*Proceedings of the 2022 Winter Simulation Conference*
*B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.*

# SIMULATION OF THE INTERNAL ELECTRIC FLEET DISPATCHING PROBLEM AT A SEAPORT: A REINFORCEMENT LEARNING APPROACH

Matteo Brunetti
Giovanni Campuzano
Martijn Mes

Department of High-Tech Business and Entrepreneurship
University of Twente
P.O. Box 217
Drienerlolaan 5
Enschede, 7500 AE, THE NETHERLANDS

## ABSTRACT

Through discrete-event simulation, we evaluate the impact of using a fleet of electric and autonomous vehicles (EAVs) to decouple inbound trucks from the internal freight flows in a seaport located in the Netherlands. To support the operational control of EAVs, we use agent-based modeling and support the decision-making capabilities using a reinforcement learning (RL) approach. More specifically, to model the assignment of EAVs to container transport or battery charge, we introduce the Internal Electric Fleet Dispatching Problem (IEFDP). To solve the IEFDP, we propose an RL approach and benchmark its performance against four different assignment heuristics. Our results are compelling: the RL approach outperforms the benchmark heuristics, and the decoupling process significantly reduces congestion and waiting times for truck drivers as well as potentially improve the traffic's sustainability, against a slight increase in length of stay of containers at the port.

## 1 INTRODUCTION

Freight transport volumes have been increasing for decades and are expected to more than double by 2050 (ITF 2021). This, combined with containerization, greatly affects the intermodal logistics areas that are at the heart of international freight networks, such as ports and business parks in the hinterland (Behdani et al. 2020). At these locations, a promising solution to increase throughput, and reduce congestion and operational costs is the use of automated vehicles (AVs). AV systems are a type of vehicle-based internal transport system traditionally used in manufacturing plants, distribution centers, container terminals, and other confined environments (Le-Anh and De Koster 2006). Recent advances in technology have increased the popularity of AVs along the logistics chain, where companies can now automate their logistics operations outside of private yards, for example, with AVs shunting containers between terminals in a port area.

Currently, research is focused on electric automated vehicles (EAVs) due to increasing sustainability concerns and decarbonization goals (Vdovic et al. 2019). By implementing EAV systems, organizations aim to improve sustainability, flexibility, and efficiency. Furthermore, following the improvements in EAV technology and the increase in freight volumes, the authorities of logistics areas, e.g., ports and business parks, are now looking at scaling up the use of EAV systems for the whole area, thus servicing multiple logistics companies (LCs), e.g., terminals, warehouses, and cross-docking centers. The goal is to share an EAV fleet and coordinate transportation to increase operational efficiency and reduce congestion in logistics areas, thus improving both safety and throughput of goods. By sharing EAVs, the LCs can benefit from economies of scale and risk sharing, e.g., with regards to the cost of EAV ownership. The shift of paradigm

from an intra-company to inter-company fleet of EAVs, e.g., for inter-terminal transport (ITT), brings many operational challenges regarding the dynamic mixed-traffic environment, the efficient fleet management considering operational constraints, and the coordination of decisions and information.

In this work, we address the first two problems by means of a detailed simulation model, and solve the third by a reinforcement learning (RL) approach, i.e., the centralized dispatching of a shared EAV fleet for internal transport at a logistics area, considering charging schedules, time-windows for delivery, and uncertain arrivals of shipments. We refer to this problem as the Internal Electric Fleet Dispatching Problem (IEFDP). We provide a Markov Decision Process (MDP) formulation that minimizes logistics costs and lateness, and solve the IEFDP using the RL approach. We train the RL algorithm in a simplified environment and test its performance in a detailed simulation model of the Port of Moerdijk (PoM), the Netherlands, to consider complex traffic dynamics and operational constraints. We aim at (*i*) modeling a detailed logistics system that faces uncertain events, where the delayed reward of current actions is considered in the decision-making process, and (*ii*) designing a centralized dispatching algorithm to efficiently manage a shared EAV fleet with operational constraints. The main contributions of this paper are the following.

1. We show the applicability of a general simulation framework for logistics areas to a major port within the Netherlands (i.e., PoM), for the analysis of freight flows and emerging technologies, such as EAVs for ITT.
2. We provide an MDP formulation for a new ITT problem called the Internal Electric Fleet Dispatching Problem (IEFDP). The IEFDP focuses on the dispatching of EAVs with containers, considering uncertain arrival of containers, energy consumption, and time-windows.
3. We provide insight into how the PoM could benefit from an EAV fleet and RL approach for the ITT of inbound containers between a pre-gate parking site and various LCs at the port. We compare this new scenario against the current situation in a detailed simulation model.

The remainder of this paper is structured as follows. A concise literature review is presented in Section 2. We formally introduce the problem as an MDP and provide the RL algorithm for the IEFDP in Section 3. The simulation model and specific elements of the port simulation are described in Section 4. Numerical experiments and results are presented in Section 5. Finally, we draw conclusions in Section 6.

## 2 LITERATURE REVIEW

The growth in freight volumes greatly impacts intermodal logistics areas, such as ports and business parks in the hinterland. In this context, several ports such as Shanghai and Rotterdam are investing in multi-terminal systems, which inherently result in highly complex transport systems within the port (Hu et al. 2019). As a result, companies face new challenges in the field of ITT problems.

At the operational level, companies need to properly plan transport, deciding which vehicle transports which container and using what route. Literature displays a wide range of problems to perform the freight consolidation, which may involve different transportation services such as truck, rail, and barge (Heilig and Voß 2017). In port areas, it is crucial to adopt digital platforms that share real-time information to plan, control, and coordinate vehicle movements in ITT networks (Evers 2006). Heilig et al. (2017) propose the Inter-Terminal Truck Routing Problem (ITTRP) that considers economic and ecological factors, and a prototype decision support system for managing and planning ITT. To solve the ITTRP, they implement two greedy heuristics and two hybrid simulated annealing algorithms, evaluating real-life instances based on the port of Hamburg. Results show that the two hybrid heuristics achieve better performance than the greedy heuristics. Hu et al. (2018) focus their study on the integrated planning of an inter-terminal network connected with a hinterland rail network. They develop a tabu search algorithm, where results show that the connection of ITT and external hinterland transport processes yields a reduction of 20% in ITT costs and 44% in operational railway costs. In addition, most of the research on ITT focuses on land-side vehicles in port areas (Hu et al. 2018). However, since in certain terminals the travel distances are

much shorter by water, Zheng et al. (2021) propose a dynamic waterborne ITT problem with autonomous guided vessels to face ITT requests. A tabu search algorithm with a restart strategy, in combination with a rolling horizon, is developed to solve this problem. Simulation results based on the port of Rotterdam show that larger fleet sizes and longer prediction horizons result in better scheduling performance. For an extensive review on ITT, we refer the reader to Heilig and Voß (2017). When considering electric vehicles for the ITTRP, research on the electric vehicle routing problem with time-windows and recharging stations (E-VRP-TW-RS) (Schneider et al. 2014) is relevant.

Although extensive research has been carried out on AVs and ITT separately, to our knowledge, optimal dispatching policies for EAVs in ITT have not been studied. This becomes particularly important due to the high costs of EAVs, the cost and environmental impacts of battery replacements, and the effects on operational service levels. Consequently, we aim to determine optimal charging actions of EAVs, following a similar approach for mixed RL and simulation as in Asadi and Pinkley (2021). Usually, simulation studies on port logistics tend to implement human-like heuristics to support decision-making. These strategies might be accurate and fast but typically lack the ability to look ahead utilizing past experience. As a result, we aim at analyzing the simulation of a port using an RL algorithm for the assignment of EAVs to delivery and charging tasks, and to compare this system with a traditional internal logistics process.

## 3 REINFORCEMENT LEARNING APPROACH

The IEFDP is formally introduced in Section 3.1 using an MDP formulation. We solve the MDP approximately using a value-based RL algorithm as presented in Section 3.2.

### 3.1 Markov Decision Process Formulation

In the IEFDP, a logistics area comprises a single parking site (PS), at which trucks arrive with containers, and several LCs, at which the containers should be delivered. As such, this problem models the daily situation of a central fleet dispatcher that should transport containers from the PS to the LCs during working hours. The transport of containers is carried out by EAVs and the delivery should meet the containers' time-windows for delivery. The EAVs perform the deliveries in round trips that start and end at the PS and have a re-chargeable battery with several energy levels. If unable to fulfill these transport jobs, a manned vehicle takes care of the transport. Figure 1 illustrates a logistics area with a PS and several LCs.

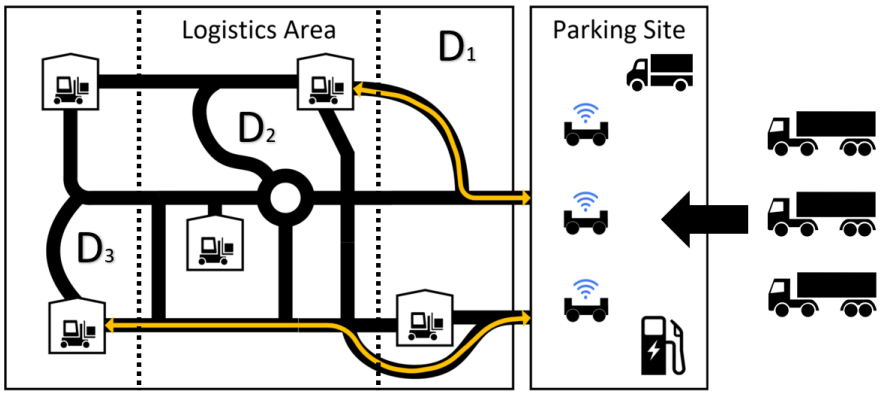


Figure 1: Visualization of the IEFDP with three distance classes for the LCs.

The problem can be defined as a set of EAVs that should transport containers $j \in J = \{1,2,...,\mathrm{J}\}$ in round trips from a PS to LCs of distance class $d \in D = \{1,...,\mathrm{D}\}$, which refers to the distance between the PS and the LC to which the container will be transported. An arbitrarily long horizon is discretized in consecutive time periods $t \in T = \{1,2,...,\mathrm{T}\}$, from now on called stages. This finite horizon allows all input to the model to be time-dependent and enables incorporating anticipated or forecasted fluctuations

between stages. At the last stage, the final costs are computed to evaluate the dispatcher's performance. Furthermore, containers arrive at the PS from outside the system according to a stochastic process with a rate $\lambda_{j,t}, \forall j \in J, t \in T$. Every container should be delivered within its corresponding time-window of length $k \in K$ stages, which starts when its release period $r \in R = \{0, ..., \mathrm{R}\}$ is equal to zero. For example, a given container that has $r = 2$ and $k = 1$ will become available for pickup after the next two upcoming stages and it will have only one stage to be transported. If the shipment is not delivered on time, an alternative transportation mode, i.e., a manned yard tractor, performs the transport at a high cost $C^L$. Furthermore, EAVs have one unit load capacity and $b \in B = \{0, 1, 2, ...\mathrm{B}\}$ incremental battery levels, where $b = 0$ means that the EAV ran out of battery and can only recharge. There are $Q$ charging stations located at the PS. Transporting a given container $j \in J$ to an LC of distance class $d \in D$ takes $d$ stages and battery levels.

The goal of the central dispatcher is to manage the fleet of EAVs, with regards to transport and charging decisions, and thereby minimize the logistics costs under uncertainty, i.e., to maximize the use of the EAV fleet and minimize the use of the manned vehicle while considering stochastic container arrivals and anticipatory information. In the IEFDP, we make the following assumptions. First, we assume that charging the battery by one level requires one stage. Second, we do not consider transportation costs for the EAVs' deliveries, as the system should always bear those costs. Third, the manned yard tractor is always available and transports all late containers at the end of the stage instantaneously. Fourth, for the same LC, the route choice is negligible and the travel times are deterministic. Fifth, all LCs of distance class $d \in D$ require the same travel time and battery consumption at any given moment. Last, we assume that container arrivals are independent and identically distributed events, and their destination is a random LC.

We now formulate the MDP for the IEFDP described above. In the IEFDP, each period of time $t$ corresponds to a *stage* in the MDP formulation. Thus, stages are discrete and consecutive. Furthermore, at each stage $t$, there are $J_{t,d,r,k}$ containers with destination $d$, release stage $r$, and time-window length $k$ at the PS, and there are $V_{t,r,b}$ EAVs with release stage $r$ and battery level $b$ to transport the containers. The state of the system $S_t$ consists of all container and vehicle variables at stage $t$, as seen in (1). We denote the state space of the system by $S$, i.e., $S_t \in S$.

$$S_t = \Big[(J_{t,d,r,k}, V_{t,r,b})\Big]_{\forall d \in D, r \in R, k \in K, b \in B} \tag{1}$$

At each stage $t$, the decision consists of (*i*) how many EAVs should transport containers of a certain type and (*ii*) how many EAVs with a certain range should charge their battery. This decision is restricted by the release day of the shipments, the battery level of the vehicles, and the number of vehicles available. We use continuous variables $X^V_{t,d,k,b}$ and $X^C_{t,b}$ to represent the number of vehicles used to transport containers to an LC of type $d$ with time-window $k$ and battery level $b$, and the number of vehicles with battery level $b$ sent to charge the battery, respectively. Decision $x_t$ consists of all decision variables at stage $t$, i.e., $x_t \in X_t = \Big[(X^V_{t,d,k,b}, X^C_{t,b})\Big], \forall d \in D, k \in K, b \in B$. Here, the decision space $X_t$ is subject to constraints that establish that (*i*) the maximum number of EAVs used cannot be larger than the available number of EAVs, i.e., $\sum_{k \in K} \sum_{d \in D} X^V_{t,d,k,b} + X^C_{t,b} \leq V_{t,0,b}, \forall b \in B$; (*ii*) the EAVs used to transport containers to LC's of type $d$ cannot exceed the number of containers available with the given destination type $d$, i.e., $\sum_{b \in B} X^V_{t,d,k,b} \leq J_{t,d,0,k}, \forall d \in D, k \in K$; and (*iii*) the number of EAVs sent to charge the battery at any stage $t$ cannot be larger than the number of charging stations, $\sum_{b \in B} X^C_{t,b} \leq Q$.

The transition from $S_{t-1}$ to $S_t$ is influenced by the decision $x_t \in X_t$ and the containers that arrive after this decision. Note that arriving containers and their characteristics, i.e., the destination, the time-window length, and the releasing stage, are stochastic and characterized by probability distributions. To model these stochastic processes, we introduce $\hat{J}^e_{t,d,r,k}$ to represent the number of newly arriving containers to be transported. This variable is defined with respect to stages $t-1$ and $t$, such that at $t$ all information is known. The exogenous information $W_t$ at stage $t$ consists of all the new information $\hat{J}^e_{t,d,r,k}$, i.e., $W_t = [(\hat{J}^e_{t,d,r,k})]_{\forall d \in D, r \in R, k \in K}$. The state $S_t$ at stage $t$ occurs as a result of the state of the previous stage $S_{t-1}$, the decision of the previous stage $x_{t-1}$ plus the exogenous information captured in $W_t$ that became known between the stages. Accordingly,

the transition of the jobs is set by the time-window $k$ of the container, relative to the release time $r$, the number of containers transported in the previous stage $t-1$, and the stochastic arrival of new containers. All of these factors, and index relations, are used to capture the transition of the system. We represent them using the transition function $S^M$, i.e., $S_t = S^M(S_{t-1}, X_{t-1}, W_t(\hat{J}_t^e))\ \forall t \in T |\ t > 0$.

The objective function $C(S_t, X_t) = \sum_{d \in D} C^L \cdot z_{t,d}$ in stage $t$ depends on the use of the alternative transportation mode. We define the variable $z_{t,d}$ as the number of containers delivered to $d$ by the alternative transportation mode at stage $t$. This variable is constrained by the given state and the decision variables, as $z_{t,d} = J_{t,d,0,0} - \sum_{b \in B \setminus \{0\}} X^V_{t,d,0,b}\ \forall d \in D$. Our goal is to find the policy that minimizes logistics costs over our planning horizon. Therefore, we define a policy $\pi \in \Pi$ as a function $\pi : S_t \rightarrow x_t$ that maps each state to a corresponding decision. The optimal policy $\pi^*$ may be found by solving the well-known Bellman optimality equations for each state: $V_t^{\pi^*}(S_t) = \min_{x_t \in X(S_t)} \{C(S_t, x_t) + \sum_{S_{t+1} \in S} \mathbb{P}(S_{t+1} | S_t, x_t) V_{t+1}^{\pi^*}(S_{t+1})\}\ \forall S_t \in S$.

### 3.2 Approximate Value Iteration Algorithm

This section presents the Reinforcement Learning approach to approximately solve the MDP formulation from Section 3.1. More specifically, we use approximate value iteration (Powell 2011; Sutton and Barto 2018), as outlined in Algorithm 1. Before explaining the steps of this algorithm, we first introduce some notation. We formulate the value functions around the post-decision state $S_t^x$, which is defined as the state of the system directly after a decision $x_t$ has been made but before the arrival of the next-stage exogenous information $W_{t+1}$. The transition from a state $S_t$ to a post-decision state $S_t^x$ is given by the transition function $S^{M,x}(S_t, x_t)$. After the arrival of the exogenous information, we have the transition $S^M(S_t^x, W_{t+1})$.

The expected value $\overline{V}_t(S_t^x)$ of a post-decision state $S_t^x$, i.e., the value function approximation (VFA), is given by a linear regression model using a set of features $\phi_t^f(S_t^x)\ \forall f \in F$ with corresponding weights $\theta_t^f\ \forall f \in F$, which are iteratively updated. The downstream cost $\hat{v}_t$ provides the direct reward $C(S_t, x_t)$ plus the approximated downstream costs of the post-decision state $\overline{V}_t(S_t^x)$. That is, a one-step look-ahead with a bootstrap estimate (Sutton and Barto 2018). The input data are the number of iterations $N$, the feature set $F$, the value $\varepsilon$, the learning rate $\gamma$, and the initial values for $\overline{V}, \hat{v}$, and $\theta^f$. The output data are the learned weights $\theta^f\ \forall f \in F$, which indirectly determine the policy through the VFA $\overline{V}_t(S_t^x)$.

**Algorithm 1:** Approximate Value Iteration

**Data:** $(N, F, \varepsilon, \gamma, \overline{V}, \hat{v}, \theta^f)$
$\overline{V}_0, \hat{v}_0, \theta_0^f \leftarrow$ *Initialize()*, $\forall\ f \in F$
$n = 1$
**for** $n < N$ **do**
  **for** $t < T$ **do**
    **if** $t > 0$ **then**
      $\hat{v}_t = \min_{x \in X_t} \left\{ C(S_t, x_t) + \gamma\, \overline{V}_t \left( S^{M,x}(S_t, x_t) \right) \right\}$
      $\theta_t^f = \leftarrow$ *update*$(\theta_{t-1}^f, \phi_{t-1}^f, \hat{v}_t), \quad \forall\ f \in F$
    $\tilde{x}_t \leftarrow \varepsilon$-*greedy*$(X_t)$
    $S_t^x = S^{M,x}(S_t, \tilde{x}_t)$
    $\phi_t^f \leftarrow$ *Compute*$(S_t^x), \forall\ f \in F$
    $W_{t+1} \leftarrow$ *Random*$(\Omega)$
    $S_{t+1} = S^M(S_t^x, W_{t+1})$
return $\theta^f\ \forall\ f \in F$

Algorithm 1 starts by setting initial values for $\overline{V}_0, \hat{v}_0, \theta_0^f$, and $n$ (lines 1 - 2). Next, line 3 loops over $N$ training iterations and line 4 loops over $T$ stages. When the current stage is different from 0, lines 6-7 update the weights of the linear regression model. More specifically, line 6 computes the expected downstream costs $\hat{v}_t$ of the best possible decision given our current knowledge, i.e., the lowest sum of direct costs plus expected downstream costs. Line 7 updates the feature weights $\theta_t^f$ based on the least-squares error (Powell 2011) between $\overline{V}_{t-1}$ (given by $\theta_{t-1}^f$ and $\phi_{t-1}^f$) and $\hat{v}_t$. In Line 8, a decision $\tilde{x}_t$ is chosen based on the $\varepsilon$-greedy decision policy (Powell 2011). The system then transitions towards the post-decision state $S_t^x$ in line 9. Next, in line 10, we compute the feature weights $\theta_t^f$ corresponding with the current post-decision state $S_t^x$ such that we can update them in the next stage. In line 11, we generate the next-stage exogenous information $W_{t+1}$ and with this we transition to the next state $S_{t+1}$ in line 12.

To select a suitable set of features $F$, we analyzed the predictive value of a wide set of features. We first ran a long simulation and stored each feature value for all encountered states. Next, for each of the encountered states, we sum the costs over the states encountered in the subsequent ten stages, and chose the feature set resulting in the lowest error and the highest predictive power explaining the upcoming 10 stage horizon costs. Accordingly, the selected set of features consists of the number of available EAVs per battery level, the number of containers per release time, the number of urgent containers, i.e., that cannot wait one more stage, the number of non-urgent containers, the number of non-urgent containers per distance class, the total number of containers in the system, the number of newly arrived containers, the average distance class over all containers, and the total travel time to transport all containers at the PS.

## 4 SIMULATION MODEL

We perform a simulation study for the authority of the PoM to evaluate the performance of an EAV fleet for internal transport. For this, we follow the simulation framework described in Brunetti et al. (2020) and implement a simulation model of the PoM in Siemens Tecnomatix Plant Simulation. In this section, we introduce the PoM and present several characteristics and elements of the resulting detailed simulation model that make it a challenging environment for the RL algorithm, as the latter is trained in a simplified simulation environment based on the MDP model. Hence, we also describe our approach to adapt the stage-based approach of the MDP model to a discrete event simulation with a virtually continuous horizon.

The PoM is the fourth largest seaport in the Netherlands. It services approximately 14,000 vessels per year, for a total of 18.5 million tons of transshipment. In its area of 26.35 km$^2$, there are about 430 companies, of which more than a third are purely logistics companies. The main problem in PoM is its lack of maneuvering space on certain roads and intersections in the central area, leading to congestion during peak hours. Furthermore, trucks wait for an available docking bay in front of the LC gate, leading to more congestion, safety issues, and inefficiencies for the drivers, as they cannot leave or take an official break. In PoM, all freight flows, i.e., hub-to-hub, first-, and last-mile transport, rely on manned tractors or trucks and simple planning logic. We support the port authority by providing information on the potential impacts of decoupling inbound and internal freight flows by means of a PS and an EAV fleet, to ease congestion and improve hub-to-hub transport.

The simulation framework provides inputs, outputs, assumptions, process flowcharts, and a multi-agent system for the simulation of emerging technologies at logistics areas, e.g., EAV fleets. In the multi-agent system, independent LCs send transport requests to a central dispatcher, which, in this study, corresponds to the RL dispatching algorithm. To obtain accurate infrastructure, we create a 3D visualization of the PoM, shown in Figure 2, using the large-scale model generator from van Steenbergen et al. (2021) and the LC data from the port authority. Specifically, we model road flows and roadside operations at LCs, as road traffic is the current concern of the port authority. Moreover, we restrict ourselves to inbound shipments, as in the IEFDP. Therefore, the outbound process for containers is not modeled and it is assumed that trucks (i) leave the LCs after unloading their container or (ii) leave the PS right after decoupling their container. In Figure 2, we see four port entrances for road modalities (white squares with truck images), more than a hundred LCs (green squares), and the PS in the bottom left corner (blue rectangles). Furthermore, shipment

arrivals at the PoM are on the order of thousands per day, with hourly variability and peak hours. Last, the large-scale model generator allows us to achieve the level of detail of a micro-traffic simulation. This leads to stochastic travel times, disruptions, and general uncertainty about the future availability of EAVs. The resulting simulation environment is more complex than the training environment of the RL algorithm, i.e., the value-based RL model is trained in a Monte Carlo simulation of the corresponding MDP, and the resulting value functions are used within a more realistic simulation. The latter simulation drops several of the assumptions from Section 3. In fact, we can now charge an arbitrary number of battery levels (assumption 1); the route choice is no longer negligible, and the travel times are based on the actual traffic conditions in the simulation (assumptions 4); battery consumption is still based on the discretized distance class $d \in D$ of the LC, but LCs of the same distance class require different travel times based on their actual distance from the PS (assumption 5); and, lastly, the destination of a container is not randomly allocated but is proportional to the surface area ($m^2$) of the LCs, i.e., larger LCs receive more containers.

Figure 2: 3D simulation model of the Port of Moerdijk, the Netherlands.

We will now describe several elements of the simulation that are relevant for this study: the PS, the LCs, the EAVs and trucks, and the dispatch area for the EAVs. We refer to Brunetti et al. (2020) for further details.

The PS is modeled as a buffer area where trucks and containers are decoupled, containers are forwarded to the PS, and later coupled to the EAVs. When a container reaches the PS, its information is stored on a cloud-based platform, represented by a data table, and copied to the destination LC.

The LC is modeled as multiple docking bays, the number of which is based on its surface size. Operations at an LC are check-in, container unload and departure, and waiting at the entrance in case of congestion. The type of unload operation in the docking bay depends on the LC, where the terminals perform the transshipment of the containers, and the warehouses perform the unloading of the freight from the containers, thus requiring a longer processing time. Furthermore, the type of LC determines whether the LC is open 24/7, e.g., at terminals, or working during limited shifts, e.g., at warehouses.

Trucks, containers, yard tractors, and EAVs are represented as 3D objects with realistic sizes, scaled down to the scale of the automatically generated infrastructure. EAVs and trucks can (de)couple containers and select the fastest route to the container's destination. The fastest route is automatically calculated by Tecnomatix Plant Simulation considering both the length of each segment of the potential route, and the speed limit of each type of road. Truck arrivals are generated at the beginning of the day and scheduled for each hour, based on historical data and a fitted distribution. Lastly, containers are generated with a

destination, a due date and an earliest availability for transport, as in the formulation of Section 3. Figure 3 shows trucks (red), containers (white), and EAVs (black and white tractors without driver cabin) in our simulation environment, at two intersections of the 3D model.

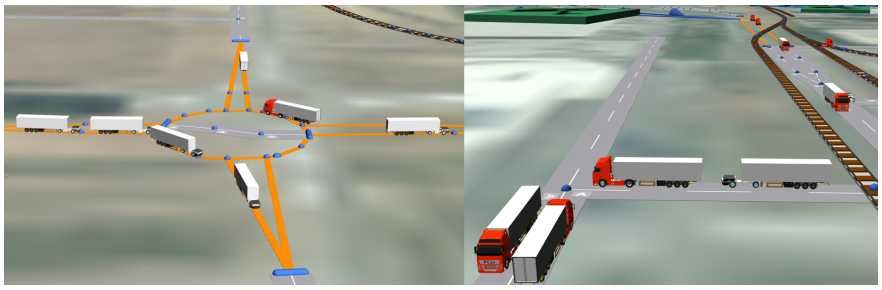

Figure 3: Trucks, containers, and EAVs at a roundabout (left) and a rail crossing (right).

Finally, the dispatch area for the EAVs is next to the PS and includes multiple charging stations. Idle EAVs wait here for their next transport or charging job. When assigned to a transport job, the EAV drives to the PS, picks up a container, drives to the corresponding LC, decouples the container, and drives back to the dispatch area. The EAV battery is consumed on the basis of the distance class of the LC, which is simply calculated by discretizing the maximum distance between all LCs and the PS in the desired number of classes. When assigned to a charging job, the EAV waits at a charging station for the inputted charging time. Currently, the charging process takes a fixed time, but it could be extended to be stochastic and dependent on the charging targets, e.g., from battery level $a$ to battery level $b$. After charging, the battery level is updated and the EAV is immediately available for dispatch.

In this discrete event simulation of the PoM, the processing times at LCs and travel times are continuous variables, just as in reality. However, this causes a mismatch with the stage-based simplified simulation we use to train the RL algorithm. In the detailed simulation, the processing times for containers are computed with triangular distributions derived from experts at the main LCs of the port. Regarding travel times, note that their stochastic duration is emerging from several factors: the interactions of vehicles on the road, the route length, and the speed limits over different route segments. Moreover, operations at the PS or LCs may further delay the return of an EAV to the dispatch area. Therefore, these unloading and transshipment operations can take longer or shorter than the time corresponding to one stage of the MDP. To merge the discrete event simulation environment and the stage-based simulation relying on the MDP formulation, we schedule recurring dispatch choices over the discrete-event horizon based on the desired duration of a stage. Also, we update the number of EAVs per battery level before the dispatch choice. The RL dispatcher only assigns EAVs in the dispatch area to transport or charge. It does not consider EAVs that are currently performing other tasks, although the system could be extended to consider multiple scheduled jobs per EAV. After the dispatch choice, we update the due dates of the containers and count the amount of late containers. Then, the amount of late containers is also checked when a container reaches the destination LC, by transforming the stage-based due date in a date-time format, i.e., by multiplying the number of stages by the duration of one stage. Note that reducing the duration of one stage, while maintaining the total length of due dates, would reduce the misalignment between the discrete-event and stage-based simulation models, although it could increase the computational burden of the simulation.

## 5 EXPERIMENTS

In this section, we present the numerical results of our experiments. First, we provide the experimental settings and parameters both for the training of the RL algorithm and for the port simulation. Then, Section 5.1 compares the performance of the RL algorithm to different assignment heuristics, while Section 5.2 analyzes the impact of the RL approach in combination with an EAV fleet in the simulation model of

the PoM. The experiments were executed on a computer equipped with a 1.90 GHz Intel(R) Core(TM) i7-8665U, 16 GB of RAM, and running Windows 10 in 64-bit mode.

Regarding the training of the RL algorithm, the MDP settings considered in the instance of Section 5.1 are defined by 96 stages ($T$), 3 location types ($D$), a maximum length of 4 release periods ($R$), a maximum time-window length of 6 periods ($K$), 9 battery levels, 1 charging level per stage, 12 EAV ($V$), 9 charging stations ($Q$), a container arrival rate of 12 per hour ($\lambda$), and 1 cost unit per delayed container ($C^L$). The RL algorithm converges to a relatively stable estimate of the costs in 1000 iterations, and within reasonable computational times.

Regarding the PoM simulation, we provide additional information on the location of the PS, the time to (de)couple containers at the PS, and the time to process a container at the LCs. We locate the PS in the south-west corner of the port area, close to the highway entrance, which leads to more trucks entering the port from the south-west entrance. For the (de)coupling time, we consider ten minutes for EAVs and four minutes for trucks, which also includes the time to drive through the PS to the selected container. These values were obtained by a preliminary study of the PS concept. For container processing time, we use triangular distributions ranging between 30 and 90 minutes with a mode of 60 for warehouses, and between 15 and 60 minutes with a mode of 20 for terminals. The data were obtained by experts at the main LCs of the PoM, considering both variability and a difference in operations: warehouses usually unload containers, while terminals mostly perform transshipment to other modes. Due to some LCs operating 24/7, i.e., the terminals, and to the truck arrivals varying over the day, we have a non-terminating simulation with steady-state cycles over the day. However, if no major disruptions occur during the day, the new day begins without any serious backlog. Therefore, we run a simulation of 11 days for both scenarios, where the first day is regarded as warm-up, and the next ten days of data are treated as independent replications. The MDP settings studied in the simulation of Section 5.2 differ from the RL training instance in the number of stages, which are 48, the maximum time-window length of 10 periods, 10 battery levels, full charging per stage, 100 EAVs, 25 charging stations, and a container arrival rate at the PS ranging between 1 and 60 per hour depending on the hour of the day. In addition to these instance parameters, the recurring EAV dispatch and charge described in Section 4 occurs every 30 minutes, achieving the 48 stages per day.

We perform two sets of experiments. The first set is presented in Section 5.1 and compares the value-iteration RL algorithm with four assignment heuristics with the purpose of validating the performance of our RL approach. The second set of experiments is presented in Section 5.2 and shows the combined performance of a PS, an EAV fleet, and the RL dispatching algorithm in a detailed simulation model of the Port of Moerdijk. The main goal is to evaluate the impact on the freight flows as well as the effective management of EAVs.

### 5.1 Comparison of the Reinforcement Learning Algorithm and the Heuristics

This set of experiments compares the performance of our RL approach with four different operational strategies, which we implement and test as heuristics in C++. The first heuristic algorithm chooses a *random* decision at every stage. The second is a *risky heuristic* that always prioritizes the transport of containers and sends EAVs to charge the battery only when they run out of power, or need to wait. The third is a *conservative heuristic* that always prioritizes charging the battery of the EAVs and sends them to transport containers only when the battery stations are occupied or the batteries are already full. The last algorithm is a *flexible heuristic* that charges a predefined percentage of EAVs based on the average battery of the fleet, then the rest of the vehicles are sent to deliver containers, if possible. The metrics upon which we compare the different approaches are costs (late containers), standard deviations in costs, and running time, as shown in Table 1. Costs and running times are averaged over 2000 replications.

Results show that the RL algorithm outperforms the benchmark heuristics. For the whole set of algorithms, we see a standard deviation smaller than or equal to 6.97, corresponding to the random selection algorithm, while the RL algorithm results in the smallest standard deviation of 5.62. This shows that the dispersion of the RL algorithm's performance with respect to its objective value is stable. On the

Table 1: Comparison of algorithmic performance.

| Algorithm | Final Cost | Std. Deviation | Running time |
|---|---|---|---|
| Random selection | 542 | +/- 6.97 | 5.81 s |
| Risky heuristic | 484 | +/- 6.28 | 4.08 s |
| Conservative heuristic | 518 | +/- 6.55 | 3.04 s |
| Flexible heuristic | 459 | +/- 5.79 | 3.59 s |
| **RL algorithm** | **435** | **+/- 5.62** | **58.26 s** |

other hand, the running times of the heuristic algorithms do not exceed 6 seconds, whereas the running time of the RL algorithm is up to 58.26 seconds. Consequently, for this set of experiments, we conclude that the RL algorithm is able to outperform the heuristic policies by exploring the solution space and learning from the addressed scenarios in computational times under 60 seconds.

### 5.2 Simulation of the Port of Moerdijk

Here, we compare two scenarios for inbound logistics at the PoM: the as-is scenario of direct truck access and the decoupled scenario of the PS plus EAV fleet with the RL dispatcher. In the first, the inbound trucks drive directly to their destination, without information on the situation at the LC. In the second, the trucks drive to the PS, drop their containers, and leave the port area, while the EAVs pick up the containers and transport them to their destination. In relation to the scenarios, we mention here the percentage of trucks decoupling. Not all trucks (and shipping companies) may accept to stop at the PS and let an EAV transport their shipment. In addition, servicing all trucks entering the port would require both a large PS and a large EAV fleet. Therefore, for this study, we simulate the decoupling of one-third of the trucks, while the remaining two-thirds drive directly to the LCs, creating a mixed-traffic environment for our study.

We show results for the freight flows and for the port as a whole in Table 2 and Table 3, respectively. We compare throughput times and waiting times, both for containers and inbound trucks, as well as the number of late containers and the average speed and total distance traveled in the port for trucks and EAVs. The simulation of the as-is scenario took 415 seconds whereas the simulation of the decoupled scenario took 2304 seconds, for the whole set of replications.

Table 2: Comparison of freight flows impacts for the two scenarios.

| Scenario | Freight Flow | Throughput Time | Waiting Time LC | Waiting Time PS |
|---|---|---|---|---|
| Direct | Trucks | 02:09:33 +/- 00:03:48 | 01:23:51 +/- 00:20:48 | - |
| | Containers | 02:03:05 +/- 00:03:54 | 01:23:51 +/- 00:20:48 | - |
| Decoupled | Trucks to LC | 02:08:16 +/- 00:03:34 | 00:56:46 +/- 00:21:30 | - |
| | Trucks to PS | 00:17:13 +/- 00:00:05 | - | 00:00:00 +/- 00:00:00 |
| | Containers | 02:24:30 +/- 00:02:44 | 00:56:46 +/- 00:21:30 | 01:35:55 +/- 00:07:50 |

From the results in Table 2, we see that the freight flows are both positively and negatively affected by the new operational scenario. On the one hand, decoupling reduces the waiting time of the trucks driving directly to the LCs and the total time that the decoupling trucks spend in the port. This makes the stay in the PoM more efficient for truck drivers and improves the valuation of the port by logistics partners. Also, containers spend less time waiting at LCs, smoothing operations and reducing congestion at the latter. On the other hand, containers now take longer to be delivered and processed, due to the time to decouple and the slightly longer waiting time at the PS. Last, our model is further validated by (i) the difference between the throughput times of trucks and containers in the as-is scenario, which corresponds to the additional time for trucks to leave the port; and (ii) the matching waiting times for trucks and containers when waiting at the LCs, as the containers waiting at the LCs are coupled to the truck and, in the decoupling scenario, did not stop at the PS.

Table 3: Comparison of port-wide impacts for the two scenarios.

| Scenario | Vehicle Type | Late Containers | Std. Dev. | Traffic Speed | Distance |
|---|---|---|---|---|---|
| Direct | Trucks | 39.25 (0.94%) | +/- 10.58 (0.2%) | 9.10 m/s | $2.84*10^7$ km |
| Decoupled | Total | 57.13 (1.38%) | +/- 12.19 (0.28%) | 6.56 m/s | $3.04*10^7$ km |
| | Trucks | 39.75 (0.96%) | +/- 11.11 (0.28%) | 7.48 m/s | $1.79*10^7$ km |
| | EAVs | 17.38 (0.42%) | +/- 5.83 (0.26%) | 4.69 m/s | $1.25*10^7$ km |

From the results in Table 3, we see that decoupling slightly increases the number of late containers, as a result of the longer throughput and waiting times. In addition, we see that the total average traffic speed is reduced by almost 28%, due to the EAVs performing part of the trips and also slowing down the trucks on the road, as our micro-traffic simulation does not allow overtaking. The slower traffic speed also contributes to the longer container throughput time. However, a lower traffic speed may improve traffic safety in the port. Finally, we see that the total distance traveled increased slightly and that 41% of those kilometers are performed by an EAV, potentially reducing $CO_2$ emissions.

To conclude, decoupling inbound trucks and internal logistics by means of a PS and an EAV fleet, even though it is used by only one-third of the arrivals, could lead to many benefits with regards to improving traffic flows, the valuation of the logistics area by external logistics partners, and the sustainability of logistics at the port, while reducing congestion at LCs. However, the current process design and RL dispatcher slightly increase the length of stay of containers at the port, potentially resulting in lateness.

## 6 CONCLUSIONS AND FUTURE WORK

We created a detailed simulation model of the Port of Moerdijk, the Netherlands, to evaluate the benefits of a fleet of electric automated vehicles (EAVs) transporting containers from a parking site to their corresponding logistics companies, while meeting time-window and battery constraints. The dispatching of EAVs is introduced as the Internal Electric Fleet Dispatching Problem (IEFDP).

We addressed the IEFDP by providing a Markov decision problem formulation and developing an approximate value iteration-based reinforcement learning (RL) algorithm in a simplified environment. After training the RL algorithm, we used it as a decision-making tool in the detailed simulation of internal logistics at the Port of Moerdijk. Here, travel times are stochastic due to an accurate representation of traffic flows, and operational constraints at logistics companies make the environment more dynamic and uncertain. We validated the effectiveness of our RL dispatcher against different human-like strategies, which our dispatcher outperforms. Then, in the detailed simulation, the RL-assisted fleet together with a pre-gate parking site leads to reductions in truck waiting times and traffic congestion, as well as a range of potential benefits, e.g., sustainability. However, the total length of stay of containers increases slightly, due to inefficiencies in the (de)coupling process.

Regarding future research, our aim is to extend the integration of the RL algorithm into the discrete event simulation environment. First, we want to shift from a hybrid stage-based horizon to a complete event-driven horizon, reducing the information mismatch. Second, we aim to speed up the RL dispatcher by considering situational information from the simulation, e.g., by discarding supposedly sub-optimal actions. Finally, we aim to test other RL approaches, e.g., based on policy iteration or the use of neural networks, which could perform better than value iteration in a realistic environment.

## ACKNOWLEDGMENTS

This research has been funded by the Dutch Research Council (NWO) and TKI Dinalog as part of the CATALYST Living Lab (reference 439.18.458 B), of which the Port of Moerdijk is a partner. The CATALYST project is coordinated by The Netherlands Organization for Applied Scientific Research (TNO).

## REFERENCES


Asadi, A., and S. N. Pinkley. 2021. "A Stochastic Scheduling, Allocation, and Inventory Replenishment Problem for Battery Swap Stations". *Transportation Research Part E: Logistics and Transportation Review* 146:102212.

Behdani, B., B. Wiegmans, V. Roso, and H. Haralambides. 2020. "Port-Hinterland Transport and Logistics: Emerging Trends and Frontier Research". *Maritime Economics & Logistics* 22(1):1–25.

Brunetti, M., M. Mes, and J. van Heuveln. 2020. "A General Simulation Framework for Smart Yards". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, R. Thiesing, and T. Roeder, 2743–2754. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Evers, J. J. 2006. "Real-Time Hiring of Vehicles for Container Transport". *European Journal of Transport and Infrastructure Research* 6(2):173–198.

Heilig, L., E. Lalla-Ruiz, and S. Voß. 2017. "Port-IO: An Integrative Mobile Cloud Platform for Real-Time Inter-Terminal Truck Routing Optimization". *Flexible Services and Manufacturing Journal* 29(3):504–534.

Heilig, L., and S. Voß. 2017. "Inter-Terminal Transportation: An Annotated Bibliography and Research agenda". *Flexible Services and Manufacturing Journal* 29(1):35–63.

Hu, Q., F. Corman, B. Wiegmans, and G. Lodewijks. 2018. "A Tabu Search Algorithm to Solve the Integrated Planning of Container on an Inter-Terminal Network Connected with a Hinterland Rail Network". *Transportation Research Part C: Emerging Technologies* 91:15–36.

Hu, Q., B. Wiegmans, F. Corman, and G. Lodewijks. 2019. "Critical Literature Review into Planning of Inter-Terminal Transport: In Port Areas and the Hinterland". *Journal of Advanced Transportation* 2019:1–15.

ITF 2021. *ITF Transport Outlook 2021*. Paris: OECD Publishing.

Le-Anh, T., and M. De Koster. 2006. "A Review of Design and Control of Automated Guided Vehicle Systems". *European Journal of Operational Research* 171(1):1–23.

Powell, W. B. 2011. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Hoboken, New Jersey: John Wiley & Sons.

Schneider, M., A. Stenger, and D. Goeke. 2014. "The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations". *Transportation science* 48(4):500–520.

Sutton, R. S., and A. G. Barto. 2018. *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: MIT press.

van Steenbergen, R., M. Brunetti, and M. Mes. 2021. "Network Generation for Simulation of Multimodal Logistics Systems". In *Proceedings of the 2021 Winter Simulation Conference*, edited by S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, 1537–1548. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Vdovic, H., J. Babic, and V. Podobnik. 2019. "Automotive Software in Connected and Autonomous Electric Vehicles: A Review". *IEEE Access* 7:166365–166379.

Zheng, H., W. Xu, D. Ma, and F. Qu. 2021. "Dynamic Rolling Horizon Scheduling of Waterborne AGVs for Inter Terminal Transportation: Mathematical Modeling and Heuristic Solution". *IEEE Transactions on Intelligent Transportation Systems* 23(4):3853–3865.


## AUTHOR BIOGRAPHIES


**MATTEO BRUNETTI** is a PhD candidate within the Industrial Engineering and Business Information Systems section at the High Tech Business and Entrepreneurship department at the University of Twente, The Netherlands. He received a MSc in Industrial Engineering and Management in 2019. His research interests are supply chain management, logistics digitalization, discrete event simulation, simulation optimization, and artificial intelligence. His email address is m.brunetti@utwente.nl.

**GIOVANNI CAMPUZANO** is a PhD candidate within the Industrial Engineering and Business Information Systems section at the High Tech Business and Entrepreneurship department at the University of Twente, The Netherlands. He holds a master's (2018) and a bachelor's degree in Industrial Engineering (2018), University of Bío-Bío, Chile. His research interests lie in the fields of logistics, operations research, mathematical programming, stochastic programming, reinforcement learning, and artificial intelligence. His email address is g.f.campuzanoarroyo@utwente.nl.

**MARTIJN R.K. MES** is an Associate Professor within the Industrial Engineering and Business Information Systems section at the High Tech Business and Entrepreneurship department at the University of Twente, The Netherlands. He holds a MSc in Applied Mathematics (2002) and a PhD in Industrial Engineering and Management at the University of Twente (2008). After finishing his PhD, Martijn did his postdoc at Princeton University, Department of Operations Research and Financial Engineering. His research interests are transportation, multi-agent systems, stochastic optimization, discrete event simulation, and simulation optimization. His email address is m.r.k.mes@utwente.nl.