HAL
open science

An Artificial-Immune-System-Based Algorithm Enhanced with Deep Reinforcement Learning for Solving Returnable Transport Item Problems

Fatima Ezzahra Achamrah, Fouad Riane, Evren Sahin, Sabine Limbourg

To cite this version:

Fatima Ezzahra Achamrah, Fouad Riane, Evren Sahin, Sabine Limbourg. An Artificial-Immune-System-Based Algorithm Enhanced with Deep Reinforcement Learning for Solving Returnable Transport Item Problems. Sustainability, 2022, 14 (10), pp.5805. 10.3390/su14105805 . hal-03668656

HAL Id: hal-03668656

https://hal.science/hal-03668656v1

Submitted on 6 Dec 2023

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.

sustainability

MDPI

Article

An Artificial-Immune-System-Based Algorithm Enhanced with Deep Reinforcement Learning for Solving Returnable Transport Item Problems

Fatima Ezzahra Achamrah [1,2], Fouad Riane [1,2,3], Evren Sahin [1] and Sabine Limbourg [4,*]

[1] Laboratoire Genie Industriel, CentraleSupelec, Paris Saclay University, 3 Rue Joliot-Curie, 91192 Gif-sur-Yvette, France; fatimaezzahra.achamrah@centrale-casablanca.ma (F.E.A); fouad.riane@centrale-casablanca.ma (F.R.); evren.sahin@centralesupelec.fr (E.S.)
[2] Complex Systems and Interations, Ecole Centrale of Casablanca, Ville Verte, Bouskoura 27182, Morocco
[3] Laboratoire Ingénierie, Management Industriel et Innovation (LIMII), Hassan First University, 577 Route de Casablanca, Settat 26000, Morocco
[4] HEC Management School, University of Liege, 14 Rue Louvrex, 4000 Liege, Belgium
* Correspondence: sabine.limbourg@uliege.be

Abstract: This paper proposes a new approach, i.e., virtual pooling, for optimising returnable transport item (RTI) flows in a two-level closed-loop supply chain. The supply chain comprises a set of suppliers delivering their products loaded on RTIs to a set of customers. RTIs are of various types. The objective is to model a deterministic, multi-supplier, multi-customer inventory routing problem with pickup and delivery of multi-RTI. The model includes inventory-level constraints, the availability of empty RTIs to suppliers, and the minimisation of the total cost, including inventory holding, screening, maintenance, transportation, sharing, and purchasing costs for new RTIs. Furthermore, suppliers with common customers coordinate to virtually pool their inventory of empty RTIs held by customers so that, when loaded RTIs are delivered to customers, each may benefit from this visit to pick up the empty RTI, regardless of the ownership. To handle the combinatorial complexity of the model, a new artificial-immune-system-based algorithm coupled with deep reinforcement learning is proposed. The algorithm combines artificial immune systems' strong global search ability and a strong self-adaptability ability into a goal-driven performance enhanced by deep reinforcement learning, all tailored to the suggested mathematical model. Computational experiments on randomly generated instances highlight the performance of the proposed approach. From a managerial point of view, the results stress that this new approach allows for economies of scale and cost reduction at the level of all involved parties to about 40%. In addition, a sensitivity analysis on the unit cost of transportation and the procurement of new RTIs is conducted, highlighting the benefits and limits of the proposed model compared to dedicated and physical pooling modes.

Keywords: closed loop supply chain; returnable transport items; pickup and delivery; inventory routing problem; artificial immune systems; deep reinforcement learning

check for updates

Citation: Achamrah, F.E.; Riane, F.; Sahin, E.; Limbourg, S. An Artificial-Immune-System-Based Algorithm Enhanced with Deep Reinforcement Learning for Solving Returnable Transport Item Problems. *Sustainability* **2022**, *14*, 5805. https://doi.org/10.3390/su14105805

Academic Editors: Lu Zhen, Junliang He and Lufei Huang

Received: 24 March 2022
Accepted: 2 May 2022
Published: 11 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Returnable transport items (RTIs) are all reusable assets used to facilitate product shipping, storing, handling, and protection in the supply chain [1]. RTIs cover reusable drums, pallets, crates, rolls, boxes, and barrels [2–4]. Along with globalised supply chains, the use of RTIs has become more popular in recent decades as they eliminate the wastes that one-way secondary packaging may generate [5]. The use of RTIs has been proved to be an enabler for better ergonomics and productivity while facilitating automation, better inventory control, and improved quality [3,4,6,7]. Furthermore, their operational benefits help reduce the disposal costs of packaging material and improve productivity [8]. These assets usually flow in a closed-loop supply chain between players [5,9]. Loaded RTIs are

received and unloaded at a given level of the supply chain. Either the empty RTI can be collected and returned to the sender or the receiver can reuse them to ship his products and thus continue to flow downstream the supply chain. Therefore, there exist two flows of RTIs that must be managed [10]: forward flows, which correspond to the forward distribution of goods loaded on RTIs, and reverse flows, which correspond to the collection and return of empty RTIs to their owners. This paper aims to optimise both forward and reverse flows of RTIs in a two-level closed supply chain.

Managing such assets has become a primary concern of supply chain managers, along with managing warehouses, machines, and vehicles [7,9]. Indeed, it has become very pressing for companies to effectively package products and guarantee to have them in the proper quantity, at the right place, and at the right time. To avoid shortages, many companies frequently tend to invest in more RTIs, resulting in higher holding and purchasing costs [3,11]. Moreover, supply chain players experience RTI losses with rates varying from 3 to 20% [12]. This mismanagement lengthens turnaround times and pushes players to overinvest in these assets, leading to inefficient budgetary practices: companies buy new RTIs to replace the lost ones and recruit additional staff to handle them [2,9,12].

According to [7,13,14], RTI management can be divided into two modes depending on the ownership of empty RTIs: a dedicated mode (private RTIs) and a shared mode (public RTIs). The dedicated mode (DM) refers to the case where RTIs are owned by players (suppliers, for example) who use them exclusively to deliver their products without considering sharing them with others. They are responsible, in general, for collecting, refurbishing, and managing the inventory of their specific assets. In this system, RTIs received by a partner are shipped back to their specific owner. In the shared mode (SM), players agree to share their RTIs within a "pooled" system. A service provider company manages this shared system, and running such a pool is its core business [15]. In this pool, empty RTIs are physically stored and can be used by all players without any obligation for these assets to return to their starting point at their next movement [15]. RTI pools can be categorised into two types: "rented" and "open" pools [15]. The "rented" pool is based on a one-owner pool model: RTIs are owned by one company that rents and provides the supply chain players with the empty RTIs they need. In this case, the company manages and oversees its RTI pool's day-to-day operations and services. The "open" pool is based on a changing-owner pool model: all partners store their RTIs in a pool, and when an RTI is used, its ownership is transferred to the receiving partner, who must return similar RTIs of comparable quality (1:1 exchange concept). In both cases, a pooling system involves a pooler responsible for supplying ready-to-use RTIs to all partners, collecting them from downstream levels, refurbishing damaged ones, and holding inventory within its facilities until new RTI orders are placed [16].

The literature review (see Section 2) shows that most papers exclusively address DM and SM and highlight each mode's benefits on the overall supply chain performance. However, both modes may not always be profitable and practicable. Compared to the SM, the DM may be easier to implement and it does not lead to resource dependency, as each player is always free to manage and use his inventory of empty RTIs [7]. On the other hand, the SM is typically less expensive, as it may offer cost benefits through the shared use of RTIs among tier suppliers [13,14]. However, the prerequisites of commonly serviceable RTIs for various materials from several suppliers are hard to meet [13]. Moreover, the SM compels advanced decision-making on where to locate pooler facilities, how to set facilities' capacities, and how to distribute transportation flows (i.e., delivery, pickup, inventory balancing, and supply) across the network, which may imply additional managerial costs (i.e., transportation, inventory holding) and a need for solid information system support [17]. The SM may also establish a resource dependency, as each player is not always free to pick up the empty RTIs needed to deliver his products. This is particularly true for complex supply chains, which include multiple origins and destinations and multiple RTIs that flow within, and in which constraints such as variable demands, vehicle capacity, and shortage are to be considered. This paper proposes a new approach to overcome the shortcomings

above of both modes in a closed-loop supply chain. Specifically, we consider the case of a two-level closed-loop supply chain comprising a set of suppliers delivering products to joint customers. We assume that each supplier owns RTIs that can be held in either his or the customer's inventory. In addition, each supplier is responsible, as in DM, for collecting, refurbishing, and managing his inventory. We also assume that the suppliers coordinate their logistics operations so that, while delivering loaded RTIs to customers, each supplier may benefit from this visit to pick up empty RTIs regardless of the ownership. This has earmarks of the classic lateral transshipment that relies on authorising the virtual pooling of finished products' inventory between members belonging to the same echelon of supply chains [18]. This practice usually takes place to re-balance the entire system's stock levels to react to scenarios where one of the locations faces a shortage while others have residual stock in hand. Accordingly, instead of calling upon a pooler or a leasing company to acquire the needed quantities of RTIs, this paper suggests that suppliers arrange to "virtually" pool/share their stock of identically substitutable RTIs: no need for a real and physical pool to store RTIs as in SM. As such, we conserve the ownership of RTIs as in the DM and allow the shared use of RTIs as in SM (Table 1). Moreover, each supplier buys, when needed, and adds new RTIs to the whole system. Therefore, the order may be filled, the customers receive what they want, and the partners free up space in their inventory and reduce idle stock. It is mutually beneficial for all parties. Consequently, suppliers can sidestep the shortage of empty RTIs at their levels and reduce the cost of transportation, inventory holding, and the procurement of new RTIs. Such a strategy creates a valuable partnership but implies additional logistics operations that must be optimised.

Table 1. Characteristics of RTIs management strategies.

	DM	SM	Virtual Pooling Mode
Owner of RTIs	Each supplier	All suppliers or a pooler company	Each supplier
Management of empty RTIs, collection, refurbishing…	Each supplier	One pooler company	All suppliers
Storage of empty and shared RTIs	-	In dedicated facilities	At suppliers' level

Our paper has three main contributions. First, we develop a new mathematical formulation of the RTI pickup and delivery problem in a closed-loop supply chain consisting of a set of suppliers shipping their products to a set of common customers (e.g., plants, retailers) and using a set of RTIs, i.e., a multi-supplier multi-customer inventory routing problem with the pickup and delivery of multi-type shared RTIs (IRPPDS). We assume that supply chain partners adopt a vendor-managed inventory policy (VMI): their operations are coordinated to organise deliveries and pickups to fulfil customers' demands. Thus, we address a multi-supplier, multi-customer, multi-RTI inventory routing problem that is hard to solve due to its inherent combinatorial complexity. Suggesting an efficient way to cope with this complexity by developing a breakthrough solving approach is the second contribution of this paper. Indeed, we use a matheuristic that hybridises an artificial-immune-system-based metaheuristic and a mathematical programming algorithm. Furthermore, thanks to its generality and flexibility, this matheuristic uses deep reinforcement learning techniques that were initially proposed by [19] for solving dynamic and stochastic inventory routing problems successfully. Furthermore, the performance of the approach is compared to the one developed in [19] and to two pure metaheuristics. Finally, broad experiment campaigns are conducted on instances of large sizes. These experiments stress that the resolution approach is very competitive compared to other existent metaheuristics: it leads to better quality solutions and reduces computational time. Furthermore, we evaluate the cost reduction enabled by the virtual pooling of RTIs compared to DM and SM.

The remainder of the paper is organised as follows. Section 2 presents an overview of related works. After a detailed definition of the problem in Section 3, the mathematical formulation is provided in Section 4. Section 5 describes the proposed resolution approach and explains the hybridisation scheme used to integrate the mathematical model, the artificial-immune-system-based algorithm and the deep reinforcement learning technique. Section 6 provides the computational results and presents the matheuristic performance analysis compared to three resolution approaches. Finally, Section 7 summarises the main findings and provides perspectives for further research.

2. Related Work

This section reviews research streams that are mostly related to our work. The objective is to position our contributions in line with papers on the inventory routing problem (IRP) with pickup and delivery and RTIs management modes and highlight our contribution to the resolution approaches applied to solve similar problems.

The vehicle routing problem (VRP) calls for determining the optimal set of routes to be performed by a fleet of vehicles to serve a given set of customers [20]. In the literature, three different variants related to the structure of pickup and delivery and the number of origins and destinations are to be distinguished [21]: one-to-one (1-1), in which a request is originated at one location and destined for another location; one-to-many-to-one (1-M-1), in which each customer receives a delivery originating from a common depot and sends a pickup quantity to the depot; and finally, many-to-many (M-M), in which a commodity may be picked up at one of many locations and also delivered to one of many locations [22–25]. The IRP calls for inventory management, vehicle routing, and delivery scheduling decision-making problems [26]. Our paper's most relevant research stream addresses IRP with pickup and delivery (IRPPD). According to [27], this problem has three variations regarding vehicle routing: (1) VRP with simultaneous pickup and delivery (SPD), in which products are delivered whilst others are simultaneously sent back to the origin; (2) VRP with backhauls, where all deliveries must be undertaken before any pickup on each route; (3) VRP with mixed pickup and delivery, which can be characterised as a particular case of the VRP with SPD in which customers may have pickup or delivery demands. Some recent applications of the VRP/IRP with pickups and deliveries can be found in [4,7,28–31]. IRP problems have been intensively studied in the literature, and the reader is referred to [26] for a thorough overview of more related papers. Furthermore, for the more recent papers on decision support models for RTIs, the reader is referred to the review by [5,32], which provides a systematic literature review of decision models in managing closed-loop supply chains, including RTIs. Along with developing decision support models, significant research efforts have also been devoted to investigating RTI management strategiesin both the dedicated and the shared modes [14]. Most related works address the management of RTIs as part of a VMI policy and develop decision support models for cost reduction under stochastic or deterministic environments for the dedicated mode. Applications can be found in [1,2,14]. In [4,33,34], the authors propose models for inventory routing problems with simultaneous pickups and deliveries for a single-supplier, single-RTI, multi-customer (1-M-1) closed-loop supply chain. The models consider the maintenance costs of the reused RTIs returned from customers and the cost of buying a new one. In [33], other scheduled pickups and the supply of new RTIs are integrated as alternatives to sidestep the shortage of empty RTIs at the supplier level. Finally, in [11], a decentralised two-stage supply chain with a Retailer Stackelberg game is studied. The authors develop an analytical model to determine lot-sizing and pricing decisions for the product and its secondary packaging. As for the shared mode, most related work has studied different scenarios for the pooling or rental of RTIs with the help of mathematical modelling and simulation. The authors of [35] investigate a lot-sizing problem and assignment strategy that minimises the pallet management cost under environmental constraints. The authors of [36] study the pallet allocation problem under stochastic supply scenarios and customer priority, while those of [6] study a fresh fruit and vegetable supply chain and develop a mathematical model to

select the best packaging (reusable/disposable) and minimise holding and handling costs. The authors of [37] analyse the effects of pallet service conditions and repair facilityon the economic and environmental performance of a pallet pooling system. In their paper, a new RTI procurement decision is also taken into consideration. The authors of [38] analyse the reverse logistics of plastic pallets in Canada, focusing on recovery options, such as reusing, remanufacturing, and recycling. A mathematical model is developed to determine the best locations in a pallet reverse logistics network and optimise the distribution flows between the network players. The authors of [16] analyse the transportation operations of a pallet pooling company serving a set of retailers. A pooler company is assumed to be responsible for supplying, collecting, and refurbishing pallets. Buying/selling and pooling management strategies are assessed and compared through what-if analysis. The authors of [39] study the service centres' location problem considering a pallet pool mode. By integrating the forward and reverse flow of pallets, the objective is to minimise the total cost, including fixed construction, inventory, delivery, and recovery costs. The authors of [7] develop a mixed-integer program model for planning the distribution and vehicle routing for a single type of RTI and in a single period. They consider a pooler company responsible for dispatching leased empty containers to its customers and collecting the customers' surplus empty containers. In their model, minimising procurement, storage, and maintenance costs is not considered. The authors of [40] use a simulation-based approach to model sharing a single RTI between two producers in a closed-loop supply chain. The results show that collaboration can lead to economies of scale and cost reduction. They also highlight the need for a third party to manage the entire system to promise mutual benefits for the concerned parties. On the other hand, the routing decisions are not optimised in their simulation model. Moreover, the model is not generic and realistic, as it considers a simple supply chain and only one type of RTI that flows in.

As for combinatorial complexity, VRP/IRP with pickup and delivery problems are well known to be NP-hard [41,42]. To tackle this complexity, approximation algorithms or metaheuristics are used. The most commonly encountered metaheuristics are either stochastic algorithms such as simulated annealing (SA) or ones based on artificial intelligence algorithms such as artificial immune system (AIS), genetic algorithm (GA), particle swarm optimisation (PSO) and ant colony (AC). Though AIS-based algorithms are a relatively new complex-problem-solving approach compared to other metaheuristics, the inherent characteristics of the immune memory, vaccination process, and self-recognition ability of the antibody and the diversity of immunity allow it to have a high level of flexibility and a good balance between global and local search [43]. Furthermore, AIS has demonstrated efficiency in convergence compared to other algorithms for large instances. The authors of [44–49] reported that AIS has a higher convergence rate than GA, PSO, AC, and SA. Therefore, AIS is used to solve our model for large-sized instances for all the reasons above.

To further enhance the convergence speed of AIS, we use machine learning (ML) techniques. Indeed, metaheuristics, through their iterative search processes, generate a lot of data that can be turned into explicit knowledge if coupled with ML models. This data concern decision-making solutions and the objective spaces visited during the search process, the sequence of solutions or trajectories, successive populations of solutions, moves, recombination, local optima, elite solutions, and bad solutions [50]. ML techniques can help analyse this data, extract valuable knowledge, and enhance metaheuristics' search performance. Thus, metasearch techniques become "data-driven", "well informed", and therefore "smarter". In this respect, ML was used to address discrete optimisation problems that focus on the travelling salesman problem and VRP. The data-driven metaheuristics have been proven to be advantageous in convergence speed, solution quality, and robustness. The methodologies in ML for decision problems, typically addressed by operation research (OR), are mainly found in reinforcement learning (RL), learning to search, and multi-armed bandits. The authors of [19,51–55] illustrate the recent successes achieved by RL concerning problems typically addressed by OR. For instance, the authors of [19] develop a matheuristic enhanced by RL techniques to solve a dynamic and stochastic IRP. The authors of [56,57]

introduce ML in the solution processes of inventory and location problems. Finally, the authors of [58–60] use an RL-based technique to solve a VRP. As far as we are concerned, our paper is the first that combines RL with AIS to solve a multi-supplier multi-customer multi-RTI IRP with pickup and delivery in a closed-loop supply chain [61,62].

This review shows that despite the extensive literature on RTIs related to IRP, there is a lack of efficient tools and techniques to solve complex combinatorial problems such as closed-loop multi-product, multi-period, inventory routing problems with deliveries and pickups of multiple types of RTIs. As already mentioned, our research makes three main scientific contributions. Firstly, we develop a mathematical model to address the deterministic, multi-supplier, multi-customer (M-M), inventory routing problem, considering the delivery and return flows of multiple RTIs which are virtually pooled between a given number of suppliers. Secondly, we use a new artificial-immune-system-based algorithm and combine its strong global search capability with RL's strong self-adaptability and goal-driven performance, all tailored to the mathematical model. Thirdly, computational experiments on specially designed instances highlight the performance of the proposed algorithm. From a managerial point of view, the results stress that this new approach allows for economies of scale and cost reduction at the level of all the involved parties. Furthermore, a sensitivity analysis on unit cost and the procurement of new RTIs is conducted and highlights the benefits and limits of the proposed model compared to other RTI management modes.

3. Mathematical Formulation

This section presents the mathematical models developed for IRPPDS, DM, and SM.

3.1. Mathematical Model for IRPPDS

We examine a multi-supplier, multi-customer, multi-RTI closed-loop supply chain. A set of m suppliers distribute different types of products using a set of r types of RTIs to a set of n common customers over a finite planning horizon. Each supplier delivers RTIs loaded with products to a set of customers. Each customer uses these products in his production process and constitutes an inventory of empty RTIs. The supplier then collects those empty RTIs to be reused for future productions and deliveries at his level. We assume that all supply chain players adopt a centralised management policy to synchronise operations according to each player's requirements, optimise deliveries and pickups, and meet customers' expectations.

The planning horizon is defined by a discrete and finite set of periods (days). Each player has a storage zone separated into two areas: an area for the inventory of empty RTIs (E) and another for the inventory of loaded RTIs (L). Each of these inventory areas is characterised by an initial inventory level and a maximum holding capacity. Initial inventories of loaded and empty RTIs are supposed to be positive and known at the beginning of the planning horizon. Deliveries and pickups are carried out by a set of homogeneous fleets of vehicles. Each vehicle can transport loaded or empty RTIs, or both, with a determined capacity in terms of the number of RTIs without distinction between empty and loaded RTIs (foldable RTIs are not considered). It is assumed that each constructed route starts from a supplier to visit a set of customers, and there is no route built between suppliers. Furthermore, customers are visited by each supplier independently of other suppliers' planned routes. Since vehicles have a limited capacity, multiple suppliers' routes are allowed. We assume that a vehicle can perform at most one pickup and delivery per period, all routes start and finish at each supplier, and split pickup/deliveries are not allowed. In each period, the sequence of events is as follows. First, each supplier prepares the quantity of loaded RTIs to be shipped by considering the current inventory. He uses his empty RTIs and those of other suppliers to load products on the appropriate type of RTIs. Then, each supplier visits each customer in each period to deliver the required quantity of products (in terms of loaded RTIs) for his production. The available inventory of empty RTIs at each level of the supply chain is checked. Depending on the demand that he must satisfy in the next period, each supplier picks up empty RTIs belonging to him. If these are not sufficient, he

picks up other RTIs belonging to the other suppliers as long as these latter have sufficient inventory to meet demands for the next period. After pickups are performed, the empty RTIs are subject to quality control at each supplier location. Damaged RTIs are disposed of, serviceable RTIs are repaired, and undamaged RTIs are transferred to the inventory of empty RTIs. All the RTIs present in the inventory (repaired/cleaned) at the end of each period can be reused in the next period. Moreover, we assume that, in addition to the virtual pooling of empty RTIs, a supplier can purchase empty RTIs that he may need to fulfil future demands. In this case, buying RTIs is permitted in each period, and each RTI is available for use in the following one.

The objective of the IRPPDS model is to determine, for each level of the supply chain and over the finite planning horizon, the quantity of loaded RTIs to be delivered by each supplier and the quantity of empty RTIs to be picked up by each customer and shared. The demand is supposed to be deterministic by being time-varying. Such planning considers the inventory-level constraints (no shortages, backlogs, or overstocking are allowed), the availability of empty RTIs to suppliers, and the minimisation of the total cost, including inventory holding, maintenance, transportation, sharing, and the purchasing of new RTIs.

To model IRPPDS, we introduce different notations. We consider: a set $N = i|i = 1, \ldots n$ of n customers; a set $P = 0_p|0_p = 0_1, \ldots 0_m$ of m suppliers; a set $N_p = i|i = 0_p, 1 \ldots n$ that represents the n customers, and the node 0_p that represents the supplier p; a set $R = r|r = 1, \ldots u$ of u types of RTIs that are used to carry on different types of products; and a set $V = v|v = 1, \ldots k$ of k homogeneous vehicles with a capacity of Q in terms of the number of RTIs. Accordingly, loaded and empty RTIs occupy the same volume as in the case of boxes and containers. We also consider a horizon $T = t|t = 1, \ldots l$ of l periods. Each supplier p and customer i incurs a holding cost for loaded RTIs (L) and empty RTIs (E): $H_p^{L,r}, h_i^{L,r}, H_p^{E,r}$ and $h_i^{E,r}$ (€ per unit), respectively. $I_{p0}^{L,r}, L_{i0}^{L,r}, I_{p0}^{E,r}$ and $L_{i0}^{E,r}$ represent the initial inventory level of loaded and empty RTI of type r, respectively, at the supplier p and customer i. C_p^L, c_i^L, C_p^E and c_i^E represent the maximum holding capacity for loaded and empty RTI, respectively, for the supplier p and customer i. At the beginning of the planning horizon, each supplier p receives information on demand to satisfy D_{pit}^r (expressed in terms of loaded RTIs) of each customer $i \in N$ for each period $t \in T$ and for each RTI r. The distance between actors $i \in N_p$ and $j \in N_p$ is denoted by d_{ij}^p. The fixed cost of transportation is represented by a in € per km, and b represents the variable cost of transportation in € per weight unit and per km. The weights of a loaded and empty RTI are w_L^r and w_E^r, respectively. The cost of buying an RTI is e_r in € per unit. The sharing cost incurred by each supplier p is s_r per unit of unowned empty RTIs of type r belonging to other suppliers p used at his level to deliver products. This cost represents the utilisation cost of an unowned RTI used by each supplier if it occurs. Finally, g_r is the maintenance cost per RTI of type r used by the suppliers to deliver products, including inspection and cleaning costs. The model's notation is summarised in Table 2.

Table 2. Model's notation summary.

	Sets
N	Set of n customers.
P	Set of m suppliers.
R	Set of u RTIs.
V	Set of k vehicles.
T	Set of l periods.

Table 2. *Cont.*

Parameters	
a	Fixed cost of transportation (€ per km).
b	Variable cost of transportation (€ per weight per km).
$H_p^{L,r}$, $h_i^{L,r}$, $H_p^{E,r}$, $H_i^{E,r}$	Cost of holding inventory of loaded and empty RTIs, respectively, for each supplier p and customer i.
e_r	Cost of buying a new RTI of type r (€ per unit).
s_r	Cost of sharing incurred by each supplier which is proportional to the number of unowned empty RTIs of a type r used at his level to deliver products (€ per unit of unowned RTI used).
g_r	Cost of maintenance of one RTI of type r (€ per RTI loaded).
w_L^r, w_E^r	Weights of a loaded and empty RTIs of type r, respectively.
Q	Capacity of vehicle in terms of the number of RTIs.
d_{ij}^p	Distance between nodes i and $j \in N_p$.
D_{pit}^r	Demand of each customer i for each period t loaded on an RTI r satisfied by supplier p.
$I_{p0}^{L,r}$, $L_{i0}^{L,r}$, $I_{p0}^{E,r}$, $L_{i0}^{E,r}$	Initial inventory level of loaded and empty RTIs of type r, respectively, for the supplier p and customer i.
C_p^L, c_i^L, C_p^E, c_i^E	Maximum holding capacity for loaded and empty RTIs, respectively, for the supplier p and customer i.
Decision variables	
x_{ijvt}^p	Binary variable stating whether the vehicle v visited node j immediately after node i in period t.
$F_{pt}^{p't}$	Quantity of empty RTIs of type r owned by supplier p that have been filled with products by supplier p in period t. This quantity also includes the case where $p = p'$ (supplier uses his RTI).
I_{Lr}^{pt}	Inventory level of loaded RTIs of type r held by the supplier p at the end of period t.
L_{pit}^{Lr}	Inventory level of RTIs of type r filled with the product of supplier p by customer i at the end of period t.
$Q_{pit}^{p'r}$	Quantity of loaded RTIs of type r owned by supplier p' and delivered by supplier p to customer i in period t.
X_{ijt}^{pr}	Quantity of loaded RTIs of type r filled with a product of supplier p transported from node i to node j in period t.
L_{it}^{Er}	Inventory level of empty RTIs of type r held by the customer i at the end of period t.
I_{pt}^{Er}	Total quantities of all empty RTIs of type r held by the supplier p at the end of period t.
Z_{it}^{pr}	Quantity of empty RTIs of type r owned by supplier p collected from a customer i in period t.
$W_{ip't}^{pr}$	Quantity of empty RTIs of type r owned by supplier p and collected from customer i by supplier p in period t.
E_{ijt}^{pr}	Quantity of empty RTIs of type r collected by supplier p transported from node i to node j in period t.
n_t^{pr}	Quantity of RTIs of type r bought by supplier p in period t.

The IRPPDS in a multi-supplier, multi-customer, multi-RTI closed-loop supply chain is modelled as follows:

$$\begin{aligned}
&\min \sum_{i\in N}\sum_{t\in T}\sum_{r\in R}(h_i^{Lr}L_{it}^{Lr} + h_i^{Er}L_{it}^{Er}) + \sum_{p\in P}\sum_{t\in T}\sum_{r\in R}(H_p^{L,r}I_{pt}^{Lr} + H_p^{Er}I_{pt}^{Er}) + \\
&\sum_{p\in P}\sum_{t\in T}\sum_{r\in R} e_r n_t^{p,r} + \sum_{p\in P}\sum_{t\in T}\sum_{p'\in P}\sum_{r\in R} g_r F_{pt}^{p'r} + \\
&\sum_{i\in N}\sum_{p\in P}\sum_{p'\in P}\sum_{t\in T}\sum_{r\in R} s_r W_{ip't}^{pr} + \\
&\sum_{p\in P}\sum_{t\in T}\sum_{i\in N_p}\sum_{j\in N_p}\left(a\sum_{v\in V} x_{ijvt}^p + \sum_{r\in R} b\,(w_L^r X_{ijt}^{pr} + w_E^r E_{ijt}^{pr})\, d_{ij}^p\right)
\end{aligned} \tag{1}$$

subject to:

$$L^{Lr}_{pit} = L^{Lr}_{pit-1} + \sum_{p' \in P} Q^{p'r}_{pit} - D^{r}_{pit} \qquad \forall i \in N, t \in T, p \in P, r \in R \tag{2}$$

$$I^{Lr}_{pt} = I^{Lr}_{pt-1} - \sum_{i \in N} \sum_{p' \in P} Q^{p'r}_{pit} + \sum_{p' \in P} F^{p'r}_{pt} \qquad \forall t \in T, p \in P, r \in R \tag{3}$$

$$L^{Er}_{it} = L^{Er}_{it-1} - \sum_{p \in P} Z^{pr}_{it} + \sum_{p \in P} D^{r}_{pit} - \sum_{p \in P} \sum_{p' \in P} W^{pr}_{ip't} \qquad \forall i \in N, t \in T, r \in R \tag{4}$$

$$I^{Er}_{pt} = I^{Er}_{pt-1} + \sum_{i \in N} Z^{pr}_{it} - \sum_{p' \in P} F^{p'r}_{pt} + n^{pr}_{t} + \sum_{p' \in P} W^{pr}_{ip't} \qquad \forall p \in P, t \in T, r \in R \tag{5}$$

$$\sum_{i \in N_p, i \neq j} (X^{pr}_{ijt} - X^{pr}_{jit}) = \sum_{p' \in P} Q^{p'r}_{pjt} \qquad \forall j \in N, p \in P, t \in T, r \in R \tag{6}$$

$$\sum_{i \in N_p, i \neq j} (E^{pr}_{jit} - E^{pr}_{ijt}) = Z^{pr}_{jt} + \sum_{p' \in P} W^{pr}_{jp't} \qquad \forall j \in N, p \in P, t \in T, r \in R \tag{7}$$

$$0 \leq \sum_{p \in P} \sum_{r \in R} L^{Lr}_{pit} \leq c^{L}_{i} \qquad \forall i \in N, t \in T \tag{8}$$

$$0 \leq \sum_{r \in R} I^{Lr}_{pt} \leq C^{L}_{p} \qquad \forall p \in P, t \in T \tag{9}$$

$$0 \leq \sum_{p \in P} \sum_{r \in R} L^{Er}_{pit} \leq c^{E}_{i} \qquad \forall i \in N, t \in T \tag{10}$$

$$0 \leq \sum_{r \in R} I^{Er}_{pt} \leq C^{E}_{p} \qquad \forall p \in P, t \in T \tag{11}$$

$$\sum_{p \in P} \sum_{r \in R} (X^{pr}_{ijt} + E^{pr}_{ijt}) \leq Q \sum_{p \in P} \sum_{v \in V} x^{p}_{ijvt} \qquad \forall i, j \in N_p, t \in T \tag{12}$$

$$\sum_{i \in N_p} \sum_{v \in V} x^{p}_{ijvt} \leq 1 \qquad \forall j \in N, p \in P, t \in T \tag{13}$$

$$\sum_{i \in N_p i \neq j} x^{p}_{ijvt} = \sum_{i \in N_p i \neq j} x^{p}_{jivt} \qquad \forall vs. \in V, j \in N_p, p \in P, t \in T \tag{14}$$

$$\sum_{j \in N} x^{p}_{0_p jvt} \leq 1 \qquad \forall vs. \in V, p \in P, t \in T \tag{15}$$

The objective function (1) minimises inventory costs at the level of each customer and supplier, the costs of purchasing new RTIs, the cost of the maintenance of RTIs, the sharing cost of RTIs undertaken by each supplier, and finally, the fixed and variable cost of transportation for pickup and delivery. Constraints (2) define the conditions for the conservation of the inventory levels of loaded RTIs owned by supplier p at the level of each customer i. Constraints (3) state that at the level of each supplier p, the inventory level of loaded RTIs at the end of period t is equal to the inventory level at the beginning of the period minus the quantities of loaded RTIs delivered to all customers and plus the quantities of empty RTIs that were loaded by supplier p in period t. Constraints (4) indicate that the inventory level for customer i at the end of period t of empty RTIs, held by supplier p, is equal to the inventory level of empty RTIs at the beginning of the period minus the quantity picked up by each supplier p plus the RTIs that have been emptied after demand has been satisfied minus the quantity of empty RTIs belonging to each supplier p that other suppliers have collected. Constraints (5) indicate that at the level of each supplier p, the inventory level of empty RTIs at the end of period t is equal to the inventory level at the beginning of the period plus the quantity of his empty RTIs collected from all customers plus the quantity of empty RTIs belonging to other suppliers that have been collected from customers by supplier p, minus the quantity of empty RTIs that have been loaded in period t plus the quantity of purchased RTIs. Constraints (6) ensure that the quantities of loaded RTIs owned by supplier p are delivered to customer j. Constraints (7) show that the flow of empty RTIs belonging to supplier p outgoing from node j is equal to the quantity of empty RTIs belonging to supplier p collected by supplier p, plus the quantity of empty RTIs belonging to other suppliers collected by supplier p, minus the inflow from all

customers. Constraints (8)–(11) indicate the boundaries of the inventory levels of loaded and empty RTIs at the level of each supplier p and customer i. Constraints (12) stipulate that the quantities delivered and collected between two nodes i and j must not exceed the capacity of the vehicles on the arc (i, j). Constraints (13)–(15) express the conditions for the construction of tours. Constraints (13) indicate that at most one vehicle is used to visit node j. Constraints (14) guarantee the continuity of a tour. Constraints (15) ensure that vehicles leave the supplier only once per period or remain at the depot. Finally, constraints that define the non-negative constraints and the binary nature of the decision variables are imposed.

3.2. Mathematical Model for DM

For the DM model, there is no pooling of empty RTIs between the suppliers. That is, $W^{pr}_{ip't} = F^{p'r}_{pt} = 0, if\ p' \neq p, \forall p, p' \in P, i \in N, t \in T, r \in R$. Each supplier manages, independently of other suppliers, the deliveries of his loaded RTIs to customers, the pickups of empty ones from customers, and their inventories at his level and customers' location. Accordingly, the mathematical model is solved for each supplier independently and costs to minimise include inventory holding of empty and loaded RTIs, the transportation cost for delivery and pickups, maintenance, and the procurement of new RTIs. Accordingly, the formulation of the DM model, $\forall p \in P$, is as follows:

$$\begin{aligned} &\min \sum_{i\in N}\sum_{t\in T}\sum_{r\in R}(h^{Lr}_i L^{Lr}_{it} + h^{Er}_i L^{Er}_{it}) + \sum_{t\in T}\sum_{r\in R}(H^{L,r}_p I^{Lr}_{pt} + H^{Er}_p I^{Er}_{pt}) + \\ &\sum_{t\in T}\sum_{r\in R} e_r n^{p,r}_t + \sum_{t\in T}\sum_{r\in R} g_r F^r_{pt} + \\ &\sum_{t\in T}\sum_{i\in N_p}\sum_{j\in N_p}\left(a\sum_{v\in V} x^p_{ijvt} + \sum_{r\in R} b\,(w^r_L X^{pr}_{ijt} + w^r_E E^{pr}_{ijt})\, d^p_{ij}\right) \end{aligned} \tag{16}$$

The objective function minimises inventory costs for the supplier p and each customer, the costs of purchasing new RTIs, the maintenance cost of RTIs, and finally, the fixed and variable transportation costs for pickup and delivery.

It is subject to:

$$L^{Lr}_{pit} = L^{Lr}_{pit-1} + Q^r_{pit} - D^r_{pit} \qquad \forall i \in N, t \in T, r \in R \tag{17}$$

$$I^{Lr}_{pt} = I^{Lr}_{pt-1} - \sum_{i\in N} Q^r_{pit} + F^r_{pt} \qquad \forall t \in T, r \in R \tag{18}$$

$$L^{Er}_{it} = L^{Er}_{it-1} - \sum_{p\in P} Z^{pr}_{it} + \sum_{p\in P} D^r_{pit} \qquad \forall i \in N, t \in T, r \in R \tag{19}$$

$$I^{Er}_{pt} = I^{Er}_{pt-1} + \sum_{i\in N} Z^{pr}_{it} - F^r_{pt} + n^{pr}_t \qquad \forall t \in T, r \in R \tag{20}$$

$$\sum_{i\in N_p, i\neq j} (X^{pr}_{ijt} - X^{pr}_{jit}) = Q^r_{pjt} \qquad \forall j \in N, t \in T, r \in R \tag{21}$$

$$\sum_{i\in N_p, i\neq j} (E^{pr}_{jit} - E^{pr}_{ijt}) = Z^{pr}_{jt} \qquad \forall j \in N, t \in T, r \in R \tag{22}$$

$$0 \leq \sum_{p\in P}\sum_{r\in R} L^{Lr}_{pit} \leq c^L_i \qquad \forall i \in N, t \in T \tag{23}$$

$$0 \leq \sum_{r\in R} I^{Lr}_{pt} \leq C^L_p \qquad \forall t \in T \tag{24}$$

$$0 \leq \sum_{p \in P} \sum_{r \in R} L_{pit}^{Er} \leq c_i^E \qquad \forall i \in N, t \in T \tag{25}$$

$$0 \leq \sum_{r \in R} I_{pt}^{Er} \leq C_p^E \qquad \forall t \in T \tag{26}$$

$$\sum_{p \in P} \sum_{r \in R} (X_{ijt}^{pr} + E_{ijt}^{pr}) \leq Q \sum_{p \in P} \sum_{v \in V} x_{ijvt}^p \qquad \forall i, j \in N_p, t \in T \tag{27}$$

$$\sum_{i \in N_p} \sum_{v \in V} x_{ijvt}^p \leq 1 \qquad \forall j \in N, t \in T \tag{28}$$

$$\sum_{i \in N_p i \neq j} x_{ijvt}^p = \sum_{i \in N_p i \neq j} x_{jivt}^p \qquad \forall vs. \in V, j \in N_p, t \in T \tag{29}$$

$$\sum_{j \in N} x_{0_p jvt}^p \leq 1 \qquad \forall vs. \in V, t \in T \tag{30}$$

with:

Q_{pit}^r: Quantity of loaded RTIs of type r owned by supplier p and that have been delivered to customer i in period t.
F_{pt}^r: Quantity of empty RTIs of type r owned by supplier p and that have been filled with products at his level in period t.

3.3. Mathematical Model for SM

In the SM model, a pooler company manages the inventory, pickups, and procurement of empty RTIs. On the other hand, each supplier is responsible for delivering loaded RTIs and managing their corresponding inventory. Furthermore, empty RTIs are delivered directly from customers to a series of centres (pooler facilities) managed by the company rather than to suppliers, as in the DM and IRPPDS models. The centres are assumed to be located near the suppliers. To determine the location of these centres, we solve a multi-period weighted clustering problem (MPC). The clustering consists in grouping supplier nodes into clusters to minimise the total distance between suppliers. Each cluster centroid of suppliers represents the centre in which empty RTIs of these suppliers are stored, cleaned, and repaired. When needed, the centre sends empty RTIs to suppliers so that they can produce and deliver their products to customers. As for costs, two other costs are to be considered: inventory holding at each centre ι and pooling cost. The latter incorporates the management of centres by the pooler company and each unowned RTIs used by each supplier (which is assumed, for the purposes of simplification, to be equivalent to the sharing cost in IRPPDS). The constraints of IRPPDS for the inventory and routing of pickups of empty RTIs from customers to centres and from the centres to suppliers are rewritten accordingly. In the following, the formulation of the SM model is presented.

3.3.1. Multi-Period Clustering Problem

To determine the location of the centres, we first solve an MPC. To do so, we define the binary variables $\theta_{p\iota}$ that have a value of 1 if a supplier p belongs to the cluster $\iota(\iota \in K = \{\iota | \ \iota = \iota, \ldots \kappa \leq m\})$, and 0 otherwise with a binary variable $\epsilon_{pp'\iota}$ having a value of 1 if the suppliers p and p' belong to the same cluster. MPC can be then modelled as follows:

$$\min \sum_{p,p' \in P} \sum_{\iota \in K} d_{pp'} \epsilon_{pp'\iota} \tag{31}$$

subject to:

$$\sum_{\iota \in K} \theta_{p\iota} = 1 \qquad \forall\, p \in P \tag{32}$$

$$\sum_{p \in P} \sum_{r \in R} \sum_{1 \le t' \le t} D^{r}_{pit'} \theta_{p\iota} \le tQ \qquad \forall\, \iota \in K, t \in T \tag{33}$$

$$\epsilon_{pp'\iota} \le \theta_{p\iota}\ ,\ \epsilon_{pp'\iota} \le \theta_{p'\iota} \qquad \forall \iota \in K, p \in P, p' \in P, p \ne p' \tag{34}$$

$$\epsilon_{pp'\iota} \ge \theta_{p\iota} + \theta_{p'\iota} - 1 \qquad \forall \iota \in K, p \in P, p' \in P,\ p \ne p' \tag{35}$$

$$\epsilon_{pp'\iota},\ \theta_{p\iota} \in \{0,1\} \qquad \forall \iota \in K, p \in P, p' \in P,\ p \ne p' \tag{36}$$

The objective function (31) is to minimise the distance between suppliers (p, p') belonging to the same cluster (ι). Constraints (32) ensure that each supplier is assigned to a unique cluster. Constraints (33) state that the aggregate quantity of empty RTIs in each cluster in terms of demands over the planning horizon must fit into the available capacity, tQ, where Q is the vehicle capacity. Constraints (34) and (35) state that the distance between suppliers p, p', and $d_{pp'}$ is included in the objective function if and only if suppliers p and p' are assigned to the same cluster. Constraints (36) define the binary nature of the decision variables.

3.3.2. SM Model

In the SM model, two costs are considered:

- Inventory holding at each centre $\iota : \sum_{\iota \in K} \sum_{t \in T} \sum_{r \in R} H^{Er}_{\iota} \mathcal{L}^{Er}_{\iota t}$;
- Pooling cost for each unowned RTI used by each supplier (which is equivalent to the sharing cost in IRPPDS).

The constraints of IRPPDS for the inventory and routing of pickups of empty RTIs from customers to centres and from the centres to suppliers are rewritten as follows. Conservation of inventory levels and flows of empty RTIs at the level of each supplier p, customer i, and centre ι (with $\theta_{p\iota}$ and $\epsilon_{pp'\iota}$ already determined by solving MPC):

$$L^{Er}_{it} = L^{Er}_{it-1} - \sum_{\iota \in K} \sum_{p \in P} \theta_{p\iota} Z^{pr}_{it} + \sum_{\iota \in K} \sum_{p \in P} \theta_{p\iota} D^{r}_{pit} \qquad \forall i \in N, t \in T, r \in R \tag{37}$$

$$I^{Er}_{pt} = I^{Er}_{pt-1} + \sum_{\iota \in K} \theta_{p\iota} R^{\iota r}_{pt} + \sum_{p' \in P} \sum_{i \in N} \sum_{\iota \in K} \epsilon_{pp'\iota} W^{pr}_{ip't} - \sum_{p' \in P} \sum_{\iota \in K} \epsilon_{pp'\iota} F^{p'r}_{pt} \qquad \forall p \in P, t \in T, r \in R \tag{38}$$

$$\mathcal{L}^{Er}_{\iota t} = \mathcal{L}^{Er}_{\iota t-1} + \sum_{i \in N} \sum_{p \in P} \theta_{p\iota} Z^{pr}_{it} - \sum_{p \in P} \theta_{p\iota} R^{\iota r}_{pt} - \sum_{p,p' \in P} \sum_{i \in N} \epsilon_{pp'\iota} W^{pr}_{ip't} - \sum_{p \in P} \sum_{i \in N} Z^{pr}_{it} + n^{\iota r}_{t}$$
$$\forall \iota \in K, t \in T, r \in R \tag{39}$$

$$\sum_{i \in N_{p\iota}, i \ne j} (E^{\iota r}_{jit} - E^{\iota r}_{ijt}) = Z^{pr}_{jt} \qquad \forall j \in N_{p\iota}, t \in T, r \in R, p \in P : \theta_{p\iota} = 1\ \iota \in K \tag{40}$$

$$\sum_{\substack{p' \in P \\ p' \ne p}} (E'^{\iota r}_{pp't} - E'^{\iota r}_{p'pt}) = \theta_{p\iota} R^{\iota r}_{pt} + \sum_{i \in N} \sum_{p' \in P} \epsilon_{pp'\iota} W^{pr}_{ip't} \qquad \forall p \in P, \iota \in K, t \in T, r \in R \tag{41}$$

$$0 \le \sum_{r \in R} \mathcal{L}^{Er}_{\iota t} \le c^{E}_{\iota} \qquad \forall\, \iota \in K, t \in T \tag{42}$$

$$\sum_{p \in P} \sum_{r \in R} X^{p,r}_{ijt} \le Q \sum_{p \in P} \sum_{v \in V} x^{p}_{ijvt} \qquad \forall i, j \in N_p, i \ne j, t \in T \tag{43}$$

$$\sum_{r \in R} E^{\iota,r}_{ijt} \le Q \sum_{v \in V} y^{\iota}_{ijvt} \qquad \forall i, j \in N_{p\iota}, i \ne j, \iota \in K, t \in T \tag{44}$$

$$\sum_{r \in R} E'^{\iota,r}_{pp't} \le Q \sum_{v \in V} y^{\iota}_{pp'vs.t} \qquad \forall t \in T, p,\ p' \in P, p \ne p', \iota \in K : \epsilon_{pp'\iota} = 1 \tag{45}$$

$$\sum_{i \in N_{p\iota}} \sum_{v \in V} y^{\iota}_{ijvt} \le 1 \qquad \forall j \in N, t \in T, \iota \in K \tag{46}$$

$$\sum_{i \in N_{p\iota} i \neq j} y^{\iota}_{ijvt} = \sum_{i \in N_{p\iota} i \neq j} x^{\iota}_{jivt} \qquad \forall vs. \in V, j \in N_{p\iota}, t \in T \tag{47}$$

$$\sum_{j \in N} y^{\iota}_{0_\iota jvt} \leq 1 \qquad \forall v \in V, t \in T, \iota \in K \tag{48}$$

$$\sum_{p \in P_{p\iota} \cup 0_\iota} \sum_{v \in V} y^{\iota}_{pp'vs.t} \leq 1 \qquad \forall p' \in P_{p\iota}, t \in T, \iota \in K \tag{49}$$

$$\sum_{p' \in P_{p\iota} \cup 0_\iota p' \neq p} y^{\iota}_{p'pvt} = \sum_{p' \in P_{p\iota} \cup 0_\iota p' \neq p} y^{\iota}_{pp'vt} \qquad \forall vs. \in V, p \in P_{p\iota}, t \in T \tag{50}$$

$$\sum_{p \in P_{p\iota}} y^{\iota}_{0_\iota pvt} \leq 1 \qquad \forall v \in V, t \in T, \iota \in K \tag{51}$$

with:
$N_{p\iota}$: set of customers for whom the supplier p belongs to the cluster of centre ι (node 0_ι).
$P_{p\iota}$: set of suppliers belonging to the cluster of centre ι.
$R^{\iota,r}_{pt}$: quantity of empty RTIs of type r belonging to supplier p and sent to centre ι to which supplier p belongs.
$E^{\iota,r}_{ijt}$: quantity of empty RTIs of type r transported from node i to node j in period t and sent to centre ι.
$E'^{\iota,r}_{p'pt}$: quantity of empty RTIs of type r transported from node p to node p' in period t and sent by centre ι.
$\mathcal{L}^{E,r}_{\iota t}$: inventory level of empty RTIs of type r at centre ι in period t.
y^{ι}_{ijvt}: binary variable equal to 1 if node j is visited right after node i by vehicle v, 0 otherwise.
$y^{\iota}_{0_\iota jvt}$: binary variable equal to 1 if customer j is visited by v from node (cluster) 0_ι, 0 otherwise.
$y^{\iota}_{pp'vs.t}$: binary variable equal to 1 if supplier p' is visited right after supplier p by vehicle v, 0 otherwise.
$y^{\iota}_{0_\iota pvt}$ binary variable equal to 1 if supplier p is visited by v from node (cluster) 0_ι, 0 otherwise.

4. Resolution Approach

The DM, SM, and IRPPDS models described in the previous section are NP-hard. To tackle their combinatorial complexity, a resolution approach is proposed.

We aim at determining over a given planning horizon the required quantities of RTIs to allow for supplying the needed quantities of products from a set of suppliers to a set of customers. We also seek to construct the optimal routes for pickups and deliveries of RTIs. Since the construction of the routes is the most complex part of the problem, we first use an appropriate heuristic to determine those routes. Once constructed, we solve a modified version of the three MILPs described in Section 3 to determine the other decision variables related, for example, to the quantities transported, delivered, and collected. Each of these versions is a min-cost network flow problem that is easier to solve. Regarding IRPPDS, its modified version is called FMILP, where the routing decision variables, x^{p}_{ijvt}, are fixed:
FMILP:

$$\begin{aligned} \min & \sum_{i \in N} \sum_{t \in T} \sum_{r \in R} (h^{Lr}_i L^{Lr}_{it} + h^{Er}_i L^{Er}_{it}) + \sum_{p \in P} \sum_{t \in T} \sum_{r \in R} (H^{Lr}_p I^{Lr}_{pt} + H^{Er}_p I^{Er}_{pt}) + \\ & \sum_{p \in P} \sum_{t \in T} \sum_{r \in R} e_r n^{p,r}_t + \sum_{p \in P} \sum_{t \in T} \sum_{p' \in P} \sum_{r \in R} g_r F^{p'r}_{pt} + \\ & \sum_{i \in N} \sum_{p \in P} \sum_{p' \in P} \sum_{t \in T} \sum_{r \in R} s_r W^{pr}_{ip't} \end{aligned} \tag{52}$$

subject to Constraints (2)–(11).

We use a matheuristic to construct routes and improve the final solution as described above. The matheuristic hybridises the FMILP with an artificial-immune-system-based algorithm and a deep Q-learning process into a global solving scheme called AIS-DQL. The overview of the matheuristic AIS-DQL is presented in Figure 1.

These steps are described in detail in the following subsections.

4.1. Artificial Immune System

Artificial-immune-system-based algorithms are bio-inspired metaheuristics that imitate the principles and processes of immune system functioning [63]. The algorithms are typically modelled after the immune system's characteristics of learning and memory for use in problem solving. They imitate antigen recognition, antigen and antibody binding, and the antibody production process. Furthermore, they abstractly use the diversity and memory mechanism of the immune system. Therefore, they can ensure individual diversity while maintaining a high affinity, thereby avoiding premature phenomena and showing a strong global search ability. In this paper, antigens correspond to the input data of the problem, and the antibodies correspond to the routes to construct or the different suppliers. Their structure, depicted in Figure 2, consists of sequences of possible nodes to be visited in each route and for each supplier and each period.

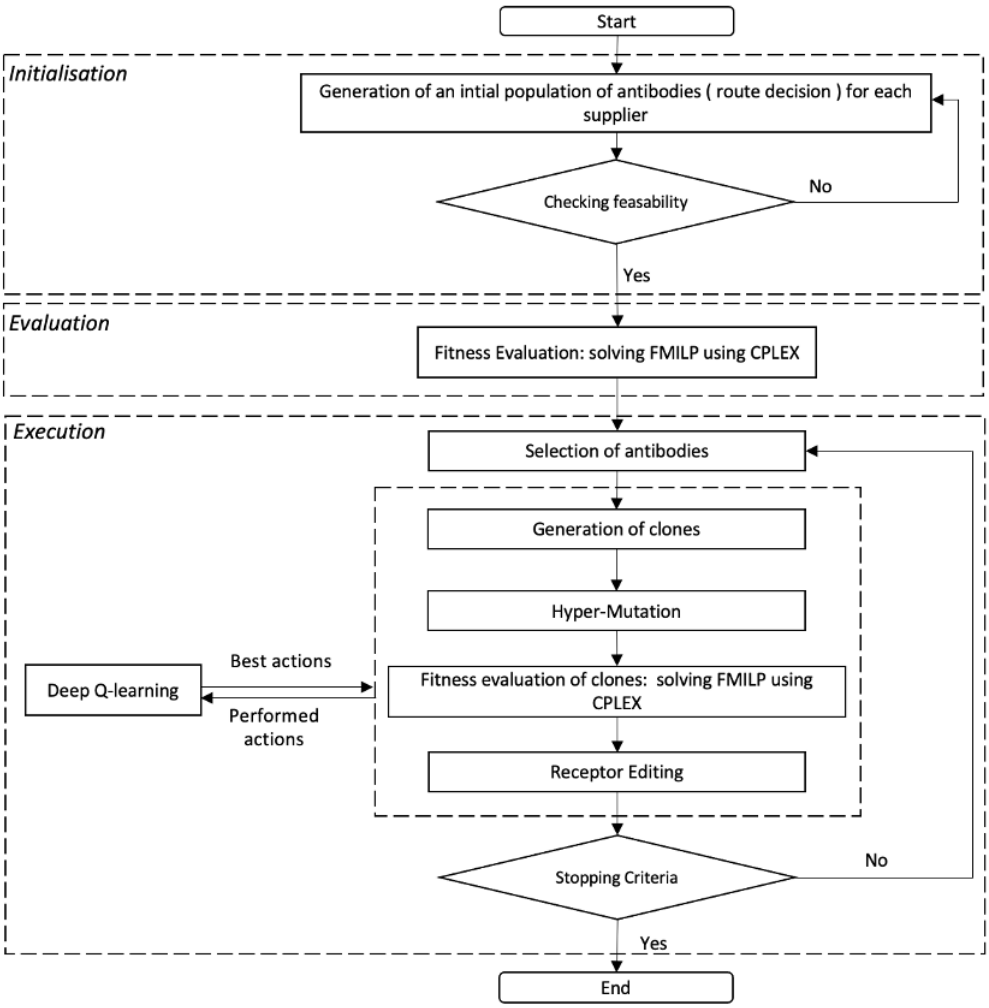

Figure 1. Overview of the implementation of the AIS-DQL matheuristic.

Supplier 1	Customer 1	Customer 2	----	Customer n-1	Customer n
Supplier 2	Customer 3	Customer 10	----	Customer n-10	Customer n-5
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
Supplier m	Customer n-2	Customer n	----	Customer 1	Customer 2

Figure 2. Structure representing the route of Supplier i (antibodies).

As depicted in Figure 1, AIS starts with an initialisation phase. A population of random routing solutions representing a pool of antibodies (routes) is initially generated. The routes are built using a 2-opt local search algorithm [64]. Proliferation and maturation processes are undergone by cloning each member of the initial pool, i.e., copying each of the initial solutions based on their affinity. The rate of proliferation is chosen to be directly proportional to the affinity, such that the higher the affinity, the more offspring there are. For this purpose, selection, hypermutation (HM), and receptor editing (RE) operators are used.

4.1.1. Affinity and Cloning Selection

Each time an antibody (routing decisions) is generated, it is used as an input to solve the FMILP. Therefore, the corresponding feasible objective function (OF) and the remaining decision variables of the model are computed. The affinity f_ι of an antibody, ι, is computed using the corresponding objective function OF_ι: $f_\iota = \frac{1}{OF_\iota}$. Thus, the higher affinity value would have a lower total cost. Hence, as an antibody's cloning rate is proportional to its affinity, more antibody clones have lower costs in the next generation than antibodies with higher costs. The probability, PS, of selecting an antibody to be cloned depends on its affinity. If f_ι is the affinity of an antibody ι in the population, its probability PS_ι is defined as: $PS_\iota = \frac{f_\iota}{\sum_\varsigma f_\varsigma}$.

4.1.2. Affinity Maturation

Since the algorithm needs to thoroughly explore and exploit the search space to obtain a good solution, exploration and exploitation are carried out depending on the evolution operator's capability variation. These operators conduct random perturbations on each gene to generate the next generation's population in the current population. The variation in the antibodies is performed through HM and RE mechanisms. The HM mechanism ensures that the higher-affinity antibodies are hyper-mutated at a slower rate. The HM_ι the rate for an antibody ι is defined as $HM_\iota = e^{-\omega f_\iota}$, where ω is the decay control factor. A new population is created after hyper-mutation, and each antibody undergoes various affinity changes. Antibodies are therefore reorganised once again based on the affinity assessment.

After cloning and mutation processes, a percentage of the antibodies in the current population is eliminated (the worst ϕ% of the population) and replaced by the randomly generated antibodies. This mechanism, which is a vertebrate immune system mechanism, is called receptor editing [44]. This mechanism generates new antibodies that correspond to the new search area of the search space. Exploring new search areas may help the algorithm to escape from local optima. The new antibody population then becomes the next generation of antibodies.

Finally, if a generation's objective function value does not improve over that of the previous generation, convergence is assumed to be achieved, and it is possible to retrieve the best equivalent antibody as the best solution, and the algorithm stops.

4.2. AIS Enhanced with Deep Q-Learning

In this section, we highlight AIS limitations and present an RL technique used to overcome them.

4.2.1. AIS and RL

According to [43], although many results have proved the convergence of AISs to a global optimum, a Markov-chain analysis shows weak convergence of the AIS algorithms. Indeed, due to the single-point random mutation of the antibody, AIS converges slowly, meaning that a given antibody selects a gene bit and changes its value randomly to some of the other selectable values. Moreover, it cannot retain any locally excellent gene blocks in some low-affinity antibodies because of other poor gene blocks. As a result, the search speed is low. From this stems the idea of using RL to tackle this problem. Indeed, since random searching leads to slow evolution and weak AIS convergence efficiency, environmental feedback signals and the updated action policy of deep Q-learning are used to construct an algorithm with strong self-adaptability and goal-driven performance.

In this paper, RL is employed to assist in analysing data on moves and recombination that have been performed to construct solutions to the problem. The goal is to extract meaningful information from this data to direct and improve the AIS's search performance and speed. Indeed, just like a human being, the agent that symbolises the antibody (solution to the problem) learns of its own to acquire successful strategies that result in the largest long-term rewards. RL is a paradigm of learning by trial and error based entirely on rewards or penalties. The agent constructs and learns its information directly from moves it makes using operators such as HM and RE. RL is indeed used to assist AIS in determining the optimal actions to take in terms of the best moves for each operator.

4.2.2. Q-Learning

Q-Learning is a self-adaptive RL off-policy method characterised by strong environmental feedback signals [65]. The fundamental idea is to use the feedback signal to adjust an agent's action policy to make the best decision when interacting with the environment (i.e., antibody space). The agent (i.e., antibody) arrives in different states based on actions (i.e., AIS operators). Actions determine positive and negative rewards. The concept behind Q-learning is to put the agent in a series of state-action combinations, observe the rewards, and then change the predictions of a table (called a Q-table) to those rewards until the best policy properly predicts them. As a result, the "Q" stands for quality, which indicates how effective a particular action is in earning a possible reward.

4.2.3. Deep Q-Learning

Q-learning is a relatively basic and effective algorithm. However, it may be time-consuming, as the amount of memory required to save and update the Q-table grows with the number of states, and the amount of time required to investigate each state to construct the appropriate Q-table is impracticable. In this paper, these Q-values are estimated using neural networks known as deep Q-learning (DQ). Accordingly, the state is an input, and the output is the Q-value of all potential actions. Once the network is trained, selecting the right action means comparing each action's possible rewards and choosing the best one.

4.2.4. Deep Q-Learning Architecture

DQ begins by estimating random Q-values to explore the environment, as shown in Figure 3. DQ enhances its Q-value estimations by employing the same dual-action paradigm, with a present action having a current Q-predicted value and a target action with a target Q-value. The direction of the predicted Q-target values varies since the network and its weights are equal; they remain unchanged but may fluctuate with each update. The Q-target values are stabilised by employing a second network that has not been trained. After a pre-determined number of iterations, referred to as C-iteration, the learned weights from the Q-predicted network are copied to the Q-target network. The DQ design has two

neural networks (Q-predicted and Q-target) and an experience replay agent, as shown in Figure 3. For data generation during Q-network training, the experience replay interacts with the environment. These data contain all of AIS's operators' moves, which are recorded as <st,a,$\mathscr{R}$,st'> tuples (see notation below Equation (53)). Then, a sample is picked randomly from these data such that it consists of a mix of older and more recent samples. This batch of training data is used in the Q-predicted and Q-target networks. The Q-predicted network takes the current state and moves out of each sample, and for that move predicts the Q-value. The Q-predicted value, the Q-target value, and the observed data sample reward are used to compute the loss for the Q-network training (see Equation (53)). A batch of data is selected from all prior experiences to reduce variance and guarantee the algorithm's stability in C-iteration. Next, a sample is chosen from these data, with an equal mix of older and newer samples. The Q-predicted and Q-target networks use this batch of training data. The Q-predicted network predicts the Q-value based on the current state and moves out of each sample. The loss for the Q network training is computed using the Q-predicted value, the Q-target value, and the observed data sample reward (see Equation (53)). After each C-iteration, a batch of data is selected from all prior experiences to reduce variation and ensure the algorithm's stability.

$$Loss = [\mathscr{R}_{t+1} + \gamma \max_{a}(\theta^T Q(st', a') - \theta^T Q(st, a))]^2 \tag{53}$$

where:

- γ: discount-rate parameter to measure the weight of the future awards.
- a, a': current and future action, respectively.
- st, st': current and future state, respectively.
- $\mathscr{R}_{t+1}$: future reward.
- $Q(st, a)$: learned action-value function.
- θ^T: transpose matrix of network weights.

Finally, as for the AIS memory, a set of the best antibodies having the highest affinity is stored, and the best moves have been obtained so far. Instead of starting from scratch every time the algorithm is run to solve the model for a given antigen, similar to the antigens (instances) already solved, we use the genetic memory to rapidly obtain the best solutions and the optimal policies for the antibodies. Similar antigens are selected based on the K-nearest neighbours algorithm [66].

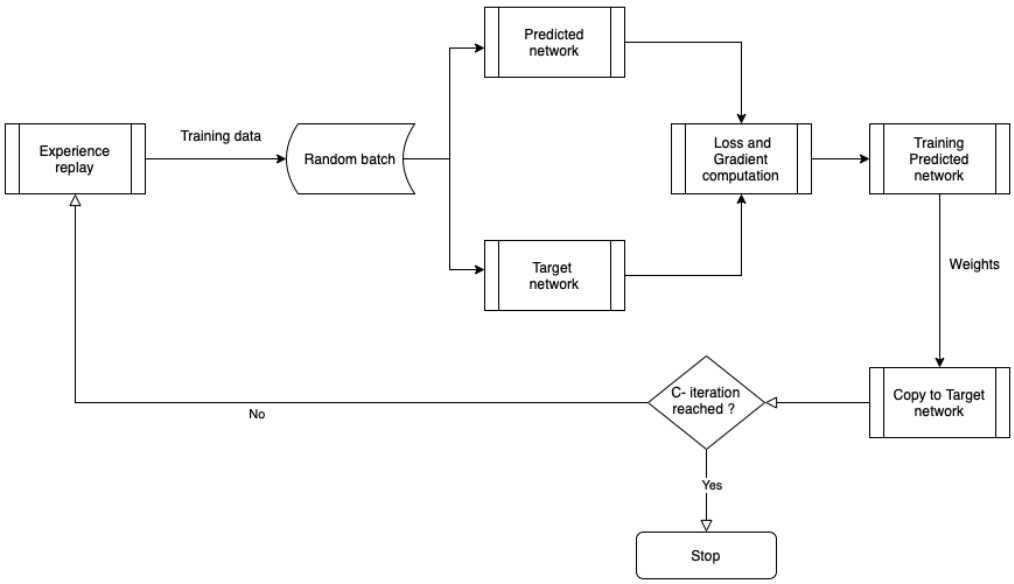

Figure 3. DQ architecture adapted from [19].

5. Implementation and Experimental Analysis

This section presents the experimental design adopted for this study and the analysis of the computational results. All the optimisation steps were carried out on a personal computer (MacBook Pro, macOS Cataline, CPU 3.3 GHz Quad-Core Intel Core i7, 8 GB of RAM).

Experimental Design and Parameters Tuning

The MILP developed for the multi-supplier, multi-customer, multi-RTI closed-loop supply chain was first solved to optimality for small-and-medium-size instances using the Branch-and-Cut solver of CPLEX 12.9 (academic version). The objective was to check the model's validity, representativeness, and exact solving approach limitations.

To implement the matheuristic, we used Python 3.7 and Pytorch interfaced with CPLEX. The approach was first tested on the same instances optimally solved using CPLEX. The objective was to assess its performance. We ran the AIS algorithm without the learning process (AIS) and compared the improvement provided by the deep Q-learning when coupled with the AIS algorithm (AIS-DQL). We also compared the algorithm's performance with a pure genetic algorithm (GA) and its improved learning version (GA-DQL). GA is also a population-based metaheuristic that mimics the principle of natural genetics to find a solution. The algorithm is known for its strong global search. The algorithm starts with an initial set of random solutions called a population. Each individual in the population is called a chromosome, representing a solution to the problem at hand. The best parents (best chromosomes having the highest affinity) are selected from the current generation and considered for a two-point crossover operation to form their offspring. The mutation process is also integrated as it helps to obtain new information randomly for the genetic search process and ultimately helps avoid getting trapped at local optima. In this paper, the chromosomes also represent the routing decisions and are decoded as the antibodies of AIS. For a thorough description of GA-DQL, the reader is referred to [19].

The tests were performed in 20 replications for the 40 generated instances to evaluate the stability of the algorithms, and the average value of the objective function is presented. A statistical analysis using ANOVA was also conducted to assess the eventual randomness of the differences between the obtained results (see Table 3). These results stress that for all resolution approaches under consideration, p-value > 0.05, which means that there is no significant difference between the algorithms and the solutions obtained using CPLEX. Table 4 reports the algorithm parameters tuned so that a trade-off between the algorithm's performance and speed is satisfied.

Table 3. Statistical analysis using ANOVA.

Resolution Approach	F	p-Value
GA	2.16	0.14
AIS	1.57	0.22
GA-DQL	0.91	0.34
AIS-DQL	0.49	0.49

Table 4. Values of the tuning parameters.

Tuned Parameter	Value
Population size (GA/AIS)	200
Maximum iteration number (GA/AIS)	200
Crossover probability (GA/AIS)	0.81
Mutation probability (GA/AIS)	0.46
Selection probability	0.80
Receptor editing rate	0.28

The instances had a number of suppliers varying from 5 to 25, a number of customers from 6 to 24, and a number of RTIs varying from 2 to 10. The planning horizon of de-

liveries and pickups was five days, corresponding to a workweek. Customer demands were randomly generated between 5 and 70 in loaded RTIs. For each instance, suppliers' and customers' locations were randomly chosen in the Euclidean space between (0, 0) and (1000, 1000). Moreover, we considered initial inventory levels and unit costs for transportation, holding and maintenance of the self-same scale as in [4], which considers a 1-M-1 IRP for a single type of RTIs. As the unit cost of an RTI may go from a few euros for plastic boxes to 1300 euros for stillages, according to the study conducted by [3], we considered a randomly generated purchase cost varying between 3 and 1000 euros. Finally, we considered a unit cost of sharing varying between 2 and 10 euros per type of RTI.

In the remainder of the paper, we refer to the instances using the following notation: (number of RTIs) R, (number of suppliers) S, (number of customers) C, (number of vehicles) V, and (number of periods) T; e.g., 1R2S5C2V5T refers to the instance where one type of RTI is shared and used to ship the products of two suppliers to a set of five customers, transported by two vehicles over a planning horizon of 5 days.

6. Computational Experiments

First, the three models, developed for the DM, SM, and IRPPDS modes, were solved using CPLEX. The objective was to compare the benefits and limitations of each mode on the performance of the overall supply chain. Then, given their combinatorial complexity, the three models were solved using four approaches: AIS and GA with and without DQL. The performance of each of these approaches was analysed by comparing it to the solutions obtained using CPLEX on small instances. The benefits of DQL on the performance of the methods were also highlighted. Given the contribution of DQL, the three models were solved on large instances using the AIS-DQL and GA-DQL approaches.

6.1. Results on Small Instances Solved Using CPLEX

The SM, DM, and IRPPDS models were first solved using the Branch-and-Cut solver of CPLEX until reaching optimality. We first considered solving the models with only one type of RTI for a number of customers varying from 6 to 24 in a planning horizon that corresponds to a week of 5 days. We also conducted additional experiments in which we considered a number of RTIs varying from 2 to 10, and finally a number of suppliers varying between 5 and 25. The objective was to provide partial insights regarding the benefits of RTI sharing, the representativeness of the results, and the run time needed to solve this problem. Table 5 summarises the computational results for each instance under consideration. It reports the breakdown of the total cost (TC), namely: transportation (T), inventory of the suppliers (I-S), inventory of the customers (I-C), inventory at the centres for SM (I-K), maintenance (M), procurement of new RTIs (P), and sharing (S). Table 5 also provides the saving (%) between total costs for SM and IRPPDS regarding the total cost of DM and the CPU time in seconds. The saving, noted CS, is computed as follows:

$$CS = \frac{Total\ Cost_{DM} - Total\ Cost_{SM\ or\ IRPPDS}}{Total\ Cost_{DM}} 100.$$

From Table 5, we can see that, as expected, compared to DM, both SM and IRPPDS reduce total costs. Moreover, IRPPDS can help achieve significant cost savings; for IRPPDS, the average total cost was reduced by 40% against 17% for SM. Indeed, in DM, each supplier needs to manage his inventory, deliveries, and the pickups of his empty RTIs from customers. As no shortage is permitted, if his inventory of empty RTIs is not sufficient to meet customer demand, he buys this needed quantity, and a procurement cost is then incurred. Furthermore, regarding transportation costs, as each supplier can only use his RTIs, which cannot be shared among suppliers, the cost of picking up these latter from customers is incurred.

Table 5. Computational results for DM, SM, and IRPPDS on small and medium instances solved using CPLEX.

Instances	Model	T (€)	I-S (€)	I-C (€)	I-K (€)	M (€)	P (€)	S (€)	TC (€)	CS (%)	CPU (s)
	DM	106,899	1386	1428	0	141	280,224	0	390,078	-	424
1R2S6P40V5T	SM	105,309	592	1368	751	133	122,771	2371	233,294	40	629
	IRPPDS	84,854	1188	1308	0	141	98,739	865	187,095	52	451
	DM	315,279	3706	1721	0	299	907,085	0	1,228,090	-	5050
1R2S12P40V5T	SM	245,866	2032	1359	2390	331	737,268	8940	998,186	19	6445
	IRPPDS	228,259	3562	1190	0	294	415,480	2998	651,784	47	5265
	DM	519,475	3067	4988	0	366	831,051	0	1,358,947	-	8776
1R2S18P40V5T	SM	471,330	1710	2680	1765	351	731,678	5504	1,215,018	11	12,115
	IRPPDS	402,425	2532	1911	0	352	611,920	4835	1,023,975	25	9331
	DM	853,012	4136	8040	0	685	3,280,781	0	4,146,653	-	24,314
1R2S24P40V5T	SM	711,300	3230	4061	2890	688	2,758,063	7761	3,487,994	16	31,701
	IRPPDS	552,893	4744	3046	0	696	1,886,929	6107	2,454,415	41	24,399
	DM	267,334	1013	4607	0	152	1,306,998	0	1,580,105	-	473
2R2S5P40V5T	SM	203,254	629	2957	746	171	1,188,760	3240	1,399,757	11	591
	IRPPDS	158,771	928	2011	0	154	781,631	1326	944,821	40	496
	DM	575,795	2055	11,269	0	502	1,719,310	0	2,308,932	-	1309
4R2S5P40V5T	SM	508,117	1177	5829	1432	426	1,493,784	6928	2,017,692	13	1626
	IRPPDS	413,915	1859	5141	0	457	1,056,450	5324	1,483,147	36	1316
	DM	984,677	3886	11,401	0	645	1,953,350	0	2,953,958	-	3013
6R2S5P40V5T	SM	601,250	1993	13,834	3007	578	1,608,985	12,681	2,242,328	24	4131
	IRPPDS	571,245	3372	6552	0	562	1,499,512	14,711	2,095,952	29	4405
	DM	1,196,050	3559	19,337	0	799	2,649,398	0	3,869,143	-	5423
8R2S5P40V5T	SM	1,027,848	2099	17,123	3738	734	2,270,434	221,00	3,344,076	14	7346
	IRPPDS	704,453	3307	11,693	0	765	1,726,959	15,446	2,462,623	36	5968
	DM	1,536,319	7376	22,048	0	1005	3,674,097	0	5,240,844	-	7981
10R2S5P40V5T	SM	1,450,099	3844	23,893	9146	1122	2,852,572	45,735	4,386,411	16	10,675
	IRPPDS	878,014	6777	11,199	0	967	2,263,598	18,631	3,179,185	39	8687
	DM	223,327	1698	3473	0	257	414,556	0	643,312	-	8084
1R5S5P40V5T	SM	213,934	947	3137	964	241	340,187	3415	562,825	13	13,686
	IRPPDS	157,568	1591	1754	0	233	245,555	1440	408,141	37	8969
	DM	470,266	3753	6052	0	544	1,040,464	0	1,521,078	-	22,526
1R10S5P60V5T	SM	465,720	1744	5879	1869	537	487,739	4204	967,693	36	33,882
	IRPPDS	383,377	2661	5856	0	505	209,126	2136	603,661	60	24,005
	DM	1,018,250	6620	12,547	0	798	882,493	0	1,920,708	-	32,387
1R15S5P40V5T	SM	995,463	4898	6364	4941	845	677,583	18,544	1,708,639	11	41,501
	IRPPDS	715,381	6145	4966	0	823	612,775	7922	1,348,012	30	34,055
	DM	1,595,794	9500	21,299	0	1310	1,929,223	0	3,557,125	-	55,543
1R20S5P40V5T	SM	1,419,879	6413	9549	7967	1139	1,724,055	25,988	3,194,989	10	67,511
	IRPPDS	893,618	10,265	7933	0	1181	754,922	9303	1,677,222	53	55,135
	DM	2,251,175	11,306	28,742	0	1488	1,758,976	0	4,051,687	-	67,115
1R25S5P40V5T	SM	2,044,658	6530	25,480	5387	1388	1,510,890	17,212	3,611,545	11	85,413
	IRPPDS	1,439,157	11,087	14,999	0	1536	1,394,438	13,932	2,875,148	29	63,468

In SM, by contrast, as empty RTIs are owned and centrally managed by a pooler company, procurement costs could be reduced thanks to the risk pooling effect. Transportation costs (which include a variable cost that depends on the quantity of RTIs transported) are slightly reduced. Indeed, deliveries incurred by the suppliers remain the same as in DM, but not for the pickups of empty RTIs from customers. These empty RTIs are later transported instead to RTI centres owned by the pooler company, which are assumed to be located near suppliers, and they are transported to the suppliers when required. However, in SM, since the requests of RTI are not balanced between the suppliers, the pooler company must buy the needed quantities and ship them to the suppliers, which increases transportation and procurement costs.

As for IRPPDS, the transportation and procurement costs are significantly reduced. Indeed, in this configuration, the supply chain is centrally managed, and each supplier has his RTIs held at his inventory/customers and picks up empty ones from customers when vehicles visit customers to deliver the required products. In addition, as each supplier can also benefit from this visit to pick up not only his RTIs but also the RTIs of other suppliers, vehicle fill rates are improved (as transportation cost includes a variable cost that depends on the quantity of RTIs transported), and each supplier no longer needs to buy the RTIs

he may need to meet his customers' demand. Orders are thus satisfied by any RTI, and a procurement order is only triggered if needed, which reduces its relative cost compared to DM.

Moreover, in IRPPDS, the risk pooling is maintained as each supplier, when he buys RTIs, adds them to the system pool. Furthermore, the additional management costs of SM, including pooling, inventory, and transportation on the level of the pooler company's centres, are no longer incurred. Furthermore, as is shown in Figure 4, compared to IRPPDS, the quantity of new RTIs bought may represent, for some instances, up to 70% of the available inventory of empty RTIs in SM. More RTIs are purchased at each centre to meet the needs of the suppliers it serves. Finally, from Table 5 and Figure 4, we notice that IRPPDS takes on more interest as the number of RTIs and suppliers increases. Indeed, in SM, when the number of suppliers increases, more centres are needed, especially in different and distant geographical areas, making it challenging to reduce the cost of logistics and procurement for centres servicing clusters housing a significant number of suppliers. As a result, the demands for empty RTIs are not balanced; procurement, inventory, and transportation costs increase as the number of visits from customers to these centres and from these centres to suppliers increases.

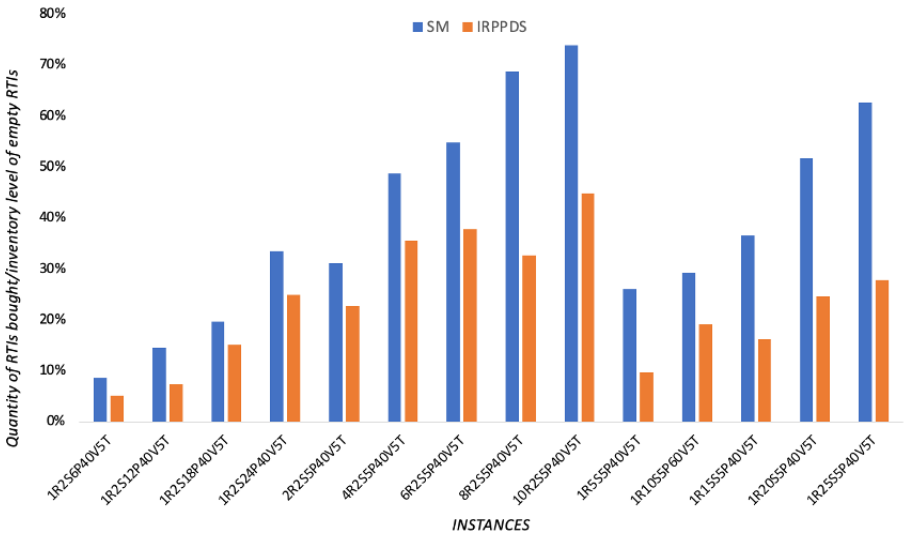

Figure 4. Ratio of quantity of RTIs bought over the inventory level of empty RTIs made available in SM and IRPPDS.

6.2. First Insights into the Effectiveness of the Resolution Approach on Small Instances

As we can see, solving exactly the three models under consideration is very combinatorially complex, and the CPU time increases drastically with the number of suppliers. As described in the experimental design, the resolution approach AIS-DQL was compared to other metaheuristics to assess its performance: GA, AIS, and GA-DQL. Table 6 gives the results of the comparison. The gap regarding total cost is computed as:

$$Gap = \frac{Total\ Cost_{Metaheuristic} - Total\ Cost_{CPLEX}}{Total\ Cost_{CPLEX}} 100.$$

From Table 6, we can see that for all the instances under consideration, AIS-DQL, compared to AIS, GA-DQL, and GA, can find solutions with minor gaps. On average, GA provided solutions with a gap of 12.6%, AIS with 9.4%, GA-DQL with 4.8%, and AIS-DQL with a gap of 0.1%. Indeed, AIS-DQL was more stable, as it was less sensitive to small changes (perturbations) in the input data and the instances' size. Moreover, AIS-DQL allowed for reducing the computational time considerably. GA and AIS may have similar mutation mechanisms, but AIS's immune memory makes it more robust and stable. Furthermore, AIS learning requires increasing the relative population size of each of these antibodies, which proved valuable. A clone is generated temporarily, and those

low-affinity antibodies are eliminated. The goal is to solve the problem using minimal resources and time. Therefore, the algorithm's response efficiency was greatly enhanced by the memory associated with the first and best antibodies obtained for different and similar antigens, and it was capable of providing the best solutions with a high affinity for a given instance only after a few iterations. Indeed, our algorithm ensures that both the speed and accuracy of the immune response are progressively higher after each model resolution. In addition, combined with a deep reinforcement learning technique and KNN, the immune memory further strengthens the interaction with the environment, resulting in a continuous improvement of the algorithm's ability and prior knowledge of similar problems to solve the model for a given instance.

6.3. Extra Experiments on Large Instances Solved Using GA-DQL and AIS-DQL

To obtain more insights into the effectiveness of IRPPDS and AIS-DQL, we further ran tests on large instances and compared the results to those obtained using GA-DQL. We solved the DM, SM, and IRPPDS. Then, we computed the total cost and the corresponding savings. We present the results obtained within a CPU time of less than half an hour. The computational results are summarised in Table 7. Table 7 also reports the difference (Diff) between the total costs computed using GA-DQL and AIS-DQL as follows:

$$Diff = \frac{Total\ Cost_{GA\text{-}DQL} - Total\ Cost_{AIS\text{-}DQL}}{Total\ Cost_{GA\text{-}DQL}} 100.$$

As expected, the AIS-DQL allows for feasible solutions to large-sized problems within a reasonable time. AIS-DQL allowed for obtaining better solutions with an average of 14% compared to GA-DQL and with less time, with an average CPU of 479 s for DM, 557 s for SM, and 556 s for IRPPDS against 743 s for DM, 644 s for SM, and 697 s for IRPPDS. As for the results, for all the instances under consideration, IRPPDS reduced total cost compared to DM, with an average saving of 35% (against 16% for SM). Moreover, the benefits of promoting virtual pooling were highlighted when the number of RTIs and suppliers increased. Furthermore, if the demands to be satisfied required the use of several types of RTIs, the benefits of SM were smaller compared to those of IRPDPS and even DM (according to the results of [14]). This was truer when the number of suppliers increased. Indeed, even if SM can reduce the long-distance transportation of empty RTIs compared to DM, when empty RTIs were not balanced, SM's transportation cost increased since it includes the incurred costs from customers to the pooler's centres from these latter to the suppliers. In addition, even if the centres are located near suppliers (which is often the case in the automotive industry), it would be challenging to balance the quantity of empty RTIs between all suppliers. Therefore, SM may work in favour of or against any supplier regardless of location, the demands they should meet, or the centres' number.

6.4. Sensitivity Analysis on Unit Cost

Considering that the performances may depend on the different unit costs, a sensitivity analysis was conducted, and the results are given in this section. Without loss of generality, we ran tests on the instance 10R20S30C. In each test, we considered three scenarios. The first scenario represents the case where the unit cost of sharing is significantly lower than the smallest unit cost of procurement (s_r = 5). The second one corresponds to the case where the unit cost of sharing is equal to a given unit cost of procurement (s_r = 700). Furthermore, the third scenario represents the case where the unit cost of sharing is significantly higher than the greater unit cost of procurement (s_r = 1800). Figures 5 and 6 depict the variation in cost reduction (CR) for different values of cost parameters. It is worth noting that variable transportation and procurement costs were chosen to conduct the sensitivity analysis due to their significant contributions to the total costs.

Table 6. Assessment of the performance of GA, AIS, GA-DQL, and AIS-DQL compared to CPLEX on relatively small and medium instances.

Instance	Model	CPLEX		GA			AIS			GA-DQL			AIS-DQL		
		TC (€)	CPU (s)	TC (€)	CPU (s)	Gap (%)	TC (€)	CPU (s)	Gap (%)	TC (€)	CPU (s)	Gap (%)	TC (€)	CPU (s)	Gap (%)
1R2S6P40V5T	DM	390,078	424	396,319	8	1.6	395,929	28	1.5	396,319	15	1.6	390,078	2	0.0
	SM	233,294	629	257,557	9	10.4	254,757	22	9.2	245,892	10	5.4	234,461	31	0.5
	IRPPDS	187,095	451	204,308	9	9.2	199,443	41	6.6	200,379	14	7.1	187,095	4	0.0
1R2S12P40V5T	DM	1,228,090	5050	1,331,250	256	8.4	1,312,828	240	6.9	1,251,424	421	1.9	1,228,090	25	0.0
	SM	998,186	6445	1,044,102	427	4.6	1,038,113	205	4.0	1,032,124	307	3.4	1,003,177	21	0.5
	IRPPDS	651,784	5265	677,204	401	3.9	689,587	57	5.8	682,418	461	4.7	651,784	21	0.0
1R2S18P40V5T	DM	1,358,947	8776	1,498,919	84	10.3	1,382,049	32	1.7	1,394,280	89	2.6	1,358,947	18	0.0
	SM	1,215,018	12,115	1,364,466	527	12.3	1,362,036	62	12.1	1,273,339	57	4.8	1,219,879	77	0.4
	IRPPDS	1,023,975	9331	1,135,588	824	10.9	1,087,461	925	6.2	1,054,694	483	3	1,023,975	50	0.0
1R2S24P40V5T	DM	4,146,653	24,314	4,486,679	1394	8.2	4,470,092	884	7.8	4,457,652	1674	7.5	4,146,653	47	0.0
	SM	3,487,994	31,701	3,857,721	989	10.6	3,864,697	508	10.8	3,808,889	416	9.2	3,508,922	21	0.6
	IRPPDS	2,454,415	24,399	2,947,752	2313	20.1	2,923,208	25	19.1	2,648,314	455	7.9	2,454,415	163	0.0
2R2S5P40V5T	DM	1,580,105	473	1,922,988	12	21.7	1,783,939	37	12.9	1,685,972	30	6.7	1,581,685	1	0.1
	SM	1,399,757	591	1,542,532	40	10.2	1,511,738	12	8.0	1,426,353	20	1.9	1,402,557	42	0.2
	IRPPDS	944,821	496	1,037,413	18	9.8	993,952	4	5.2	963,717	35	2	945,766	3	0.1
4R2S5P40V5T	DM	2,308,932	1309	2,673,743	75	15.8	2,542,134	76	10.1	2,403,598	4	4.1	2,308,932	12	0.0
	SM	2,017,692	1626	2,360,700	39	17	2,209,373	41	9.5	2,031,816	51	0.7	2,017,692	22	0.0
	IRPPDS	1,483,147	1316	1,739,731	124	17.3	1,576,585	21	6.3	1,566,203	79	5.6	1,483,147	5	0.0
6R2S5P40V5T	DM	2,953,958	3013	3,202,090	63	8.4	3,190,275	42	8.0	3,140,057	227	6.3	2,953,958	6	0.0
	SM	2,242,328	4131	2,679,583	58	19.5	2,684,067	6	19.7	2,419,472	8	7.9	2,249,055	53	0.3
	IRPPDS	2,095,952	4405	2,290,876	70	9.3	2,292,971	379	9.4	2,179,790	240	4.0	2,098,048	2	0.1
8R2S5P40V5T	DM	3,869,143	5423	4,604,280	538	19.0	4,016,170	527	3.8	3,927,180	392	1.5	3,873,012	29	0.1
	SM	3,344,076	7346	3,691,860	525	10.4	3,410,957	604	2.0	3,390,893	209	1.4	3,347,420	41	0.1
	IRPPDS	2,462,623	5968	2,856,643	350	16	2,570,978	46	4.4	2,561,128	31	4.0	2,462,623	44	0.0
10R2S5P40V5T	DM	5,240,844	7981	5,801,614	733	10.7	5,848,782	670	11.6	5,382,347	293	2.7	5,246,085	28	0.1
	SM	4,386,411	10,675	4,618,891	319	5.3	4,614,504	400	5.2	4,496,071	415	2.5	4,425,888	32	0.9
	IRPPDS	3,179,185	8687	3,795,947	447	19.4	3,808,664	206	19.8	3,344,503	177	5.2	3,179,185	60	0.0
1R5S5P40V5T	DM	643,312	8084	702,497	421	9.2	714,720	63	11.1	656,178	30	2.0	643,955	27	0.1
	SM	562,825	13,686	665,259	276	18.2	613,479	54	9.0	602,786	38	7.1	567,328	6	0.8
	IRPPDS	408,141	8969	454,261	726	11.3	422,426	702	3.5	436,711	100	7.0	408,549	7	0.1
1R10S5P60V5T	DM	1,521,078	22,526	1,718,818	422	13.0	1,630,596	1732	7.2	1,522,599	337	0.1	1,522,599	168	0.1
	SM	967,693	33,882	1,065,430	307	10.1	1,065,430	980	10.1	1,047,043	475	8.2	969,628	52	0.2
	IRPPDS	603,661	24,005	719,564	285	19.2	705,680	508	16.9	644,106	295	6.7	604,265	160	0.1
1R15S5P40V5T	DM	1,920,708	32,387	2,214,576	2272	15.3	2,212,656	1433	15.2	2,058,999	2871	7.2	1,920,708	48	0.0
	SM	1,708,639	41,501	1,942,722	525	13.7	1,905,132	1245	11.5	1,802,614	395	5.5	1,717,182	22	0.5
	IRPPDS	1,348,012	34,055	1,419,457	2947	5.3	1,443,721	1598	7.1	1,376,320	949	2.1	1,349,360	289	0.1
1R20S5P40V5T	DM	3,557,125	55,543	4,197,408	1656	18.0	4,030,223	5005	13.3	3,841,695	148	8.0	3,557,125	103	0.0
	SM	3,194,989	67,511	3,600,753	910	12.7	3,466,563	406	8.5	3,252,499	510	1.8	3,201,379	12	0.2
	IRPPDS	1,677,222	55,135	1,893,584	902	12.9	1,893,584	5312	12.9	1,769,469	451	5.5	1,677,222	296	0.0
1R25S5P40V5T	DM	4,051,687	67,115	4,590,561	3358	13.3	4,517,631	2022	11.5	4,221,858	1636	4.2	4,051,687	524	0.0
	SM	3,611,545	85,413	4,174,946	3140	15.6	4,113,549	400	13.9	3,795,733	211	5.1	3,611,545	42	0.0
	IRPPDS	2,875,148	63,468	3,346,672	4064	16.4	3,197,165	3750	11.2	2,955,652	70	2.8	2,875,148	385	0.0

Table 7. Computational results for large instances obtained using GA-DQL and AIS-DQL.

Instances	AIS-DQL								GA-DQL								Diff (%)		
	DM		SM		IRPPDS		CS (%)		DM		SM		IRPPDS		CS (%)				
	TC (€)	CPU (s)	TC (€)	CPU (s)	TC (€)	CPU (s)	SM	IRPPDS	TC (€)	CPU (s)	TC (€)	CPU (s)	TC (€)	CPU (s)	SM	IRPPDS	DM	SM	IRPPDS
15R15S15C90V5T	1,055,305	330	979,362	274	806,191	217	7	24	1,217,822	527	1,074,615	318	902,934	276	12	26	13	9	11
15R20S15C120V5T	2,698,516	824	2,407,032	475	1,827,526	254	11	32	3,192,344	1207	2,919,561	550	2,077,897	316	9	35	15	18	12
15R30S15C190V5T	5,826,310	44	5,075,948	1070	3,877,637	980	13	33	6,659,472	70	5,079,089	1203	4,552,346	1259	24	32	13	0	15
15R40S15C250V5T	13,140,456	181	12,054,126	335	7,920,967	391	8	40	15,715,985	300	14,922,266	372	9,006,139	473	5	43	16	19	12
15R50S15C350V5T	27,632,333	513	19,565,564	395	17,761,072	329	29	36	31,031,110	829	27,865,271	463	20,673,888	410	10	33	11	30	14
15R60S15C450V5T	64,334,836	135	58,442,222	165	42,490,648	256	9	34	72,248,021	227	71,156,121	197	49,884,021	329	2	31	11	18	15
15R70S15C550V5T	143,862,932	97	134,278,723	156	80,904,270	156	7	44	172,347,793	151	134,553,700	169	97,004,220	198	22	44	17	0	17
15R80S15C650V5T	301,974,530	717	268,660,125	922	200,455,201	1145	11	34	339,721,346	1155	286,184,629	1146	225,311,646	1454	16	34	11	6	11
15R100S15C750V5T	641,210,086	398	606,987,781	443	429,635,068	414	5	33	721,361,347	615	612,861,772	471	514,273,176	536	15	29	11	1	16
15R200S15C2000V5T	1,291,975,415	106	832,110,939	694	663,928,176	751	36	49	1,536,158,768	165	1,135,671,285	852	794,722,027	927	26	48	16	27	16
15R300S15C4000V5T	2,721,138,343	728	2,256,821,880	803	2,082,860,769	665	17	23	3,091,213,158	1146	2,503,447,969	870	2,447,361,404	827	19	21	12	10	15
15R400S15C6000V5T	5,791,824,472	268	4,014,243,737	268	3,126,883,840	237	31	46	6,718,516,388	409	4,960,776,575	289	3,592,789,532	292	26	47	14	19	13
15R600S15C8000V5T	11,431,199,715	384	8,504,611,093	357	7,582,649,005	178	26	34	13,603,127,661	624	9,546,143,904	391	8,507,732,184	217	30	37	16	11	11
31R20S34C400V5T	4,428,883	740	3,736,512	845	3,445,812	971	16	22	5,053,356	1105	4,036,051	954	3,979,913	1182	20	21	12	7	13
31R40S34C900V5T	10,976,187	301	7,719,378	569	7,089,787	839	30	35	13,028,734	475	9,160,248	690	8,507,744	1033	30	35	16	16	17
31R60S34C1300V5T	23,233,219	917	16,916,596	395	14,939,103	312	27	36	27,368,732	1388	25,748,719	424	16,731,795	392	6	39	15	34	11
31R80S34C2500V5T	53,970,019	595	43,722,965	622	28,413,814	603	19	47	62,605,222	890	57,142,014	714	31,993,955	745	9	49	14	23	11
31R110S34C4000V5T	120,114,263	331	97,779,299	724	65,665,251	938	19	45	138,972,202	492	103,217,419	858	77,288,000	1188	26	44	14	5	15
31R130S34C5200V5T	265,313,611	515	251,439,010	284	210,348,468	214	5	21	315,457,883	796	299,527,277	346	243,583,526	277	5	23	16	16	14
31R150S34C6000V5T	572,377,195	294	506,511,145	531	410,627,455	495	12	28	681,701,239	456	569,930,183	608	473,453,456	614	16	31	16	11	13
31R200S34C9000V5T	1,227,011,538	942	1,134,046,280	346	959,955,051	172	8	22	1,425,787,407	1481	1,312,205,027	385	1,136,586,780	221	8	20	14	14	16
31R300S34C14000V5T	2,529,324,214	561	2,398,032,183	563	1,745,537,723	435	5	31	3,030,130,408	869	2,645,201,253	697	2,049,261,287	556	13	32	17	9	15
31R400S3418000V5T	5,169,582,108	319	4,520,814,515	483	3,311,432,130	511	13	36	5,789,931,961	516	5,245,182,129	556	3,751,852,603	640	9	35	11	14	12
31R500S34C20000V5T	11,050,906,466	244	8,845,362,928	547	6,925,392,214	728	20	37	13,006,916,910	376	12,963,475,117	614	7,964,201,046	925	0	39	15	32	13
31R600S34C24500V5T	23,471,937,843	1027	18,265,998,726	1128	14,568,743,068	1110	22	38	26,734,537,203	1598	23,081,746,819	1420	17,278,529,279	1412	14	35	12	21	16
31R700S34C29000V5T	49,800,010,407	933	32,665,152,291	1090	26,308,338,166	1160	34	47	57,270,011,968	1439	56,207,283,103	1185	31,096,455,712	1430	2	46	13	42	15
Average	4,489,886,892	479	3,287,441,168	557	2,646,228,016	556	17	35	5,185,154,402	743	4,685,635,081	644	3,092,258,327	697	14	35	14	16	14

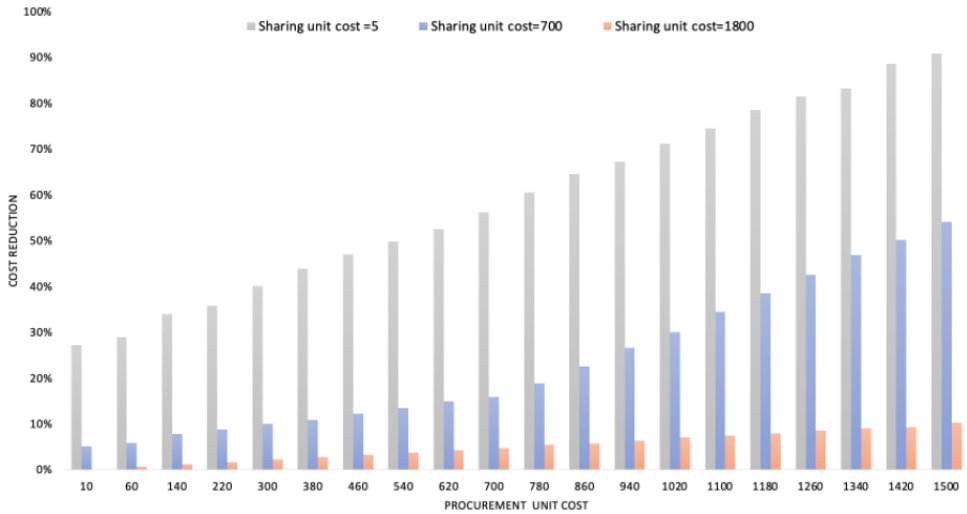

Figure 5. Saving in terms of total costs for the various unit costs of procurement.

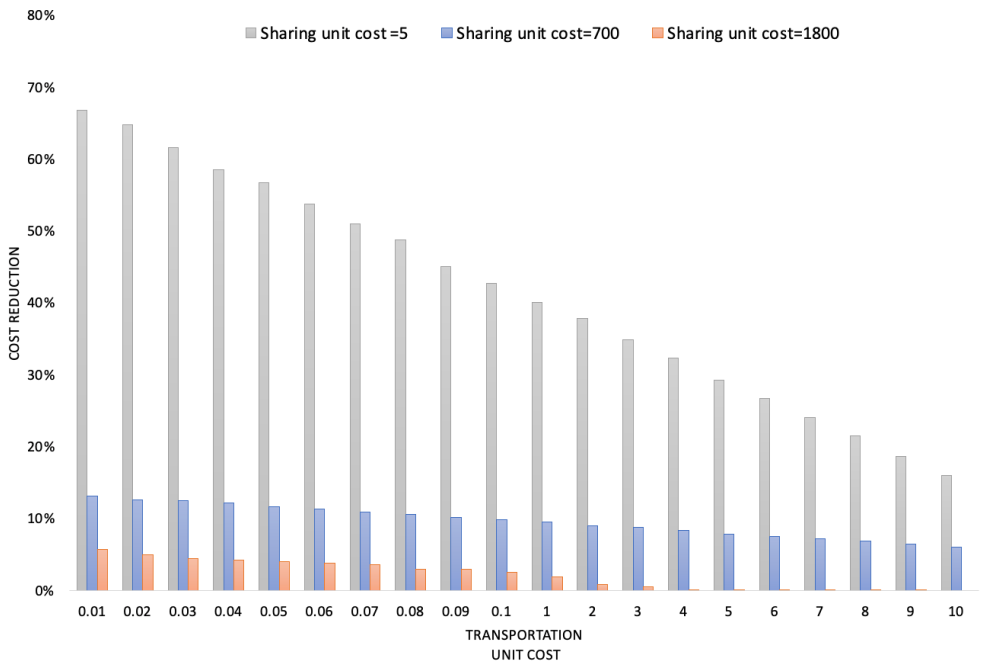

Figure 6. Variation in cost reduction for the various unit costs of transportation.

From Figure 5, we see that from a cost-reducing perspective, IRPPDS generally has obvious advantages compared to DM for the three scenarios under consideration. Moreover, we notice that any change in the procurement cost has the most significant impact on cost reduction for IRPPDS. For the lower unit cost of sharing, we see that as the procurement cost increases, the performance advantages of IRPPDS increase significantly. Indeed, authorising virtual pooling reduces inventory holding costs for empty RTI owners (lowering the idle stock of non-used empty RTIs), while suppliers who use these RTIs can meet more demands when no or fewer RTIs need to be bought. On the other hand, when the procurement cost is smaller compared to the unit cost of sharing, saving is smaller, and the advantages of DM and IRPPDS are comparable (this is even more evident in scenario 3). Thus, the advantages of IRPPDS may lessen as the cost incurred by sharing cannot be offset by the saving it brings regarding the reduction of procurement costs. Therefore, IRPPDS becomes profitable with a higher procurement cost. Moreover, as shown in Figure 6, significant savings are achieved when the sharing cost is smaller. In addition, when the transportation cost increases, IRPPDS is more profitable compared to DM. Indeed, more empty and loaded RTIs can be

transported in a period (high fill rates), while fewer customers are visited and fewer RTIs are bought in the next period. However, the predominance of IRPPDS may weaken when the sharing cost increases (this was even more evident in scenario 3, with GAP tending to zero). Indeed, with higher unit costs of sharing and transportation, it would be preferable and more cost-effective to have low fill rates (i.e., not to accept loading of unowned RTIs and to send them to the suppliers for further reuse) and to buy the needed RTIs rather than to have to pay for shared RTIs. Consequently, DM may be more profitable compared to IRPPDS.

7. Conclusions and Perspectives

This paper considered a deterministic, multi-supplier, multi-customer and multi-RTI inventory routing problem with delivery and pickup in a collaborative supply chain in which empty RTIs inventories are virtually pooled among suppliers. We developed an MILP and solved it using CPLEX. Experiments showed that the virtual pooling of RTIs significantly reduces new RTI procurement costs as well as inventory and transportation costs compared to dedicated and shared modes. Moreover, to handle the combinatorial complexity of the problem, we developed an artificial-immune-system-based algorithm coupled with deep reinforcement learning tailored to the mathematical program. We implemented our resolution approach using Python and Pytorch and compared it to the CPLEX solver and three metaheuristics, namely, AIS without deep learning and GA with and without deep learning. Both variants of GA and AIS coupled with DQL seem to be competitive. However, the AIS variant outperformed GA thanks to its immune memory, which continuously improved the algorithm's speed and stability in solving the model. AIS-QDL even allowed for obtaining optimal solutions for some instances and feasible solutions with a tiny gap and within a small amount of time. Using AIS-QDL, we solved the model for large instances of up to 700 suppliers, 34 customers, and 31 types of RTIs. A sensitivity analysis of units' costs was also conducted. These results highlight how virtual pooling can be preferable compared to the dedicated and shared modes.

While the benefits of the model and the effectiveness of the AIS-DQL were demonstrated using randomly generated instances, it would be beneficial to assess further their effectiveness on real data. Moreover, several possible applications may be investigated. For example, one could study the integration of cross docks in the RTI flows, as in the case of automotive supply chains. The idea is to combine and consolidate, when it seems advantageous, numerous smaller RTI loads provided by different suppliers and to deliver them downstream. Future research may also investigate the case of stochastic demands as room to exploit further and assess the limits of the resolution approach to tackle this kind of problem, the relative power of all parties in decision making, and maximising profit and its allocation. One way to address this latter may rely on the degree of commitment of the players. Indeed, as many supply chains experience the highest loss and damage rates of RTIs (which can be trackable using, for instance, RFID tags), the pool manager can reduce the costs incurred by "good" users and increase those of the "bad" ones or offer them training on the use of these RTIs so that they can improve on their weak points, reduce the environmental impacts, and increase the competitiveness of the whole system. Furthermore, decisions related to fleet composition and fuel consumption are to be considered in future work.

Author Contributions: Conceptualization, F.E.A., F.R., E.S. and S.L.; methodology, F.E.A. and S.L.; software, F.E.A.; validation, E.S.; investigation, F.E.A., F.R. and S.L.; resources, F.E.A.; data curation, F.E.A.; writing—original draft preparation, F.E.A., F.R., E.S. and S.L.; writing—review and editing, F.E.A., F.R., E.S. and S.L.; visualization, F.E.A. and S.L.; supervision, F.R., E.S. and S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Cobb, B. Inventory control for returnable transport items in a closed-loop supply chain. *Transp. Res. Part E Logist. Transp. Rev.* **2016**, *86*, 53–68. [CrossRef]
2. Kim, T.; Glock, C. On the use of RFID in the management of reusable containers in closed-loop supply chains under stochastic container return quantities. *Transp. Res. Part E Logist. Transp. Rev.* **2014**, *64*, 12–27. [CrossRef]
3. Limbourg, S.; Martin, A.; Paquay, C. Optimal Returnable Transport Items Management. 2016. Available online: https://hdl.handle.net/2268/200983 (accessed on 1 May 2022).
4. Iassinovskaia, G.; Limbourg, S.; Riane, F. The inventory-routing problem of returnable transport items with time windows and simultaneous pickup and delivery in closed-loop supply chains. *Int. J. Prod. Econ.* **2017**, *183*, 570–582. [CrossRef]
5. Glock, C.H. Decision support models for managing returnable transport items in supply chains: A systematic literature review. *Int. J. Prod. Econ.* **2017**, *183*, 561–569. [CrossRef]
6. Bortolini, M.; Galizia, F.G.; Mora, C.; Botti, L.; Rosano, M. Bi-objective design of fresh food supply chain networks with reusable and disposable packaging containers. *J. Clean. Prod.* **2018**, *184*, 375–388. [CrossRef]
7. Liu, G.; Li, L.; Chen, J.; Ma, F. Inventory sharing strategy and optimization for reusable transport items. *Int. J. Prod. Econ.* **2020**, *228*, 107742. [CrossRef]
8. Twede, D.; Clarke, R. Supply Chain Issues in Reusable Packaging. *J. Mark. Channels* **2005**, *12*, 7–26. [CrossRef]
9. Sarkar, B.; Ullah, M.; Kim, N. Environmental and economic assessment of closed-loop supply chain with remanufacturing and returnable transport items. *Comput. Ind. Eng.* **2017**, *111*, 148–163. [CrossRef]
10. Talaei, M.; Farhang Moghaddam, B.; Pishvaee, M.S.; Bozorgi-Amiri, A.; Gholamnejad, S. A robust fuzzy optimization model for carbon-efficient closed-loop supply chain network design problem: A numerical illustration in electronics industry. *J. Clean. Prod.* **2016**, *113*, 662–673. [CrossRef]
11. Meherishi, L.; Narayana, S.A.; Ranjani, K.S. Integrated product and packaging decisions with secondary packaging returns and protective packaging management. *Eur. J. Oper. Res.* **2021**, *292*, 930–952. [CrossRef]
12. TrackX. A Return on Returnables. Technical Report. 2017. Available online: https://www.omniq.com/wp-content/uploads/2020/11/Quest-TrackX-WP-RTI-Assets.pdf (accessed on 18 May 2022)
13. Na, B.; Sim, M.K.; Lee, W.J. An Optimal Purchase Decision of Reusable Packaging in the Automotive Industry. *Sustainability* **2019**, *11*, 6579. [CrossRef]
14. Zhang, Q.; Segerstedt, A.; Tsao, Y.C.; Liu, B. Returnable packaging management in automotive parts logistics: Dedicated mode and shared mode. *Int. J. Prod. Econ.* **2015**, *168*, 234–244. [CrossRef]
15. GS1 Global Office. *Reusable Transport Items within GS1 EANCOM*; Technical Report; GS1 Global Office: Brussels, Belgium, 2014.
16. Accorsi, R.; Baruffaldi, G.; Manzini, R.; Pini, C. Environmental Impacts of Reusable Transport Items: A Case Study of Pallet Pooling in a Retailer Supply Chain. *Sustainability* **2019**, *11*, 3147. [CrossRef]
17. Govindan, K.; Soleimani, H.; Kannan, D. Reverse logistics and closed-loop supply chain: A comprehensive review to explore the future. *Eur. J. Oper. Res.* **2015**, *240*, 603–626. [CrossRef]
18. Paterson, C.; Kiesmüller, G.; Teunter, R.; Glazebrook, K. Inventory models with lateral transshipments: A review. *Eur. J. Oper. Res.* **2011**, *210*, 125–136. [CrossRef]
19. Achamrah, F.E.; Riane, F.; Limbourg, S. Solving inventory routing with transshipment and substitution under dynamic and stochastic demands using genetic algorithm and deep reinforcement learning. *Int. J. Prod. Res.* **2021**, 1–18. [CrossRef]
20. Toth, P.; Vigo, D. *The Vehicle Routing Problem*; SIAM: Philadelphia, PA, USA, 2002.
21. Berbeglia, G.; Cordeau, J.F.; Gribkovskaia, I.; Laporte, G. Static pickup and delivery problems: A classification scheme and survey. *TOP Off. J. Span. Soc. Stat. Oper. Res.* **2007**, *15*, 1–31. [CrossRef]
22. Andersson, H.; Christiansen, M.; Fagerholt, K. The Maritime Pickup and Delivery Problem with Time Windows and Split Loads. *INFOR Inf. Syst. Oper. Res.* **2011**, *49*, 79–91. [CrossRef]
23. Rais, A.; Alvelos, F.; Carvalho, M.S. New mixed integer-programming model for the pickup-and-delivery problem with transshipment. *Eur. J. Oper. Res.* **2014**, *235*, 530–539. [CrossRef]
24. Chen, H.K.; Chou, H.W.; Hsueh, C.F.; Yu, Y.J. The paired many-to-many pickup and delivery problem: An application. *TOP* **2014**, *23*, 220–243. [CrossRef]
25. Li, B.; Krushinsky, D.; Van Woensel, T.; Reijers, H. An Adaptive Large Neighborhood Search Heuristic for the Share-a-Ride Problem. *Comput. Oper. Res.* **2015**, *66*, 170–180. [CrossRef]
26. Coelho, L.C.; Laporte, G. Improved solutions for inventory-routing problems through valid inequalities and input ordering. *Int. J. Prod. Econ.* **2014**, *155*, 391–397. [CrossRef]
27. Parragh, S.N.; Doerner, K.F.; Hartl, R.F. A survey on pickup and delivery problems. *J. Betr.* **2008**, *58*, 21–51. [CrossRef]
28. Tarantilis, C.D.; Anagnostopoulou, A.K.; Repoussis, P.P. Adaptive Path Relinking for Vehicle Routing and Scheduling Problems with Product Returns. *Transp. Sci.* **2012**, *47*, 356–379. [CrossRef]

29. Archetti, C.; Christiansen, M.; Grazia Speranza, M. Inventory routing with pickups and deliveries. *Eur. J. Oper. Res.* **2018**, *268*, 314–324. [CrossRef]
30. Van der Heide, G.; Buijs, P.; Roodbergen, K.J.; Vis, I.F.A. Dynamic shipments of inventories in shared warehouse and transportation networks. *Transp. Res. Part E Logist. Transp. Rev.* **2018**, *118*, 240–257. [CrossRef]
31. Archetti, C.; Speranza, M.G.; Boccia, M.; Sforza, A.; Sterle, C. A branch-and-cut algorithm for the inventory routing problem with pickups and deliveries. *Eur. J. Oper. Res.* **2020**, *282*, 886–895. [CrossRef]
32. Meherishi, L.; Narayana, S.A.; Ranjani, K.S. Sustainable packaging for supply chain management in the circular economy: A review. *J. Clean. Prod.* **2019**, *237*, 117582. [CrossRef]
33. Achamrah, F.E.; Bouras, A.; Riane, F.; Darmoul, S. Returnable Transport Items Management: A New Approach to Sidestep Shortage. In Proceedings of the 7th International Conference IEEE on Advanced Logistics and Transport, Marrakech, Morocco, 14–16 June 2019; pp. 92–97.
34. Singh, S.K.; Gupta, Y.; Mishra, A.; Darla, S. Inventory Routing Problem with Simultaneous Pickup and Delivery of Returnable Transport Items with Consideration of Renting and Repairing. *Int. J. Eng. Res. Technol.* **2017**, *6*, 92–97.
35. Ech-Charrat, M.R.; Amechnoue, K.; Zouadi, T. Dynamic Planning of Reusable Containers in a Close-loop Supply Chain under Carbon Emission Constrain. *Int. J. Supply Oper. Manag.* **2017**, *4*, 279–297. [CrossRef]
36. Ren, J.; Liu, B.; Wang, Z. An optimization model for multi-type pallet allocation over a pallet pool. *Adv. Mech. Eng.* **2017**, *9*, 1687814017705841. [CrossRef]
37. Tornese, F.; Pazour, J.A.; Thorn, B.K.; Roy, D.; Carrano, A.L. Investigating the environmental and economic impact of loading conditions and repositioning strategies for pallet pooling providers. *J. Clean. Prod.* **2018**, *172*, 155–168. [CrossRef]
38. Hassanzadeh Amin, S.; Wu, H.; Karaphillis, G. A perspective on the reverse logistics of plastic pallets in Canada. *J. Remanuf.* **2018**, *8*, 153–174. [CrossRef]
39. Zhou, K.; Song, R. Location Model of Pallet Service Centers Based on the Pallet Pool Mode. In Proceedings of the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 15–17 March 2019; pp. 1185–1189. [CrossRef]
40. Achamrah, F.E.; Riane, F.; Bouras, A.; Sahin, E. Collaboration Mechanism for Shared Returnable Transport Items in Closed Loop Supply Chains. In Proceedings of the 9th International Conference on Operations Research and Enterprise Systems, Valletta, Malta, 22–24 February 2020. [CrossRef]
41. Karp, R. *Reducibility Among Combinatorial Problems*; Springer: Boston, MA, USA, 1972; Volume 40, pp. 85–103. [CrossRef]
42. Papadimitriou, C.; Steiglitz, K. On the Complexity of Local Search for the Traveling Salesman Problem. *SIAM J. Comput.* **1977**, *6*, 76–83. [CrossRef]
43. Bernardino, H.S.; Barbosa, H.J.C. *Grammar-Based Immune Programming for Symbolic Regression BT—Artificial Immune Systems*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 274–287.
44. De Castro, L.; Jos, F.; von Zuben, A.A. Artificial Immune Systems: Part II—A Survey Of Applications. 2000. Available online: http://https://www.dca.fee.unicamp.br/~vonzuben/tr_dca/trdca0200.pdf (accessed on 1 May 2022)
45. Wong, E.Y.C.; Lau, H.Y.K.; Mak, K.L. Immunity-based evolutionary algorithm for optimal global container repositioning in liner shipping. *Spectrum* **2010**, *32*, 739–763. [CrossRef]
46. Tiwari, M.K.; Prakash; Kumar, A.; Mileham, A.R. Determination of an optimal assembly sequence using the psychoclonal algorithm. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2005**, *219*, 137–149. [CrossRef]
47. Panigrahi, B.K.; Yadav, S.R.; Agrawal, S.; Tiwari, M.K. A clonal algorithm to solve economic load dispatch. *Electr. Power Syst. Res.* **2007**, *77*, 1381–1389. [CrossRef]
48. Pierrard, T.; Coello Coello, C.A. *A Multi-Objective Artificial Immune System Based on Hypervolume BT—Artificial Immune Systems*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 14–27.
49. Navarro, M.; Herath, P.; Villarrubia, G.; Prieto-Castrillo, F.; Venyagamoorthy, G. An Evaluation of a Metaheuristic Artificial Immune System for Household Energy Optimization. *Complexity* **2018**, *2018*, 9597158. [CrossRef]
50. Talbi, E.G. Machine Learning into Metaheuristics: A Survey and Taxonomy of Data-Driven Metaheuristics. 2020. Available online: https://hal.inria.fr/hal-02745295/file/ACM-CR.pdf (accessed on 1 May 2022).
51. Bello, I.; Pham, H.; Le, Q.; Norouzi, M.; Bengio, S. Neural Combinatorial Optimization with Reinforcement Learning. *arXiv* **2016**, arXiv:1611.09940.
52. Dai, H.; Khalil, E.; Yuyu, Z.; Dilkina, B.; Song, L. Learning Combinatorial Optimization Algorithms over Graphs. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017.
53. Han, M.; Senellart, P.; Bressan, S.; Wu, H. Routing an autonomous taxi with reinforcement learning. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, Indianapolis, IN, USA, 24–28 October 2016; pp. 2421–2424.
54. Kaempfer, Y.; Wolf, L. Learning the multiple traveling salesmen problem with permutation invariant pooling networks. *arXiv* **2018**, arXiv:1803.09621.
55. Huang, D.; Mao, Z.; Fang, K.; Chen, L. Solving the shortest path interdiction problem via reinforcement learning. *Int. J. Prod. Res.* **2021**, 1–18.

56. Ahmadian, S. Approximation Algorithms for Clustering and Facility Location Problems. Ph.D. Thesis, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 2017. Available online: http://hdl.handle.net/10012/11640 (accessed on 1 May 2022).
57. OroojlooyJadid, A.; Hajinezhad, D. A review of cooperative multi-agent deep reinforcement learning. *arXiv* **2019**, arXiv:1908.03963.
58. Lu, H.; Zhang, X.; Yang, S. A Learning-Based Iterative Method for Solving Vehicle Routing Problems. 2019. Available online: https://openreview.net/forum?id=BJe1334YDH (accessed on 1 May 2022).
59. Duan, L.; Zhan, Y.; Hu, H.; Gong, Y.; Wei, J.; Zhang, X.; Xu, Y. Efficiently solving the practical vehicle routing problem: A novel joint learning approach. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 6–10 July 2020; pp. 3054–3063.
60. Achamrah, F.E.; Riane, F.; Aghezzaf, E.H. Bi-level programming for modelling inventory sharing in decentralized supply chains. *Transp. Res. Procedia* **2022**, *62*, 517–524. [CrossRef]
61. Nakib, A.; Hilia, M.; Heliodore, F.; Talbi, E.G. Design of metaheuristic based on machine learning: A unified approach. In Proceedings of the 2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Lake Buena Vista, FL, USA, 29 May–2 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 510–518.
62. Seyyedabbasi, A.; Aliyev, R.; Kiani, F.; Gulle, M.U.; Basyildiz, H.; Shah, M.A. Hybrid algorithms based on combining reinforcement learning and metaheuristic methods to solve global optimization problems. *Knowl. Based Syst.* **2021**, *223*, 107044. [CrossRef]
63. Talbi, E.G. *Metaheuristics: From Design to Implementation*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 74,
64. Chiang, C.W.; Lee, W.P.; Heh, J.S. A 2-Opt based differential evolution for global optimization. *Appl. Soft Comput.* **2010**, *10*, 1200–1207. [CrossRef]
65. Alom, M.Z.; Taha, T.M.; Yakopcic, C.; Westberg, S.; Sidike, P.; Nasrin, M.S.; Hasan, M.; Van Essen, B.C.; Awwal, A.A.S.; Asari, V.K. A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics* **2019**, *8*, 292. [CrossRef]
66. Mohtashami, A. A Novel Dynamic Genetic Algorithm-Based Method for Vehicle Scheduling in Cross Docking Systems with Frequent Unloading Operation. *Comput. Ind. Eng.* **2015**, *90*, 221–240. [CrossRef]