moreover, authors in [63] found that Genetic and Evolutionary Algorithms are the most used, then Simulated Annealing and finally Tabu Search. However, in single solution based meta-heuristics, Tabu Search (TS) and Simulated Annealing (SA) received significant attention in solving multi-objective optimization problems [64,65]. Another category of trajectory based meta-heuristics is Variable Neighborhood Search which is based on the dynamic changing of the neighborhood during the search. Few researches adapted the Variable Neighborhood Search to solve multi-objective optimization problems [52]. For instance, Geiger [66] proposed a randomized VNS to solve the multi-objective flow shop scheduling by applying different neighborhood operators. Arroyo et al. [67] developed two Multi-Objective Variable Neighborhood Search algorithms (MOVNS1 and MOVNS2) for the single machine scheduling problem while minimizing two conflicting objectives: earliness/tardiness and total flow time. The results of their algorithm outperformed the MOVNS proposed by Geiger [66]. Pareto based approaches are also applied in infrastructure maintenance such as pavement preservation. In this context, Lu and Tolliver [68] addressed the pavement management multi-objective optimization problem by suggesting a simulated constraint boundary method (SCBM) to optimize conflicting objectives: minimizing the total cost and maximizing the smoothness of the pavements. The proposed SCBM provides Pareto solutions without requiring any preferences parameters for the objective functions and the decision maker does not have to convert the objectives' units to a cost or monetary unit to use the approach. The performance of the SCBM approach is then compared to the Genetic Algorithm which takes additional computational time to find the Pareto solutions. In a comparative study, Jaszkievicz [69] addressed the bi-objective set covering problem. The authors proposed a Pareto memetic algorithm (PMA) to find the set of Pareto-optimal solutions. The proposed PMA was compared to ten different multi-objective meta-heuristics. Yan et al. [70] proposed a bi-objective fuzzy mixed integer nonlinear programming model for the hazardous materials routing problem in a Road–Rail multimodal transportation network under uncertainty and sustainability constraints. The problem was then solved using a three-stage exact solution strategy.

This work aims to address the multi-objective sustainable truck scheduling problem in the Rail–Road Cross-docking PI-hub. The objective is to minimize both the cost of energy consumption used by the PI-conveyors for the routing of the PI-containers as well as the cost of using outbound trucks for each different destination. A Multi-Objective Mixed-Integer Programming model (MO-MIP) is suggested for minimizing both objectives. Then, due to the complexity of the problem, two Multi-Objective Variable Neighborhood Search based meta-heuristics are developed and hybridized with Simulated Annealing (MO-VNSSA) and Tabu Search (MO-VNSTS) to make use of the features of each algorithm. The mathematical model and the two meta-heuristics are then evaluated on several randomly generated instances.

3. Problem Description and Working Assumptions

This section presents a description of the Rail–Road PI-hub cross-docking process with its different functionalities and layout. The Rail–Road PI-hub is a type of PI-nodes that is used to transfer PI-containers from the train to the outgoing trucks. It is composed of mainly a PI-sorter, and two maneuvering areas in front of the train and the loading docks. The process of cross-docking starts by unloading the PI-containers from the wagons. Then, the PI-container are routed to the outbound docks, grouped by destination, and then loaded into the outgoing trucks (Figure 1).

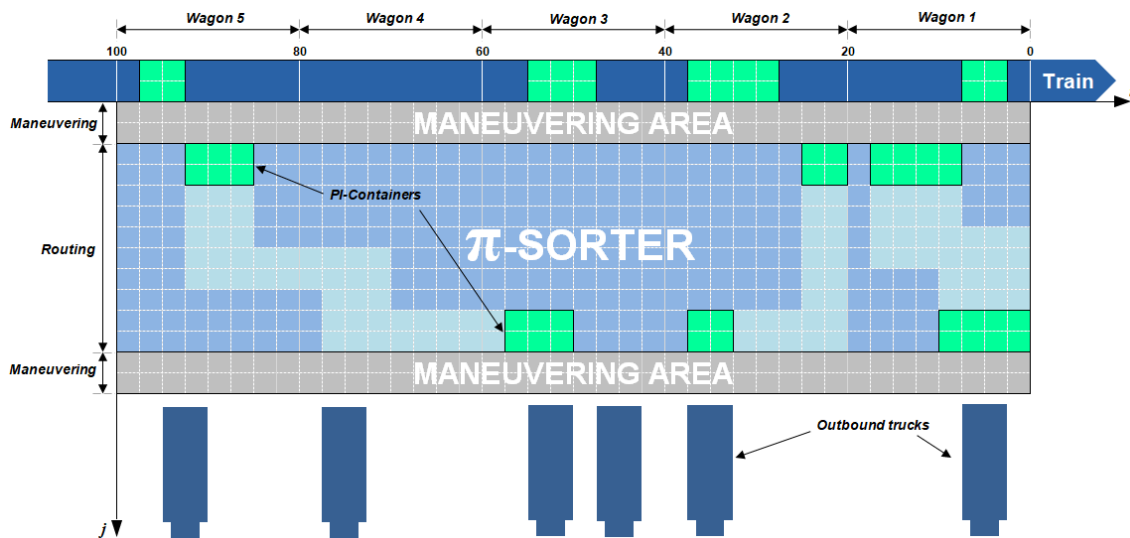


Figure 1. Overview of the layout of the Rail–Road PI-hub.

In the cross-dock scheduling context, objectives and constraints that are related to sustainability were not largely addressed in the literature. To the best of our knowledge, references [30,50] are the only ones considering sustainability or energy consumption constraints in classical cross-docking terminals. Therefore, in the PI context there is still a need for research that considers sustainable objectives and constraints for the Rail–Road PI-hubs cross-docking terminals. For this study, in the context of Rail–Road PI-hub cross-dock truck scheduling, insuring the sustainability consists on minimizing the use of the energy consumed by the PI-conveyors to move PI-containers between the outbound trucks and the train's wagons in addition to the minimization of the cost of the trucks used to serve each destination. Therefore, the cost of using outgoing trucks must be minimized while, at the same time, minimizing the PI-conveyors energy by finding the shortest path from the wagons to the outgoing trucks. The energy is then calculated by finding the number of PI-conveying units swept by the PI-containers during the way from the wagons until arriving to the trucks. Those two objectives are conflicting. Indeed, minimizing the PI-conveying energy could lead to the use of multiple trucks for the same destination, which is not an optimal choice, since the cost of using additional trucks for the same destination is expensive compared to the PI-conveying energy. Moreover, many trucks will leave the PI-hub with empty spaces. To face this issue, Multi-Objective approaches are developed and detailed in the next two sections: a Multi-Objective mathematical MIP model (MO-MIP) and two Multi-Objective meta-heuristics (MO-VNSSA and MO-VNSTS).

The following are the main assumptions considered for this problem:

- There are two objective functions to minimize: F_1 (the cost of using outbound trucks) and F_2 (the energy consumption cost for the PI-conveyors).
- The two objective functions are arranged in some order defined by the decision maker: minimizing F_1 and then F_2 in this study for example.
- The train unloads PI-containers that have different lengths and destinations.
- Each outbound truck must load only PI-containers that have the same destination.

4. Mathematical Formulation

This section presents the mathematical model of the studied sustainable truck scheduling problem in a Road–Rail PI-hub. The problem is formulated as Multi-Objective Mixed-Integer Programming (MO-MIP) model. The objective is to find the grouping of the PI-containers and the assignment and scheduling of the trucks at the docks. Given the lengths, destinations and positions of the PI-containers in the train and a set of available outbound trucks, the objective is to find the best grouping of the PI-containers in the trucks and the trucks' assignment and scheduling at the docks while minimizing

two conflicting objectives: The energy cost for using PI-conveyors to handle the PI-containers and, at the same time, the cost of using the outgoing trucks. The main input parameters/data and output variables are presented in Figure 2.

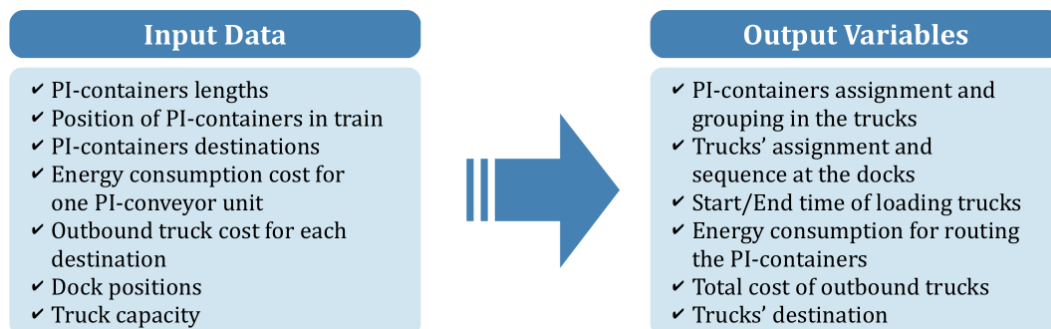


Figure 2. Input parameters and output variables of the Multi-Objective Mixed-Integer Programming (MO-MIP) model.

4.1. Problem Data and Parameters

- N Total number of PI-containers in the train
- K Total number of the outbound docks in the Rail→Road section
- D Total number of destinations of PI-containers
- H Total number of available outbound trucks
- i, j Indices of PI-containers to transfer from the train to the inbound trucks ($i, j = 1 \dots N$)
- k Index of the outbound docks in the Rail→Road section ($k = 1 \dots K$)
- d Index of the destinations of PI-containers to load into the outbound trucks ($d = 1 \dots D$)
- h, g Indices of the outbound trucks ($h, g = 1 \dots H$)
- C^E Cost of energy consumption for one unit of PI-conveyors
- C_d^T Cost of using an outbound truck for destination d
- I Time to load one PI-container into an outbound truck
- V Truck changeover time
- Y Vertical length of the Rail→Road section
- P_i Position of the bottom left corner of the PI-containers in the wagons of the train starting from the right axis of the Rail→Road section
- R_k Position of the outbound dock k starting from the right axis of the Road→Rail section
- L_i Length of a PI-container i to unload from the train
- G_{di} Two dimension binary matrix ($D \times N$) containing the destination of each PI-container, where:

$$G_{di} = \begin{cases} 1 & \text{if } d \text{ is the destination of the PI-container } i \text{ to load in outbound trucks} \\ 0 & \text{Otherwise} \end{cases}$$

- Q Truck capacity
- M A sufficient big positive number. The minimum value of M must be above:

$$M > \text{Max}(2 \times \text{Length of 5 wagons block} + Y \times \text{Max}_{i=1 \dots N}(L_i), \text{Planning Horizon})$$

4.2. Decision Variables

We consider the following decision variables in this MIP model:

4.2.1. Binary Variables

$$x_{hk} = \begin{cases} 1 & \text{if the outbound truck } h \text{ is assigned to the outbound dock } k \\ 0 & \text{Otherwise} \end{cases}$$

$$p_{ih} = \begin{cases} 1 & \text{if PI-container } i \text{ is assigned to the outbound truck } h \\ 0 & \text{Otherwise} \end{cases}$$

$$a_{hd} = \begin{cases} 1 & \text{if } d \text{ is the destination of the outbound truck } h \\ 0 & \text{Otherwise} \end{cases}$$

$$v_h = \begin{cases} 1 & \text{if outbound truck } h \text{ is used} \\ 0 & \text{Otherwise} \end{cases}$$

$$n_{hg} = \begin{cases} 1 & \text{if outbound trucks } h \text{ and } g \text{ are assigned to the same outbound dock} \\ & \text{and } h \text{ is a predecessor of } g \\ 0 & \text{Otherwise} \end{cases}$$

4.2.2. Integer Variables

z_{ih} The area (Number of PI-conveyor units) swept by PI-container i to arrive at its destination at the outbound truck h

4.2.3. Continuous Variables

b_h Start time for loading outbound truck h

q_h End time for loading outbound truck h (completion time)

4.3. Objective Function

The objective function of the this MO-MIP model is to minimize two conflicting objectives:

$$\begin{aligned} \text{Minimize:} \quad F_1 &= \sum_{h=1}^H \sum_{d=1}^D C_d^T a_{hd} \\ \text{Minimize:} \quad F_2 &= C^E \sum_{i=1}^N \sum_{h=1}^H z_{ih} \end{aligned} \tag{1}$$

where:

F_1 The total cost of trucks used for all destination

F_2 The total energy consumption cost for PI-conveyors

4.4. Constraints

$$\sum_{h=1}^H p_{ih} = 1 \quad (\forall i = 1 \dots N) \tag{2}$$

$$\sum_{i=1}^N p_{ih} L_i \leq Q \quad (\forall h = 1 \dots H) \tag{3}$$

$$p_{ih} + p_{jh} \leq \sum_{d=1}^D G_{di} G_{dj} + 1 \quad (\forall i, j = 1 \dots N, \forall h = 1 \dots H : i \neq j) \quad (4)$$

$$p_{ih} \leq v_h \quad (\forall i = 1 \dots N, \forall h = 1 \dots H) \quad (5)$$

$$a_{hd} \leq G_{di} + 1 - p_{ih} \quad (\forall i = 1 \dots N, \forall h = 1 \dots H, \forall d = 1 \dots D) \quad (6)$$

$$v_h = \sum_{d=1}^D a_{hd} \quad (\forall h = 1 \dots H) \quad (7)$$

$$z_{ih} \geq 2|P_i - R_k| + YL_i - M(2 - (p_{ih} + x_{hk})) \quad (\forall i = 1 \dots N, \forall k = 1 \dots K, \forall h = 1 \dots H) \quad (8)$$

$$\sum_{k=1}^K x_{hk} = v_h \quad (\forall h = 1 \dots H) \quad (9)$$

$$x_{hk} + x_{gk} - 1 \leq n_{hg} + n_{gh} \quad (\forall k = 1 \dots K, \forall h, g = 1 \dots H, h \neq g) \quad (10)$$

$$n_{hg} + n_{gh} \leq 1 \quad (\forall h, g = 1 \dots H) \quad (11)$$

$$b_g = \text{Max}(0, q_h + V - M(1 - n_{hg})) \quad (\forall h, g = 1 \dots H) \quad (12)$$

$$q_h = b_h + I \sum_{i=1}^N p_{ih} \quad (\forall h = 1 \dots H) \quad (13)$$

$$x_{hk}, p_{ih}, a_{hd}, v_h, n_{h,g} \in \{0, 1\} \quad (\forall i = 1 \dots N, \forall k = 1 \dots K, \forall d = 1 \dots D, \forall h, g = 1 \dots H) \quad (14)$$

$$z_{ih} \geq 0, b_h \geq 0, q_h \geq 0 \quad (\forall i = 1 \dots N, \forall h = 1 \dots H) \quad (15)$$

4.5. Constraints Description

Constraint (2) checks that each PI-container is assigned to only one outbound truck. Constraint (3) ensures that the outbound trucks' capacity is not exceeded by the PI-containers. Constraint (4) ensures that each outbound truck, if used, can load only PI-containers with the same destination. Constraint (5) guarantees that an outbound truck is used if at least one PI-container is assigned to it. Constraint (6) finds the destination of a used outbound truck based on the PI-containers that are assigned to it. Constraint (7) is to guarantee that each used truck has only one destination. Constraint (8) computes the number of PI-conveyors units swept by a PI-container to arrive at its destination in the outbound trucks. Constraint (9) ensures that each inbound truck has to be assigned to only one dock. Constraints (10) and (11) handle the correct relationship between the assignment variable (x_{hk}) and the sequencing variable (p_{ih}). Constraint (12) calculates the starting time of loading outbound trucks taking into account the end time of loading previous trucks if they are processed at the same dock. Constraint (13) calculates the end time of unloading the outbound trucks. Constraint (14) guarantees that the decision variables: $x_{hk}, p_{ih}, a_{hd}, v_h, n_{hg}$ are binary. Finally, Constraint (15) ensures that the decision variables: z_{ih}, b_h, q_h are positive.

5. Solving Methods

Due to high computational times, solving the mathematical model is not practical for large size instances. Therefore, developing meta-heuristics algorithms can provide near optimal or optimal solutions within reasonable computational times. Meta-heuristics are widely used in the literature to solve multi-objective optimization problems [71–74], using different techniques such as the weighted sum, exponential weighted criterion, lexicographic method, weighted min-max method, etc. The solving methods proposed in this paper are based on the Lexicographic Goal Programming method, which consists on ordering the objective functions in order of importance [51]. In Lexicographic Goal Programming, the first objective is considered the most important criterion, and it is worth any decreasing in the other objectives to improve the first criterion. The second objective is considered the most important one after the first criterion, and so on. The last objective is the least important among all the previous criteria.

In this paper, two hybrid meta-heuristics are proposed: Multi-Objective Variable Neighborhood Search hybridized with Simulated Annealing (MO-VNSSA) and the second one uses VNS hybridized with Tabu Search (MO-VNSTS). The VNS is chosen due its ability to dynamically change the neighborhood search. The hybridization with Tabu Search gives the VNS the ability to avoid the local search moves already performed. While Simulated Annealing lets the VNS to accept solutions at the beginning of the search even with high deviations and then the probability of accepting solutions is decreased and the algorithm become very selective in solutions. First, a construction heuristic (H_0) is proposed to build an initial solution by minimizing: on the one hand, the cost of used trucks by grouping the PI-containers in the trucks based on the Best Fit Decreasing algorithm while taking into consideration the destination constraints; on the other hand, finding an initial position of the trucks at the docks using the positions of the PI-containers to unload from the train. The main idea is to find a grouping of the PI-containers in the outgoing trucks to optimize the first objective F_1 , then, the generated solution is considered as an input or starting point for the two meta-heuristics (MO-VNSSA and MO-VNSTS) that try to optimize the second objective F_2 . In the following, a detailed description of the steps of each algorithm is presented (Construction Heuristic H_0 , MO-VNSSA and MO-VNSTS).

5.1. Construction Heuristic

Since the proposed meta-heuristics are based on the VNS which is a single solution based meta-heuristic, it is necessary to build an initial solution as a starting point for the proposed hybrid meta-heuristics. Therefore, a construction heuristic (Algorithm 1) is proposed to find an initial grouping of PI-containers and then the assignment and scheduling of the outgoing trucks at the docks. The first step of the heuristic is to group the PI-containers in the trucks after sorting them in a decreasing order depending on their lengths while taking into consideration the destination of each PI-container. In the second step, the average position of each truck is calculated depending on the position of the PI-containers that are assigned to that truck. This average position will be used later to find the dock on which the truck will be assigned for loading the PI-containers. The next step is to group and assign the PI-containers to the trucks by destination. If a truck does not have enough space or has a different destination, a new truck must be selected. The last step is to calculate the schedule of the trucks (starting/ending time of processing).

Algorithm 1 Overview of the heuristic (H_0) algorithm for the initial solution

```

1: Input : Problem data and parameters
2: Sort PI-containers in decreasing lengths
3: Group PI-containers by destination
4: Set:  $L_i$  length of each PI-container ( $i$ )
5: Set:  $A_h$  available length in each truck ( $h$ )
6: Set:  $A_h = Q$  initial capacity of the truck ( $h$ )
7: Set:  $F_h$  average position of a truck ( $h$ ) at the docks area
8: Set:  $Dist = 0$  (Sum of distances for the truck ( $h$ ))
9: for each PI-container's destination  $d$  do
10:   for each truck ( $h$ ) do
11:     if truck ( $h$ ) is a new empty truck then
12:       Assign destination ( $d$ ) to the truck ( $h$ )
13:     end if
14:     if  $L_i \leq A_h$  and PI-container ( $i$ ) and truck ( $h$ ) have same destination then
15:       PI-container ( $i$ ) is assigned to the truck ( $h$ )
16:        $A_h = A_h - L_i$ 
17:       Move to the next PI-container
18:     end if
19:   end for
20: end for
21: for each truck ( $h$ ) do
22:    $Dist = 0$ 
23:   for each PI-container ( $i$ ) do
24:     if PI-container ( $i$ ) is in truck ( $h$ ) then
25:        $Dist = Dist + PI\text{-}container's\ position\ in\ the\ train$ 
26:     end if
27:   end for
28:    $A_h = Dist / \text{Number of PI-containers in truck } (h)$ 
29:   Assign the truck ( $h$ ) to the closest dock to the average position  $A_h$ 
30:   Calculate the starting/ending time for loading truck ( $h$ )
31: end for
32: Return : Initial Solution ( $S_0$ )

```

5.2. Neighborhood Operators

In both meta-heuristics (MO-VNSSA and MO-VNSTS), there are three neighborhood operators performed at each inner iteration: Insertion, Swap and Insertion→Swap (Figure 3). The insertion consists on selecting a random truck in the current neighborhood and then assigning the selected truck

to another different random dock. The swap operator selects two different trucks and swaps their assignments. Finally, in the last operator (Insertion→Swap), a random truck is selected, inserted in a different dock, and then swapped with a random truck. All those three neighborhood operators are performed with equal probabilities at each iteration.

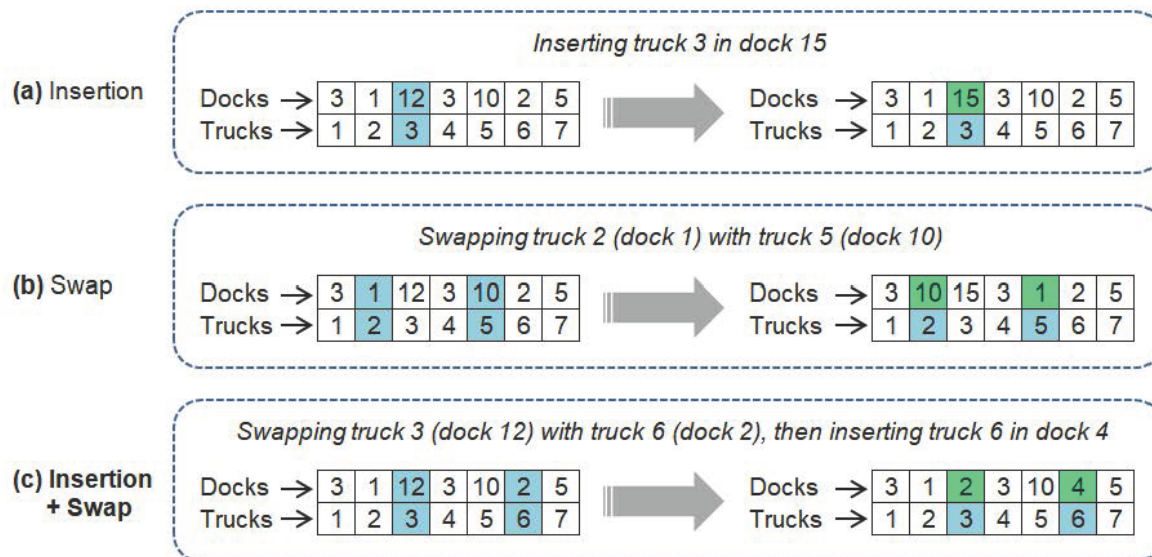


Figure 3. Neighborhood operators in Multi-Objective Variable Neighborhood Search hybridized with Simulated Annealing (MO-VNSSA) and Multi-Objective Variable Neighborhood Search hybridized with Tabu Search (MO-VNSTS): (a) Insertion (b) Swap (c) Insertion→Swap.

5.3. MO-VNSSA

The overall framework of the MO-VNSSA algorithm starts by loading the initial solution of the construction heuristic (Algorithm 2). Then a set of Nb_N neighborhood structures is generated. For each selected structure, the algorithm runs the Local Search based on the Simulated Annealing (SA) algorithm for Max_VNS iterations. At each iteration the temperature parameter T is set to its initial value T_Max . During the search, the temperature T is decreased by Δt after performing each Local Search move (Insertion, Swap and Insertion→Swap) with equal probabilities for each move. The new generated solutions (S') are accepted as current solution with a probability p :

$$p = \exp\left(\frac{S - S'}{T}\right)$$

At the beginning, when T is still higher, the algorithm tends to accept deteriorating moves. Then, as the temperature T is decreased, the algorithm becomes very selective on the new generated solutions (S'). This process is repeated for Max_VNS iterations and for each neighborhood structure. A shaking of the PI-containers assignments between the trucks with the same destination is performed after Max_Shake iterations. At each iteration of the Simulated Annealing, a random solution is generated in the current neighborhood structure, the generated solution (S') is accepted with the probability p as the current solution (S). Then, if $S' < S_{BEST}$, S' is considered as the current best solution found. The inner SA loop, which is repeated for Max_VNS iterations, stops once $T \leq \epsilon$. This process is repeated for each neighborhood structure. The algorithm stops after exploring all the neighborhood structures.

Algorithm 2 Overview of the hybrid MO-VNSSA algorithm

```

1: Input: Initial solution  $S_0$  generated by the heuristic ( $H_0$ )
2: Set:  $Max\_VNS, Nb\_N, Max\_Shake$ 
3: Set:  $T, \Delta t, T\_Max$  // SA parameters
4: Set:  $S_{BEST} = S_0$  // Best solution
5: Set:  $S = S_0$  // Current solution
6: Generate a set of  $Nb\_N$  neighborhood structures
7: for each neighborhood structure do
8:   for each Max_VNS iterations do
9:      $T = T\_Max$  // Reset  $T$ 
10:    while  $T > \epsilon$  do //  $\epsilon$  is very small float  $\epsilon = 1 \times 10^{-5}$ 
11:      Perform one of the LS moves on trucks' sequence: Insertion/Swap/Insertion→Swap in the selected neighborhood
12:      Calculate the objective function of the new solution  $S'$ 
13:       $r = Rand[0,1]$  //  $r$  is random float in  $[0,1]$ 
14:      if  $r < Exp((S - S')/T)$  then
15:         $S = S'$ 
16:        if  $S_{BEST} > S'$  then
17:           $S_{BEST} = S'$ 
18:        else
19:           $T = T - \Delta t$  // Reduce  $T$ 
20:        end if
21:      else
22:        Cancel the Local Search move
23:         $T = T - \Delta t$ 
24:      end if
25:    end while
26:  end for
27:  Shake the PI-containers assignments between the trucks with the same destination after each  $Max\_Shake$  iterations
28: end for
29: Return :  $S_{BEST}$ 

```

5.4. MO-VNSTS

In the MO-VNSTS meta-heuristic, the VNS is hybridized with Tabu Search which, contrary to the Simulated Annealing, makes use of memory. Indeed, at each iteration of the Tabu Search, each performed move is stored in a Tabu List so at the next iterations those moves will not be performed for a certain number of iterations. This mechanism helps the algorithm to learn from the past moves to avoid the deteriorating ones. An aspiration criterion is defined as the deviation between the new found solution (S') and the current solution (S) to prevent the algorithm from local optima. As described in Algorithm 3, the searching process starts by loading the initial solution from the construction heuristic. The PI-containers assignment is shaken between the trucks having the same destination after Max_Shake iterations. Then, the Tabu Search algorithm is performed for Max_TS iterations for each neighborhood structure. At each iteration, a new random solution is generated in the selected neighborhood (S'). If the aspiration criterion is checked for the new solution, the later is selected as a current solution ($S = S'$). If $S' < S_{BEST}$, then the new solution is selected as the current best solution. The algorithm ends after repeating this process for all the neighborhood structures. The main steps of the MO-VNSTS algorithm are presented in Algorithm 3.

Algorithm 3 Overview of the hybrid MO-VNSTS algorithm

```

1: Input: Initial solution  $S_0$  generated by the heuristic
2: Set:  $Max\_VNS, Nb\_N, Max\_Shake$ 
3: Set:  $TL = \emptyset, TL\_Size, Max\_TS$  // TS parameters
4: Set:  $S_{BEST} = S_0$  // Best solution
5: Set:  $S = S_0$  // Current solution
6: Generate a set of  $Nb\_N$  neighborhood structures
7: for each neighborhood structure do
8:   for each  $Max\_VNS$  iterations do
9:      $TL = \emptyset$  // Reset Tabu List
10:    for each  $Max\_TS$  iterations do
11:      Perform one of the LS moves on trucks' sequence: Insertion/Swap/Insertion→Swap in the selected neighborhood
12:      while Local Search move is in the  $TL$  do
13:        Generate a new Local Search move
14:      end while
15:      Calculate the objective function of the new solution  $S'$ 
16:      if  $S'$  satisfies the aspiration criterion then
17:         $S = S'$ 
18:        if  $S_{BEST} > S'$  then
19:           $S_{BEST} = S'$ 
20:        end if
21:      else
22:        Cancel the Local Search move
23:      end if
24:      Update the Tabu List  $TL$  with the new move
25:    end for
26:  end for
27:  Shake the PI-containers assignments between the trucks with the same destination after each  $Max\_Shake$  iterations
28: end for
29: Return :  $S_{BEST}$ 

```

6. Computational Experiments

After a brief description of the implementation of the mathematical model and the development of the two meta-heuristics, this section presents the obtained results on the randomly generated instances with an analysis of the obtained results.

6.1. Implementation and Instances

This section aims to evaluate the performance of the proposed meta-heuristics on several randomly generated instances. First, the multi-objective mathematical model is implemented and solved using Lexicographic Goal Programming optimization in IBM CPLEX solver (Version 12.9). A time limit of one hour (3600s) is set for the solver. The heuristic (H_0) and the two meta-heuristics (MO-VNSSA and MO-VNSTS) are developed in C++ and all the experiments are performed on an Intel^(R) Core^(TM) i3 processor with 4 GB of RAM. The tests are performed in 5 replications for each instance. The average value is presented. The proposed algorithms are tested on a set of small and large instances randomly generated by varying several parameters. The values of the parameters are summarized in Table 1. The tuning of the meta-heuristics parameters is presented in Table 2.

Table 1. Values of the parameters and the input data for the small instances.

Data	Abbreviations	Values
Number of PI-containers	N	[4, 12]
Number of destinations	D	[1, 3]
Number of docks	K	15
Lengths of PI-containers	L_i	{1, 2, 3, 4, 5, 10}
Cost of energy consumption for one PI-conveyor	C^E	0.5
Cost of using a truck for each destination	C_d^T	[200, 800]
Positions of PI-containers in the train	P_i	[1, 75]
Destination of PI-containers in the train	G_{di}	Random (Binary)

Table 2. Values of parameters in the meta-heuristics: MO-VNSSA and MO-VNSTS for small instances.

Algorithm Parameters	Abbreviations	Values
Number of iterations for each neighborhood	Max_VNS	500
Number of generated neighborhoods	Nb_N	3
Number of iterations before shaking PI-containers' assignments in trucks	Max_Shake	10
Simulated Annealing temperature decreasing step	Δt	0.01
Initial temperature in the Simulated Annealing	T_Max	1
Tabu List Size	TL_Size	10
Number of iterations for the Tabu Search process	Max_TS	100

6.2. Numerical Results

Table 3 shows the values of both objective functions: F_1 (Cost of used trucks per destination) and F_2 (Energy cost consumption). The first four columns present the parameters of the generated instances. The next three columns show the results obtained after solving the mathematical model using Lexicographic Goal Programming in CPLEX (F_1 and then F_2). The computational times are presented in the seventh column. CPLEX was able to determine the optimal solution for 23 instances. In the last four instances, CPLEX exceeds the time limit of 3600s without providing any feasible solution. The remaining of the columns shows the results of the two meta-heuristics (MO-VNSSA and MO-VNSTS) which provide optimal results for the first objective F_1 and near optimal values for the second objective F_2 within fast computational times compared to the ones of CPLEX. The optimal values are presented in bold. In order to evaluate more the performance of the two meta-heuristics, several large instances are generated. Parameters are modified for those large sized instances (Max_VNS = 1500, $\Delta t = 0.002$ and Max_TS = 500). The obtained results are presented in Table 4, and illustrated in Figure 4.

As it can be seen in Figure 5, the PI-conveyors' energy consumption, which represented by objective function F_2 , increases significantly with the number of PI-containers.

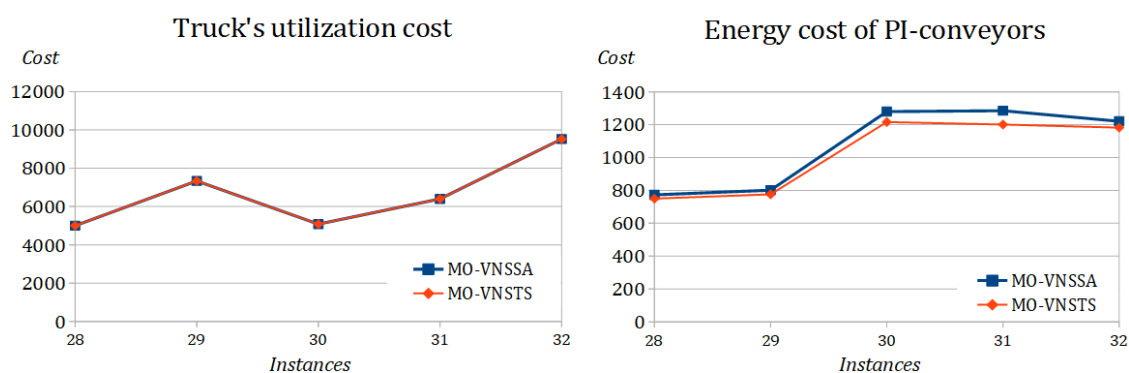
**Figure 4.** Objective functions for MO-VNSSA and MO-VNSTS for large instances.

Table 3. Experimental results for CPLEX, MO-VNSSA and MO-VNSTS on the generated instances.

Instances				CPLEX			MO-VNSSA			MO-VNSTS		
#	D	N	H	F ₁	F ₂	Time (s)	F ₁	F ₂	Time (s)	F ₁	F ₂	Time (s)
1	1	4	4	351	136	0.11	351	136	0.078	351	136	0.081
2	1	5	4	750	199	0.42	750	199	0.087	750	199	0.091
3	1	6	5	1166	152	0.69	1166	192	0.088	1166	192	0.081
4	1	7	5	1263	218	11.36	1263	282	0.084	1263	287	0.094
5	1	8	5	1821	234	2.78	1821	273	0.094	1821	273	0.100
6	1	9	7	891	310	14.86	891	357	0.116	891	360.2	0.144
7	1	10	7	1588	299	10.86	1588	431	0.134	1588	436	0.138
8	1	11	7	1276	310	13.36	1276	406	0.122	1276	406	0.138
9	1	12	7	2035	377	38.28	2035	494	0.140	2035	495	0.143
10	2	4	4	1096	136	0.14	1096	136	0.056	1096	136	0.078
11	2	5	4	1835	189	0.28	1835	194	0.066	1835	194	0.078
12	2	6	5	1033	183	0.30	1033	183	0.081	1033	183	0.091
13	2	7	5	1792	252	0.52	1792	306	0.084	1792	306	0.091
14	2	8	5	2510	243	1.69	2510	283	0.091	2510	283	0.112
15	2	9	7	1650	324	453.09	1650	355.8	0.112	1650	366	0.125
16	2	10	7	2098	362	277.84	2098	392	0.194	2098	394.2	0.131
17	2	11	7	1876	367	40.52	1876	400	0.147	1876	406.2	0.138
18	2	12	7	2688	371	186.89	2688	499	0.159	2688	499	0.153
19	3	4	4	1542	108	0.16	1542	111	0.059	1542	111	0.078
20	3	5	4	1923	206	2.09	1923	214	0.069	1923	214	0.094
21	3	6	5	1440	160	0.25	1440	160	0.084	1440	160	0.094
22	3	7	5	1797	293	17.17	1797	293	0.084	1797	293	0.094
23	3	8	5	2520	281	164.42	2520	281	0.091	2520	281	0.106
24	3	9	7	*	*	*	2331	372.8	0.106	2331	374.8	0.128
25	3	10	7	*	*	*	3102	422.6	0.116	3102	425.6	0.144
26	3	11	7	*	*	*	2042	401.4	0.122	2042	407	0.141
27	3	12	7	*	*	*	3699	495	0.141	3699	498.6	0.150

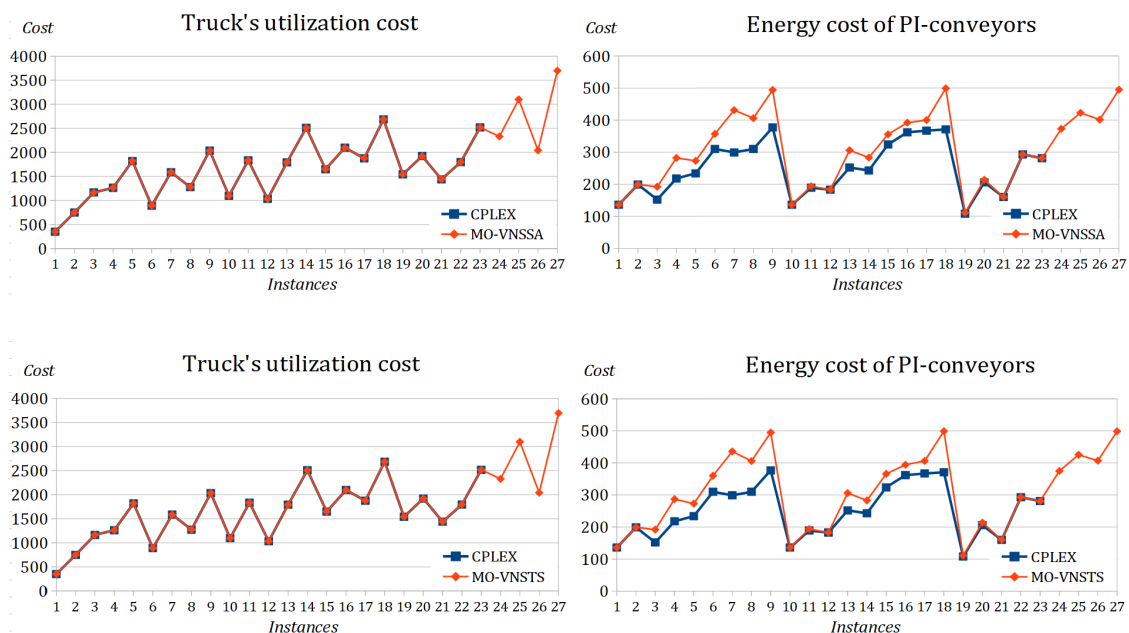
**Figure 5.** Objective functions in CPLEX, MO-VNSSA and MO-VNSTS for both objectives F_1 and F_2 .

Table 4. Experimental results for MO-VNSSA and MO-VNSTS for the large instances.

Instances				MO-VNSSA			MO-VNSTS		
#	D	N	H	F ₁	F ₂	Time (s)	F ₁	F ₂	Time (s)
28	7	20	15	5001	772.6	4.322	5001	750	4.503
29	10	20	15	7338	801.2	4.5872	7338	776	4.7218
30	7	30	20	5081	1279.8	7.4594	5081	1216	7.3938
31	10	30	20	6401	1284.8	7.35	6401	1201	7.4782
32	15	30	20	9529	1220.4	8.1656	9529	1182	8.2032

6.3. Sensitivity Analysis

6.3.1. ANOVA Measures

The two meta-heuristics MO-VNSSA and MO-VNSTS found optimal values for the first objective function F_1 . In order to show the insignificant difference for the second objective F_2 , a testing of hypothesis using analysis of variance ANOVA is performed for both meta-heuristics. The ANOVA test showed that both algorithms do not differ from the optimal value of F_2 at 95% of confidence limit. The detailed results are presented in Table 5. As p -value > 0.05 , there is not a significant difference between the meta-heuristics and the optimal values of CPLEX.

Table 5. ANOVA measure for MO-VNSSA and MO-VNSTS.

Meta-Heuristics	F	p -Value
MO-VNSSA	1.623	0.209
MO-VNSTA	0.726	0.196

6.3.2. Convergence Behavior

In order to illustrate the convergence behavior of the two meta-heuristics, multiple tests are performed with different values of the maximum number of iterations for each neighborhood structure (Max_VNS) which goes from 1 to 500 iterations. The obtained results are presented in Figure 6 which shows the impact of the value of the number of iterations (Max_VNS) on the total average deviation for the second objective (F_2).

6.4. Discussion

The results obtained by both meta-heuristics (MO-VNSSA and MO-VNSTS) seems to be promising in terms of quality of the obtained solutions especially for the first objective F_1 for which the optimal value was found for all the instances and near optimal and several optimal values for the second objective F_2 . The results seems to be positive also in terms of computational time even for large sized instances.

Within the context of multi-objective optimization, using the Lexicographic Programming method for the MO-MIP model and the two meta-heuristics (MO-VNSSA and MO-VNSTS) requires a priori knowledge about the objectives which is based on the decision makers' preferences. Therefore, the two objective functions presented in Tables 3 and 4 are solved in a lexicographic arranged order (which is F_1 and then F_2 in this study).

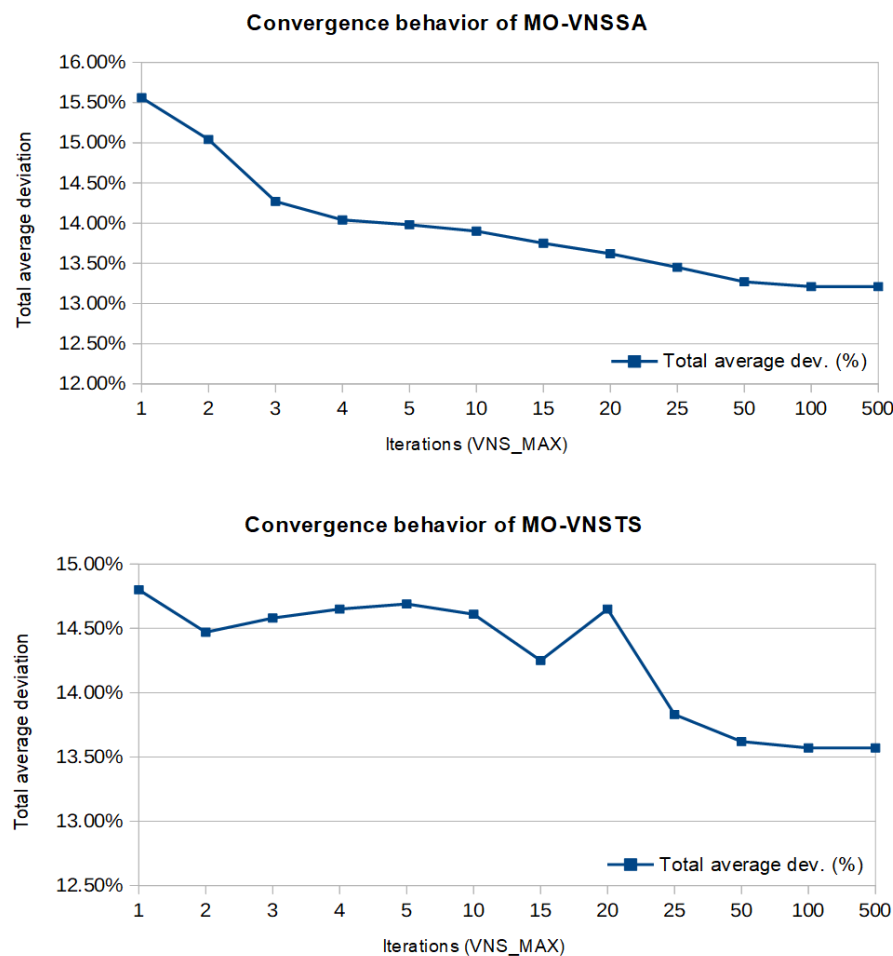


Figure 6. Convergence behavior of MO-VNSSA and MO-VNSTS for the second objective F_2 .

Taking those points into account, the proposed methods provide only solutions that are based on the decision maker's preferences and cannot provide a set of Pareto optimal solutions to be proposed to the decision maker. Moreover, the proposed meta-heuristics are not applicable to find an alternative solution in case of any changes in the relative importance of the objectives.

7. Conclusions

This paper addressed the multi-objective sustainable truck scheduling in the Road–Rail PI-hub cross-dock. The problem was formulated as a Multi-Objective Mixed Integer Programming model (MO-MIP) considering two different objectives. The first objective minimizes the energy consumption cost for the routing of the PI-containers using PI-conveyors. The second one minimizes the cost of using the outgoing trucks for each destination. The model was then solved using Lexicographic Goal Programming in CPLEX solver. Due to the long computational times, two Multi-Objective hybrid meta-heuristics were proposed: MO-VNSSA and MO-VNSTS. The MO-MIP and the two meta-heuristics were evaluated on 27 small instances and 5 additional large instances. CPLEX found optimal solutions for only 23 instances. The obtained results showed that the two meta-heuristics were able to generate near optimal and optimal solutions within short computational times. The results were validated through an ANOVA analysis.

In this study we used a lexicographic based method, which requires a priori knowledge from the decision maker. As an important direction of this work, we intend to extend our study by developing other Pareto based approaches that do not require a priori preferences about the relative importance of the objective functions. Those Pareto based approaches can be developed using population based

meta-heuristics such as NSGA, MOPSO and MOGA [75]. As another possible direction of this work, integrating external simulators could be interesting for taking into account the possible perturbations that can occur in the PI-hub facility such as trucks delays, customers changing orders at the last minute, etc.

Author Contributions: Conceptualization, T.C., A.B., M.R. and D.T.; data curation, T.C.; formal analysis, T.C.; funding acquisition, A.B., M.R. and D.T.; investigation, T.C.; methodology, T.C., A.B., M.R. and D.T.; project administration, A.B., M.R. and D.T.; software, T.C.; supervision, A.B., M.R. and D.T.; validation, T.C.; writing—original draft, T.C.; writing—review and editing, T.C., A.B., M.R. and D.T.

Acknowledgments: This research was supported by the ELSAT2020 project of CPER sponsored by the French ministry of sciences, the Hauts de France region and the FEDER. This work was supported also by the ANR PI-NUTS Project (grant ANR-14-CE27-0015).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACO	Ant Colony Optimization
GA	Genetic Algorithm
GLS	Guided Local Search
GRASP	Greedy Randomized Adaptive Search Procedure
LS	Local Search
MOGA	Multi-Objective Genetic Algorithm
MO-MIP	Multi-Objective Mixed Integer Programming Model
MOPSO	Multi-Objective Particle Swarm Optimization
MO-VNSSA	Multi-Objective Variable Neighborhood Search - Simulated Annealing
MO-VNSTS	Multi-Objective Variable Neighborhood Search - Tabu Search
NSGA	Non-dominated Sorting Genetic Algorithm
PI	Physical Internet
PMA	Pareto Memetic Algorithm
PSO	Particle Swarm Optimization
SA	Simulated Annealing
SCBM	Simulated Constraint Boundary Method
TL	Tabu List
TS	Tabu Search
VNS	Variable Neighborhood Search

References

1. Stock, J.R.; Boyer, S.L.; Harmon, T. Research opportunities in supply chain management. *J. Acad. Mark. Sci.* **2010**, *38*, 32–41. [\[CrossRef\]](#)
2. Christopher, M. *Logistics & Supply Chain Management*; Pearson: London, UK, 2016.
3. Croom, S.; Vidal, N.; Spetic, W.; Marshall, D.; McCarthy, L. Impact of social sustainability orientation and supply chain practices on operational performance. *Int. J. Oper. Prod. Manag.* **2018**, *38*, 2344–2366. [\[CrossRef\]](#)
4. Ciccullo, F.; Pero, M.; Caridi, M.; Gosling, J.; Purvis, L. Integrating the environmental and social sustainability pillars into the lean and agile supply chain management paradigms: A literature review and future research directions. *J. Clean. Prod.* **2018**, *172*, 2336–2350. [\[CrossRef\]](#)
5. Centobelli, P.; Cerchione, R.; Esposito, E. Environmental Sustainability and Energy-Efficient Supply Chain Management: A Review of Research Trends and Proposed Guidelines. *Energies* **2018**, *11*, 275. [\[CrossRef\]](#)
6. Koberg, E.; Longoni, A. A systematic review of sustainable supply chain management in global supply chains. *J. Clean. Prod.* **2019**, *207*, 1084–1098. [\[CrossRef\]](#)
7. Jadhav, A.; Orr, S.; Malik, M. The role of supply chain orientation in achieving supply chain sustainability. *Int. J. Prod. Econ.* **2018**. [\[CrossRef\]](#)
8. Ladier, A.L.; Alpan, G. Cross-docking operations: Current research versus industry practice. *Omega* **2016**, *62*, 145–162. [\[CrossRef\]](#)

9. Van Belle, J.; Valckenaers, P.; Cattrysse, D. Cross-docking: State of the art. *Omega* **2012**, *40*, 827–846. [[CrossRef](#)]
10. Montreuil, B.; Meller, R.D.; Ballot, E. Towards a Physical Internet: The impact on logistics facilities and material handling systems design and innovation. In Proceedings of the International Material Handling Research Colloquium (IMHRC), Milwaukee, WI, USA, 21–24 June 2010; pp. 1–23.
11. Montreuil, B.; Meller, R.D.; Ballot, E. Physical Internet Foundations. In *Service Orientation in Holonic and Multi Agent Manufacturing and Robotics*; Studies in Computational Intelligence; Borangiu, T., Thomas, A., Trentesaux, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; Volume 472, pp. 151–166.
12. Pan, S.; Nigrelli, M.; Ballot, E.; Sarraj, R.; Yang, Y. Perspectives of inventory control models in the Physical Internet: A simulation study. *Comput. Ind. Eng.* **2015**, *84*, 122–132. [[CrossRef](#)]
13. Treiblmaier, H.; Mirkovski, K.; Lowry, P. Conceptualizing the Physical Internet: Literature Review, Implications and Directions for Future Research. In Proceedings of the 11th CSCMP Annual European Research Seminar Conference, Vienna, Austria, 12–13 May 2016.
14. Sternberg, H.; Norrman, A. The Physical Internet—review, analysis and future research agenda. *Int. J. Phys. Distrib. Logist. Manag.* **2017**, *47*, 736–762. [[CrossRef](#)]
15. Meller, R.D.; Montreuil, B.; Thivierge, C.; Montreuil, Z. Functional design of physical internet facilities: A road-based transit center. In *12th IMHRC Proceedings, Gardanne, France*; Georgia Southern University: Statesboro, GA, USA, 2012.
16. Ballot, E.; Montreuil, B.; Thivierge, C. Functional design of physical internet facilities: A road–rail hub. In *12th IMHRC Proceedings, Gardanne, France*; Georgia Southern University: Statesboro, GA, USA, 2012.
17. Boysen, N.; Fließ, M. Cross dock scheduling: Classification, literature review and research agenda. *Omega* **2010**, *38*, 413–422. [[CrossRef](#)]
18. Yu, W.; Egbelu, P.J. Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. *Eur. J. Oper. Res.* **2008**, *184*, 377–396. [[CrossRef](#)]
19. Khalili-Damghani, K.; Tavana, M.; Santos-Arteaga, F.J.; Ghanbarzad-Dashti, M. A customized genetic algorithm for solving multi-period cross-dock truck scheduling problems. *Measurement* **2017**, *108*, 101–118. [[CrossRef](#)]
20. Keshtzari, M.; Naderi, B.; Mehdizadeh, E. An improved mathematical model and a hybrid metaheuristic for truck scheduling in cross-dock problems. *Comput. Ind. Eng.* **2016**, *91*, 197–204. [[CrossRef](#)]
21. Vahdani, B.; Zandieh, M. Scheduling trucks in cross-docking systems: Robust meta-heuristics. *Comput. Ind. Eng.* **2010**, *58*, 12–24. [[CrossRef](#)]
22. Boloori-Arabani, A.; Ghomi, S.F.; Zandieh, M. Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. *Expert Syst. Appl.* **2011**, *38*, 1964–1979. [[CrossRef](#)]
23. Amini, A.; Tavakkoli-Moghaddam, R. A bi-objective truck scheduling problem in a cross-docking center with probability of breakdown for trucks. *Comput. Ind. Eng.* **2016**, *96*, 180–191. [[CrossRef](#)]
24. Shakeri, M.; Low, M.Y.H.; Turner, S.J.; Lee, E.W. A robust two-phase heuristic algorithm for the truck scheduling problem in a resource-constrained crossdock. *Comput. Oper. Res.* **2012**, *39*, 2564–2577. [[CrossRef](#)]
25. Konur, D.; Golias, M.M. Cost-stable truck scheduling at a cross-dock facility with unknown truck arrivals: A meta-heuristic approach. *Transp. Res. Part E Logist. Transp. Rev.* **2013**, *49*, 71–91. [[CrossRef](#)]
26. Agustina, D.; Lee, C.; Piplani, R. A review: Mathematical models for cross docking planning. *Int. J. Eng. Bus. Manag.* **2010**, *2*, 47–54. [[CrossRef](#)]
27. Gelareh, S.; Glover, F.; Guemri, O.; Hanafi, S.; Nduwayo, P.; Todosijević, R. A comparative study of formulations for a cross-dock door assignment problem. *Omega* **2018**, in press. [[CrossRef](#)]
28. Assadi, M.T.; Bagheri, M. Differential evolution and Population-based simulated annealing for truck scheduling problem in multiple door cross-docking systems. *Comput. Ind. Eng.* **2016**, *96*, 149–161. [[CrossRef](#)]
29. Molavi, D.; Shahmardan, A.; Sajadieh, M.S. Truck scheduling in a cross docking systems with fixed due dates and shipment sorting. *Comput. Ind. Eng.* **2018**, *117*, 29–40. [[CrossRef](#)]
30. Dulebenets, M. A diploid evolutionary algorithm for sustainable truck scheduling at a cross-docking facility. *Sustainability* **2018**, *10*, 1333. [[CrossRef](#)]
31. Wisittipanich, W.; Hengmeechai, P. Truck scheduling in multi-door cross docking terminal by modified particle swarm optimization. *Comput. Ind. Eng.* **2017**, *113*, 793–802. [[CrossRef](#)]
32. Van Belle, J.; Valckenaers, P.; Berghe, G.V.; Cattrysse, D. A tabu search approach to the truck scheduling problem with multiple docks and time windows. *Comput. Ind. Eng.* **2013**, *66*, 818–826. [[CrossRef](#)]

33. Kuo, Y. Optimizing truck sequencing and truck dock assignment in a cross docking system. *Expert Syst. Appl.* **2013**, *40*, 5532–5541. [\[CrossRef\]](#)
34. Boysen, N.; Briskorn, D.; Tschöke, M. Truck scheduling in cross-docking terminals with fixed outbound departures. *OR Spectr.* **2013**, *35*, 479–504. [\[CrossRef\]](#)
35. Golias, M.M.; Saharidis, G.K.; Boile, M.; Theofanis, S. Scheduling of inbound trucks at a cross-docking facility: Bi-objective vs bi-level modeling approaches. *Int. J. Inf. Syst. Supply Chain Manag.* **2012**, *5*, 20–37. [\[CrossRef\]](#)
36. Mohtashami, A.; Tavana, M.; Santos-Arteaga, F.J.; Fallahian-Najafabadi, A. A novel multi-objective meta-heuristic model for solving cross-docking scheduling problems. *Appl. Soft Comput.* **2015**, *31*, 30–47. [\[CrossRef\]](#)
37. Heidari, F.; Zegordi, S.H.; Tavakkoli-Moghaddam, R. Modeling truck scheduling problem at a cross-dock facility through a bi-objective bi-level optimization approach. *J. Intell. Manuf.* **2018**, *29*, 1155–1170. [\[CrossRef\]](#)
38. Pach, C.; Sallez, Y.; Berger, T.; Bonte, T.; Trentesaux, D.; Montreuil, B. Routing Management in Physical Internet Crossdocking Hubs: Study of Grouping Strategies for Truck Loading. In *Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World*; IFIP Advances in Information and Communication Technology; Grabot, B., Vallespir, B., Gomes, S., Bouras, A., Kiritsis, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; Volume 438, pp. 483–490.
39. Sallez, Y.; Pan, S.; Montreuil, B.; Berger, T.; Ballot, E. On the activeness of intelligent Physical Internet containers. *Comput. Ind.* **2016**, *81*, 96–104. [\[CrossRef\]](#)
40. Sallez, Y.; Berger, T.; Bonte, T.; Trentesaux, D. Proposition of a hybrid control architecture for the routing in a Physical Internet cross-docking hub. *IFAC-PapersOnLine* **2015**, *48*, 1978–1983. [\[CrossRef\]](#)
41. Vo, N.V.; Berger, T.; Bonte, T.; Sallez, Y. Control of Rail–Road PI-Hub: The ORCA Hybrid Control Architecture. In *Service Orientation in Holonic and Multi-Agent Manufacturing*; Studies in Computational Intelligence; Borangiu, T., Trentesaux, D., Thomas, A., Cardin, O., Eds.; Springer International Publishing: New York, NY, USA, 2018; Volume 762, pp. 291–302.
42. Walha, F.; Bekrar, A.; Chaabane, S.; Loukil, T.M. A Rail–Road PI-hub allocation problem: Active and reactive approaches. *Comput. Ind.* **2016**, *81*, 138–151. [\[CrossRef\]](#)
43. Walha, F.; Bekrar, A.; Chaabane, S.; Loukil, T. A Rail–Road PI-hub allocation problems: Model and heuristic. In Proceedings of the IPIC 2014, Quebec, QC, Canada, 28–30 May 2014.
44. Chargui, T.; Bekrar, A.; Reghioui, M.; Trentesaux, D. A Mathematical Formulation and Tabu Search Approach for the Road–Rail Assignment Problem. In Proceedings of the 5th IPIC Conference 2018, Groningen, The Netherlands, 18–22 June 2018.
45. Taylor, K.M.; Vachon, S. Empirical research on sustainable supply chains: IJPR's contribution and research avenues. *Int. J. Prod. Res.* **2018**, *56*, 950–959. [\[CrossRef\]](#)
46. Mota, B.; Gomes, M.I.; Carvalho, A.; Barbosa-Povoa, A.P. Towards supply chain sustainability: Economic, environmental and social design and planning. *J. Clean. Prod.* **2015**, *105*, 14–27. [\[CrossRef\]](#)
47. Kong, L.; Li, H.; Luo, H.; Ding, L.; Zhang, X. Sustainable performance of just-in-time (JIT) management in time-dependent batch delivery scheduling of precast construction. *J. Clean. Prod.* **2018**, *193*, 684–701. [\[CrossRef\]](#)
48. Demartini, M.; Pinna, C.; Aliakbarian, B.; Tonelli, F.; Terzi, S. Soft Drink Supply Chain Sustainability: A Case Based Approach to Identify and Explain Best Practices and Key Performance Indicators. *Sustainability* **2018**, *10*, 3540. [\[CrossRef\]](#)
49. Guo, F.; Liu, Q.; Liu, D.; Guo, Z. On Production and Green Transportation Coordination in a Sustainable Global Supply Chain. *Sustainability* **2017**, *9*, 2071. [\[CrossRef\]](#)
50. Shahram fard, S.; Vahdani, B. Assignment and scheduling trucks in cross-docking system with energy consumption consideration and trucks queuing. *J. Clean. Prod.* **2019**, *213*, 21–41. [\[CrossRef\]](#)
51. Marler, R.T.; Arora, J.S. Survey of multi-objective optimization methods for engineering. *Struct. Multidiscip. Optim.* **2004**, *26*, 369–395. [\[CrossRef\]](#)
52. Duarte, A.; Pantrigo, J.J.; Pardo, E.G.; Mladenovic, N. Multi-objective variable neighborhood search: An application to combinatorial optimization problems. *J. Glob. Optim.* **2015**, *63*, 515–536. [\[CrossRef\]](#)
53. Glover, F. Tabu search—Part 1. *ORSA J. Comput.* **1989**, *1*, 190–206. [\[CrossRef\]](#)
54. Feo, T.A.; Resende, M.G. Greedy randomized adaptive search procedures. *J. Glob. Optim.* **1995**, *6*, 109–133. [\[CrossRef\]](#)

55. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [[CrossRef](#)] [[PubMed](#)]
56. Mladenović, N.; Hansen, P. Variable neighborhood search. *Comput. Oper. Res.* **1997**, *24*, 1097–1100. [[CrossRef](#)]
57. Voudouris, C. Guided Local Search for Combinatorial Optimisation Problems. Ph.D. Thesis, University of Essex, Colchester, UK, 1997.
58. Davis, L. *Handbook of Genetic Algorithms*; Van Nostrand Reinhold: New York, NY, USA, 1991.
59. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Western Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. doi:10.1109/ICNN.1995.488968. [[CrossRef](#)]
60. Dorigo, M.; Blum, C. Ant colony optimization theory: A survey. *Theor. Comput. Sci.* **2005**, *344*, 243–278. [[CrossRef](#)]
61. Konak, A.; Coit, D.W.; Smith, A.E. Multi-objective optimization using genetic algorithms: A tutorial. *Reliab. Eng. Syst. Saf.* **2006**, *91*, 992–1007. [[CrossRef](#)]
62. Parsopoulos, K.E.; Vrahatis, M.N. Particle swarm optimization method in multiobjective problems. In Proceedings of the 2002 ACM Symposium on Applied Computing, Madrid, Spain, 11–14 March 2002; pp. 603–607.
63. Jones, D.F.; Mirrazavi, S.K.; Tamiz, M. Multi-objective meta-heuristics: An overview of the current state-of-the-art. *Eur. J. Oper. Res.* **2002**, *137*, 1–9. [[CrossRef](#)]
64. Jaeggi, D.M.; Parks, G.T.; Kipouros, T.; Clarkson, P.J. The development of a multi-objective Tabu Search algorithm for continuous optimisation problems. *Eur. J. Oper. Res.* **2008**, *185*, 1192–1212. [[CrossRef](#)]
65. Bandyopadhyay, S.; Saha, S.; Maulik, U.; Deb, K. A simulated annealing-based multiobjective optimization algorithm: AMOSA. *IEEE Trans. Evol. Comput.* **2008**, *12*, 269–283. [[CrossRef](#)]
66. Geiger, M.J. Randomised variable neighbourhood search for multi objective optimisation. In Proceedings of the EU/ME Workshop: Design and Evaluation of Advanced Hybrid Meta-heuristics, Nottingham, UK, 4–5 November 2004; pp. 34–42.
67. Arroyo, J.E.C.; dos Santos Ottoni, R.; de Paiva Oliveira, A. Multi-objective variable neighborhood search algorithms for a single machine scheduling problem with distinct due windows. *Electron. Notes Theor. Comput. Sci.* **2011**, *281*, 5–19. [[CrossRef](#)]
68. Lu, P.; Tolliver, D. Multiobjective pavement-preservation decision making with simulated constraint boundary programming. *J. Transp. Eng.* **2013**, *139*, 880–888. [[CrossRef](#)]
69. Jaszkiwicz, A. A Comparative Study of Multiple-Objective Metaheuristics on the Bi-Objective Set Covering Problem and the Pareto Memetic Algorithm. *Ann. Oper. Res.* **2004**, *131*, 135–158. [[CrossRef](#)]
70. Sun, Y.; Li, X.; Liang, X.; Zhang, C. A Bi-Objective Fuzzy Credibilistic Chance-Constrained Programming Approach for the Hazardous Materials Road–Rail Multimodal Routing Problem under Uncertainty and Sustainability. *Sustainability* **2019**, *11*, 2577. [[CrossRef](#)]
71. Jones, D.; Tamiz, M. A review of goal programming. In *Multiple Criteria Decision Analysis*; Springer: New York, NY, USA, 2016; pp. 903–926.
72. Gandibleux, X.; Ehrgott, M. 1984–2004—20 Years of Multiobjective Metaheuristics. But What About the Solution of Combinatorial Problems with Multiple Objectives? In *Evolutionary Multi-Criterion Optimization*; Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 33–46.
73. Romero, C. *Handbook of Critical Issues in Goal Programming*; Elsevier: Amsterdam, The Netherlands, 2014.
74. Palacios, J.J.; González-Rodríguez, I.; Vela, C.R.; Puente, J. Swarm lexicographic goal programming for fuzzy open shop scheduling. *J. Intell. Manuf.* **2015**, *26*, 1201–1215. [[CrossRef](#)]
75. Ngatchou, P.; Zarei, A.; El-Sharkawi, A. Pareto multi objective optimization. In Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems, Arlington, VA, USA, 6–10 November 2005; pp. 84–91.

