Comparison of Manual and Automated Decision-Making with a Logistics Serious Game

Martijn Mes(✉) and Wouter van Heeswijk

Department of Industrial Engineering and Business Information Systems, University of Twente, Enschede, The Netherlands
m.r.k.mes@utwente.nl

Abstract. This paper presents a logistics serious game that describes an anticipatory planning problem for the dispatching of trucks, barges, and trains, considering uncertainty in future container arrivals. The problem setting is conceptually easy to grasp, yet difficult to solve optimally. For this problem, we deploy a variety of benchmark algorithms, including two heuristics and two reinforcement learning implementations. We use the serious game to compare the manual performance of human decision makers with those algorithms. Furthermore, the game allows humans to create their own automated planning rules, which can also be compared with the implemented algorithms and manual game play. To illustrate the potential use of the game, we report the results of three gaming sessions: with students, with job seekers, and with logistics professionals. The experimental results show that reinforcement learning typically outperforms the human decision makers, but that the top tier of humans come very close to this algorithmic performance.

Keywords: Intermodal transport · Serious gaming · Reinforcement learning · Approximate dynamic programming · Heuristics

1 Introduction

Despite the development of sophisticated logistics planning algorithms to automate decisions and the increasing availability of real-time data, planning in the logistics sector still heavily relies on human planners. There are several reasons why manual planning is often preferred. First, human planners require decisions to be sensible and explicable [18]. They tend to quickly lose faith in decision support algorithms when presented with counter-intuitive suggestions, even if consistently following the decision rules would work well in the long term. Second, humans are able to perform surprisingly well on certain planning tasks with vast numbers of potential solutions [8,25], therefore not always recognizing the benefits of automated planning. Third, the algorithmic expertise and experience of logistics companies with sophisticated decision support systems is

E. Lalla-Ruiz et al. (Eds.): ICCL 2020, LNCS 12433, pp. 698–714, 2020.
https://doi.org/10.1007/978-3-030-59747-4_45

often limited, yet experience is a key determinant for successfully introducing new technologies [23].

Serious games may overcome some of the aforementioned challenges. Such games can be used to educate and stimulate a "mental switch" towards decision support and automated planning within the logistics sector. They offer an opportunity for (future) planners to learn about the challenges involved in logistics planning, to gain experience with new technologies, and to become convinced about the advantages such technologies may bring. In addition, games may demonstrate the benefits of decision support, increasing faith in such systems by experimenting in an environment without real-life consequences. Finally, they illustrate the use of automated planning rules and the way such rules can be designed. Venkatesh [22] shows that game-based training is more effective than other forms of training with respect to user acceptance of new technologies. He also indicates that the effects of perceived usefulness and perceived ease of use of a new system – the main drivers behind technology acceptance – are stronger for people who have enjoyed game-based training.

This paper presents a logistics serious game that mimics an anticipatory planning problem in intermodal transport, considering uncertainty in future container arrivals. This anticipatory planning problem is based on Pérez Rivera and Mes [13] and can formally be defined as a Delivery Dispatching Problem (DDP) [12]. In its simplest form, this single-player game with group competition simulates a logistics service provider (LSP) that needs to assign containers to trucks, barges, and trains on a daily basis. The game consists of various predefined scenarios (varying in number of containers, container characteristics, and costs structures) that can be modified by a game master. For the planning algorithms, we make use of two heuristics and two implementations of reinforcement learning (RL). Implementation of these algorithms in practice is not always easy, especially for RL, due to the aforementioned lack of experience from the human planners and due to often being perceived as a 'black box' [18]. To address this phenomenon, we compare the performance of various algorithms with the human performance. Depending on the game mode, human players can either plan the containers manually, use decision support from our RL algorithms, or create their own automated planning rules.

The remainder of this paper is structured as follows. Section 2 positions our research in the body of existing literature and highlights its contributions. In Sect. 3, we describe the decision problem as represented in the game. Section 4 discusses the various solution methods deployed in this paper. The solution methods can all be evaluated using our serious game as introduced in Sect. 5. The experiments with this game are described in Sect. 6. We end with conclusions in Sect. 7.

2 Literature

This literature section is composed as follows. First, we present some related serious games. Second, we reflect on several aspects of human decision-making that are relevant to logistics planners, and highlight various experiments that have been conducted on human performance in this context, including the use of serious games. Third, we briefly describe the reinforcement learning framework that we use as a benchmark as well as for providing decision support to the human. Fourth, we discuss the anticipatory planning problem and several reinforcement learning algorithms that have been developed for variants of this problem.

In the transportation domain, serious games have mostly been developed for raising awareness about the interaction among different actors in a transportation system [17]. For example, the rail cargo management game [11] simulates the interaction among transporters, clients, and network managers. Games about training a single-actor are scarce and focus mostly on passenger or public transport as seen in the review of Raghothama and Meijer [16]. For example, Ecodealers and Waze are two location-based games (using a mobile phone) described in Rossetti et al. [17], where a single player is trained for the improvement of public and passenger transport, respectively. Similarly, Drakoulis et al. [4] considered an interactive game to motivate public transport users to participate and behave correctly with the proposed demand-responsive transport service. Examples of single player games that are closely related to ours are SynchroMania [3], the follow-up game MasterShipper, and the Modal Manager game [10].

In human decision-making, bounded rationality has a major impact on everyday decisions. Due to time pressure and cognitive limitations, humans often settle for an acceptable solution rather than the optimal one [19]. In this spirit, so-called *fast-and-frugal* heuristics are often used to make decisions [5]. Such decisions are constructed out of three building blocks, requiring limited information and processing: (i) search rules that specify how information is collected, (ii) stopping rules that specify when the information search is halted, and (iii) decision rules that specify which decision should be selected. Heuristic algorithms, as often used in logistics, reflect such intuitive human decision-making methods, yet are faster (especially for large problem instances) and more consistent.

We may extract some insightful results from experiments with humans in logistics planning problems. For example, experiments have been performed with the MIT 'Beer Distribution Game', in which multiple participants manage part of a simulated inventory distribution system [20]. Human performance is rather poor, with identified causes being anchoring (initial stock levels strongly influence decisions later on), failure to take into account time lags in supply, and a tendency to perceive controllable events as external influences. Other experiments consider capacited VRPs with time windows, for which Anderson *et al.* [1] show that humans – with visual aid – are able to find good solutions. For the Traveling Salesman Problem (TSP), Wiener *et al.* [25] conclude that humans tend to quickly identify good general solution structures, which they subsequently refine. In Kefalidou and Ormerod [7] it is shown that human participants came close to optimal VRP solutions, especially those relying on visual solution methods. Finally, Keller and Katsikopoulos [8] evaluate various human solution approaches in operations management, claiming that such methods are particularly effective when the model contains a single attribute that typifies good solutions (e.g., efficient capacity utilization). Furthermore, the authors state that humans perform well after relatively few trials, implying a steep learning curve. However, they also identify diminishing performance when dealing with larger problem instances. Summarizing, experimental studies show that humans perform well with visual cues and clear solution structures, but performance quickly declines when facing larger and more complex tasks.

Whereas logistics planners rely on manual planning for many decisions, sophisticated planning algorithms are widely spread in the industry as well. This paper considers an anticipatory planning problem, which may be formalized mathematically by a Markov Decision Process (MDP) that in theory may be solved to optimality. However, for realistic-sized problems in logistics planning this is typically not feasible [15]. To overcome computational limitations in solving the model, reinforcement learning (RL) techniques are often applied. For comprehensive and broad descriptions of RL, we refer to Powell [15] and Bertsekas [2]. RL exploits Monte Carlo simulation techniques to learn value functions by observation. Often Value Function Approximation (VFA) is used to computationally simplify the problem; for human designers the key challenge is to define a set of attributes (explanatory variables) that help estimating the downstream costs. In our serious game, we use this VFA to offer the player decision support.

The anticipatory planning problem that we study in this paper may formally be defined as a Delivery Dispatching Problem (DDP) [12]. The DDP involves one or more capacitated transport modalities that may be dispatched at certain decision moments. Containers with stochastic properties arrive dynamically over time and must be dispatched to their destination using the available transport modalities. Costs are minimized by utilizing the modalities' capacity as efficiently as possible, both considering the currently available containers and anticipating future arrivals. Many variants of the DDP exist; we build upon the variant presented by Pérez Rivera and Mes [13], involving a barge that can carry multiple containers and trucks that serve as alternative transport modality. For closely related DDP variants, Pérez Rivera and Mes [14] and Van Heeswijk *et al.* [21] present several baseline heuristics composed of simple decision rules that achieve consolidation, yet these heuristics ignore various considerations that humans would intuitively make. The aforementioned papers also develop VFA-based RL algorithms, testing a variety of attribute designs. In contrast, Voccia *et al.* [24] and Klapp *et al.* [9] present RL algorithms that utilize policy rollout, relying on scenario sampling to estimate downstream costs.

The main contribution of this paper is that we quantitatively compare between human performance and a variety of logistics planning algorithms, utilizing a serious game for this purpose. Such a comparison has not yet been made for the DDP. Furthermore, we provide insights into the drivers of human performance for the DDP and how well they are able to translate their own decision-making strategy into automated planning rules.

3 Problem Description

We formalize the DDP for intermodal transport, which forms the basis of our serious game, as a Markov Decision Process (MDP) model. The objective is to minimize the dispatching costs over a planning horizon of T days (representing decision moments), with dispatching decisions being made at day $t \in \mathcal{T} = \{0, \dots, T\}$. New containers arrive daily; they are characterized by a destination $d \in \mathcal{D}$, a release day $r \in \mathcal{R} = \{0, 1, \dots, R^{max}\}$ relative to t, and a time window length $k \in \mathcal{K} = \{0, 1, \dots, K^{max}\}$ relative to r. Containers with $r > 0$ are deterministically known to arrive at $t + r$, but cannot be dispatched until $r = 0$. Each container characterized by $r = 0$ must be dispatched at or before $t + k$ to guarantee timely delivery; the window length only starts decreasing when $r = 0$.

Every day t, we decide which containers to transport by which transport modality $m \in \mathcal{M}$: trucks ($m = 0$), a barge ($m = 1$), and a train ($m = 2$). Capacities are restricted by Q^m. The costs for both barge and train transport are given by a fixed component depending on the subset of destinations visited and variable costs depending on the number of containers per destination visited. However, the train can visit only one destination per day. Economies of scale apply to both barge and train, as there is a fixed cost component, but the variable costs per container are lower than for the truck. Trucks have a capacity of one container ($Q^0 = 1$), and we assume the number of trucks is infinite. A container with $r = 0$ and $k = 0$ must always be transported; the infinite truck fleet guarantees that each container can be delivered on time. The container dock also has infinite capacity. Each unique combination of destination, release day, and time window length constitutes a *container type*. The problem state $S_t \in \mathcal{S}$ is a vector that describes the number of containers of each type:

$$S_t = [S_{t,d,r,k}]_{\forall [d,r,k] \in \mathcal{D} \times \mathcal{R} \times \mathcal{K}}.$$

At each day $t \in \mathcal{T}$, we decide how many containers of each type to dispatch, and whether we deliver them by trucks, barge or train. Recall that only containers with a release day $r = 0$ can be dispatched, that containers with time window $k > 0$ do not need to be dispatched, and that each dispatched container must be assigned to a transport modality $m \in \mathcal{M}$. Let $\mathcal{X}(S_t)$ be the set of feasible decisions while being in state S_t, and $x_t \in \mathcal{X}(S_t)$ the decision defined by:

$$x_t = [x_{t,m,d,k}]_{\forall [m,d,k] \in \mathcal{M} \times \mathcal{D} \times \mathcal{K}},$$

s.t.

$$\begin{aligned}
x_{t,m,d,k} &\leq S_{t,d,0,k} & \forall [m,d,k] &\in \mathcal{M} \times \mathcal{D} \times \mathcal{K}, \\
\sum_{m \in \mathcal{M}} x_{t,m,d,0} &= S_{t,d,0,0} & \forall d &\in \mathcal{D}, \\
\sum_{[d,k] \in \mathcal{D} \times \mathcal{K}} x_{t,m,d,k} &\leq Q^m & \forall m &\in \mathcal{M}, \\
x_{t,m,d,k} &\in \mathbb{N} & \forall [m,d,k] &\in \mathcal{M} \times \mathcal{D} \times \mathcal{K}.
\end{aligned}$$

Each decision x_t has an associated cost $C(S_t, x_t)$. We introduce some necessary notation. Let $c^f_{m,\mathcal{D}'}$ be the fixed costs for modality m for visiting the subset of destinations $\mathcal{D}' \subseteq \mathcal{D}$ (for the train $\mathcal{D}'$ consists of at most one destination, and for the truck the costs $c^f_{m,\mathcal{D}'}$ are always zero). Next, we define $c^v_{m,d}$ as the variable costs per container, i.e., the marginal costs of transporting a single container to destination d by modality m. Let $I_{m,\mathcal{D}'} \in \{0, 1\}$ be a binary variable indicating whether destination subset $\mathcal{D}' \subseteq \mathcal{D}$ is visited by modality m. To prevent postponing shipments until T and to consistently compare solution methods, we define so-called cleanup costs for the freight remaining at T, approximating the costs to dispatch the remaining containers at the end of the time horizon. We define cleanup costs with a function $C^T : (S_T, x_t) \mapsto \mathbb{R}^+$ that is added to the cost function, and $I_t \in \{0, 1\}$ the corresponding binary variable that only activates when $t = T$. The cost function at day t is defined as follows:

$$C(S_t, x_t) = \sum_{m \in \mathcal{M}} \left(\sum_{\mathcal{D}' \subseteq \mathcal{D}} I_{m,\mathcal{D}'} \cdot c^f_{m,\mathcal{D}'} \right) + \sum_{[m,d,k] \in \mathcal{M} \times \mathcal{D} \times \mathcal{K}} c^v_{m,d} \cdot x_{t,m,d,k} + I_t C^T(S_t, x_t),$$

s.t.

$$I_{m,\mathcal{D}'} = \begin{cases} 1 & \text{if} \quad \left(\prod_{d \in \mathcal{D}'} \left(\sum_{k \in \mathcal{K}} x_{t,m,d,k} \right) > 0 \right) \wedge \left(\sum_{[d,k] \in \mathcal{D} \setminus \mathcal{D}' \times \mathcal{K}} x_{t,m,d,k} = 0 \right), \\ 0 & \text{otherwise} \end{cases}$$

$$I_t = \begin{cases} 1 & \text{if} \quad t = T \\ 0 & \text{otherwise} \end{cases} .$$

We aim to find the policy that minimizes the costs over our planning horizon. A policy $\pi \in \Pi$ is a function $\pi : S_t \mapsto x_t$ that maps each state to a corresponding decision. The optimal policy π^* may be found by solving the well-known Bellman optimality equations for each state:

$$V_t^{\pi^*}(S_t) = \min_{x_t \in \mathcal{X}(S_t)} \left(C(S_t, x_t) + \sum_{S_{t+1} \in \mathcal{S}} \mathbb{P}(S_{t+1} \mid S_t, x_t) \cdot V_{t+1}^{\pi^*}(S_{t+1}) \right) \quad \forall S_t \in \mathcal{S}.$$

The problem definition supplied in this section enables us to outline our solution methods to tackle the anticipatory planning problem. A graphical illustration of the problem can be found in Fig. 1.

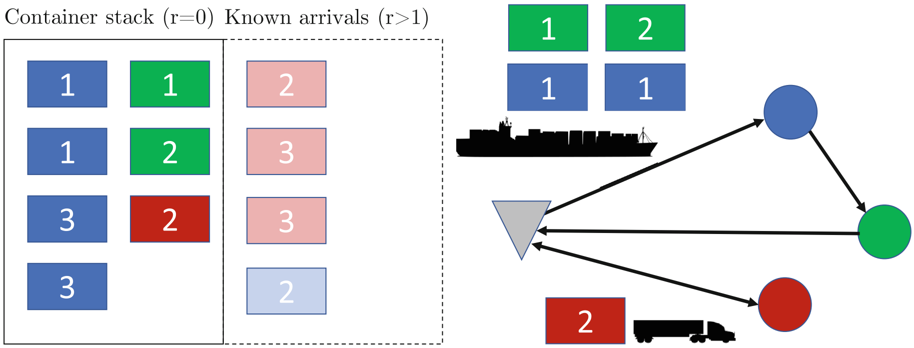

Fig. 1. Schematic representation of the problem. Left is the state, showing containers and their due dates. Right is a possible decision that allocates a subset of containers to barge and truck. Assuming a barge capacity of 8 containers, over 100 decisions exist for this example state, yet decision rules may be readily derived.

4 Solution Methods

This section discusses the following solution methods to the problem described in the previous section: (i) a heuristic policy, (ii) a myopic deterministic optimization policy, (iii) two reinforcement learning policies, (iv) human decision-making, (v) manually created planning rules, and (vi) human decision-making using decision support from one of the reinforcement learning policies. These methods are briefly described below. To ease the presentation, and because the train is not included as a modality in our experimental scenarios from Sect. 6, we only consider trucks and barges.

Heuristic. The heuristic follows three steps. In step 1 it calculates – for each possible destination for which we have an urgent container ($k = 0$) – the costs for filling up the remaining barge capacity using containers with this destination. For those destinations for which the barge is cheaper than truck, all urgent containers are assigned to the barge. In step 2, for all destinations already visited by the barge, remaining containers of the same destinations are also assigned to the barge, starting with those with smaller time windows, as long as there is capacity left. In step 3, we check which destination has the most containers remaining. We fill up the remaining barge capacity with these containers, starting with the lower time windows, when it is cheaper to transport these containers by barge compared to truck, and when either (i) the total number of containers with this destination, both released and non-released, is above a specified threshold or (ii) we do not have any non-released containers with this destination. The first condition accounts for future capacity problems when postponing the transport of containers. The second condition accounts for the idea that it is less likely that these containers can be consolidated more efficiently the next day.

Direct Cost Minimization. A flawed yet deliberate approach to tackle the planning problem is to simply minimize the direct costs at day t, without considering downstream effects: $\sum_{t \in \mathcal{T}} \min_{x_t \in \mathcal{X}_t(S_t)} C(S_t, x_t)$. When multiple decisions yield the same costs, we select the one that loads the highest number of containers onto the barge, thereby stimulating capacity utilization. This method corresponds to a naive application of traditional OR techniques such as linear programming, optimizing only for the information deterministically known at the decision moment.

Reinforcement Learning. Even for the relatively small problem instances considered in our game scenarios, it is not possible to solve the MDP model exactly. Therefore, we resort to the reinforcement learning (RL) framework to approximately solve the optimality equations through Monte Carlo simulations. More specifically, the algorithm performs N learning iterations, and in each iteration, we loop over the planning horizon $\mathcal{T}$ and use the resulting observations to improve our estimates of the downstream costs, thereby facilitating better decisions. Instead of storing estimates for each possible state (the so-called lookup table approach), we apply value function approximation (VFA) using the basis function approach, where we estimate the state-decision costs $V_t^n(S_t, x_t)$ by $\bar{V}_t^n(S_t, x_t) = \sum_{a \in \mathcal{A}} \theta_{t,a}^n \phi_a(S_t, x_t)$. Here $\phi_a : (S_t, x_t) \mapsto \mathbb{R}$ returns a given attribute from the state-decision pair (e.g., the total number of containers) that explains – to some extent – the cost of the state-decision pair. The attribute value is multiplied with a weight $\theta_{t,a}^n \in \mathbb{R}$. The weights $\theta_{t,a}^n, \forall [t, a] \in \mathcal{T} \setminus \{T\} \times \mathcal{A}$ are updated after every iteration. So, the value of a state is using a weighted linear combination of the basis functions; for more information we refer to Powell [15]. Our RL implementation is exactly the same as the one described in Pérez and Mes [14], except that we consider the single trip version, i.e., we only consider deliveries and not pickups.

Reinforcement Learning with Extended VFA. The extended RL algorithm only differs in the features used in the basis function approach. To explain the features, we introduce the notion of MayGo and MustGo containers. MustGo containers are those that must be transported today, i.e., containers with $r = 0 \wedge k = 0$. MayGo containers are those that we might transport today, i.e., containers with $r = 0 \wedge k > 0$. The basis function approach considered in our standard RL implementation uses the following features: (i) the number of containers per type (resulting in $|\mathcal{D}| \times |\mathcal{R}| \times |\mathcal{K}|$ features), (ii) the number of destinations having MustGo containers, (iii) the number of MustGo containers, (iv) the number of destinations with MayGo containers, (v) the number of MayGo containers, (vi) the number of destinations of the non-released containers, (vii) the number of non-released containers, and (viii) a constant. For our extended RL algorithm, we introduce additional features that are based on the set of containers that remain after applying our decision x_t. For each possible destination subset $\mathcal{D}'$, we introduce two binary variables indicating whether the subset may or must be visited the next day.

Manual Planning. For the human planners, the notion of bounded rationality applies. The planner has access to all cost information as well as the probabilities of container arrivals. However, as the decision time to solve the decision problem for each day is limited, players might cognitively not able to evaluate all feasible decisions. In addition, the human planner cannot possibly anticipate the potential downstream effects of their decisions. It is therefore expected that players seek a satisfactory solution rather than an optimal one. Note that in practical business settings, the combination of time restrictions and cognitive limitations would typically also apply. Humans are known to be good at decisions that involve clustering, this is reflected in the observed decision-making processes during the game. Common decision rules are to combine containers with the same destination onto the barge, postpone visiting a certain destination when containers with similar destinations are known to arrive the next day, and utilize the capacity of the barge as well as possible.

Planning Rules. The game includes a graphical algorithm creator. With this tool, the players can define their own decision rules. These rules typically have a structure similar to the manual strategies used by the players. To create an algorithm, players first define filters. A filter is a collection of container types, e.g., all containers having a time window of 1. The filters are strongly connected to the features considered in our RL methods. There is no restriction on the number of filters that can be created. After defining filters, the decision rules can be created. Each rule consists of an action and a condition. The action is applied to a given filter, and consists of transport by a certain modality or withholding transport. The action will only be performed when the conditions are met. There are three types of conditions. First, the condition that the selected transport modality in the action can only be used for containers whose destination is already assigned to the transport modality. Second, conditions related to certain thresholds that compare ($\leq, <, =, >, \geq$) the number of containers in some filter with (i) some number, (ii) the capacity of a modality, or (iii) the number of containers in another filter. Third, conditions related to costs, where we compare the costs of the proposed action with the costs of another action (combination of a filter and transport mode). The player can use as many conditions as needed for each action, and all conditions can be combined with *and/or* operators. Also for the number of decision rules, there is no limit. Players may change the sequence of decision rules; each rule will act only on the containers that are not yet assigned by one of the previous rules. After creating the algorithm, the players can apply them in the game. Depending on the game settings, the player might be allowed to (i) overrule the decisions taken by their algorithm, i.e., manually change them or (ii) adjust their algorithm during the game.

Decision Support. For any decision and its corresponding features, the learned VFA weights can be used to compute the expected downstream costs. We use the VFA to provide decision support in the form of estimated marginal costs or savings. More specifically, we show the difference in estimated value $\bar{V}_t^n(S_t, x_t)$ of the decision x_t (certain assignment of containers to modalities) and the decision of doing nothing. These estimated marginal costs can be balanced against the direct costs of the decision x_t. As the calculation of $\bar{V}_t^n(S_t, x_t)$ only requires multiplying (fixed) weights and features for a single manual decision at a time, this form of decision support can easily be provided in real-time environments.

5 Serious Game: Trucks and Barges

This section introduces the serious game that is used to compare the human planning performance with those of the algorithms from Sect. 4. The game is called Trucks & Barges and is publicly available at www.trucksandbarges.nl.

The purpose of the game is to let players gain experience with transport planning under uncertainty, raise awareness about some of the trade-offs in anticipatory planning, and familiarize them with decision support and automated planning. The game is designed to provide the player with insight into (i) a typical intermodal planning problem, (ii) the benefits and challenges in anticipatory planning, (iii) the benefits of decision support and automated planning rules, (iv) the complexity of the planning problem, and (v) the formalization of automated planning rules. On the other hand, the game also provides the game master with insights related to the behavior of the participants, their awareness about the trade-offs in anticipatory planning, their learning process, and the way they respond to various forms of decision support, automated planning rules and optimization algorithms.

The game can be played individually as well as within a serious gaming session under the guidance of a game master (e.g., a classroom setting). When playing the game within a serious gaming session, the game master first selects a scenario that describes all problem settings, such as the modalities, costs, container arrival probabilities, destination probabilities, etc. Using these settings, the game master can represent different logistics challenges. As such, human performance can be evaluated from various perspectives. Next, the game master defines the rounds to be played. Each round consists of a pre-defined number of weeks where a player is in the same mode: practice, normal, support, planning, or automated. In the *practice* and *normal* round, the player only sees the direct costs of each selected decision. The normal round corresponds with "manual planning" in Sect. 4. The practice round has a build-in tutorial where players receive information on all elements within the game. Furthermore, the performance in the practice round does not affect the player's game score. The *support* round corresponds with "decision support" in Sect. 4. Here, the learned VFA weights are used to estimate the downstream costs of decisions. For each manual decision, the player sees both the direct costs and the estimated marginal downstream costs on top of the playing screen (see Fig. 2). The sum of these two costs supports the player in evaluating the marginal future impact of decisions with different immediate costs, i.e., to find the decision that is expected to be beneficial in the long run. The *planning* round corresponds with "planning rules" from Sect. 4. Here, the player uses the algorithm creator to make decisions according to his/her own fixed rules (but depending on the game settings, the player can either overrule these decisions or change the planning rules during the game). Finally, in the *automated* round, the player can see all the algorithms from Sect. 4 into action, without the option to manually intervene.

Fig. 2. Screenshot of Trucks & Barges serious game. Containers to be transported can be dragged to truck, barge, or train. Container colors indicate destination, container numbers reflect the remaining time window. (Color figure online)

The player takes the role of an LSP planner who schedules the transport of containers from the hinterland to a deep-sea port using trucks, barges, and trains. The underlying problem is as described in Sect. 3. The main playing screen is shown in Fig. 2. Containers are colored according to their destination (red, green, and blue for the three destinations considered in the game) and are located in one of two container yards. The container yard to the left holds containers that are released for transport ($r = 0$) and the yard on the right holds container that are to be released the next day ($r = 1$). Furthermore, containers are labeled with a white number in the middle according to their current time window, which decreases as days pass and containers are not transported. In the game, the possible time windows are $k \in \{0, 1, 2\}$ (incremented in-game by 1 to reflect travel times of 1 day) after the release day r. To schedule the transport of containers, the player can drag-and-drop containers from the left container yard onto a truck, barge, or train. The player can also undo such movements, dragging containers back from the modalities onto the container yard. The daily barge and train, as well as the trucks, take one day to bring a container to its destination, meaning that a container with a time window of 1 must be transported today. All containers must be transported within their time window; the unlimited number of trucks guarantees this is always feasible. The daily plan is finalized and executed when the end button is pressed or when the maximum time for a day's decision has elapsed, which is indicated by a clock in the bottom left corner.

Dispatch decisions result in "immediate costs", which are displayed next to the used modalities. At the end of each turn, containers that had a time window of 1 and were not transported will be automatically assigned to the trucks. Then, the barge, train, and trucks depart – visualized by an animation – and a daily report is presented to the player with the costs of his or her decisions. In the transition to the next turn, two things happen: (i) containers to be released the next day (i.e., containers in the right-hand yard) are moved to the left-hand yard, and (ii) new containers arrive to the two container yards. The turns continue until the end of the week, where the game "cleans" the containers that were left and assigns cleaning costs to the player. At the end of a week, the player gets a report on his or her costs for each day and the cleanup costs. At the end of a round, a round report is displayed providing insights into the player's performance (or algorithmic performance). The game also has a dynamic leaderboard, where players can see their ranking and the performance of other players in the game session in real-time.

The initial state at the beginning of each week, as well as the daily order arrivals are generated up front, such that all players face exactly the same game instance. However, even though this information is generated up front, none of the policies from the previous section will utilize information regarding future events, e.g., the RL policies only use the probabilities used to generate the container arrivals. All costs components and container arrival probabilities are accessible by the player through the info button.

6 Experimental Results

To illustrate the possible usage of the game, we now report on three gaming sessions with a different audience: students, job seekers, and logistics professionals. Depending on the audience, we consider different game scenarios and round types. For all game scenarios, we exclude the train as transport modality to not overly complicate the human decision-making process. We distinguish between two game scenarios: easy and difficult. In the easy scenario, the barge is always cheaper per container compared to truck, even though there are still setup costs for using the barge, depending on the combinations of destinations that need to be visited. In the difficult scenario, the barge is only cheaper compared to truck when sufficiently being utilized. Each rounds consist of 3 weeks, and each week consists of 5 days/turns. For each turn, we use a time limit of 60 s, which seems to be more than sufficient given that the average decision time of the players in all of the three gaming sessions was 17.8 s, and 85.4% of the decisions were made within 30 s. Note that the times for our heuristics and RL algorithms to make decisions are negligible ($\ll$1 s). Here the RL algorithms benefit from the fact that the VFA is learned offline once per game scenario, which typically requires a couple of seconds.

The settings of the game scenarios can be found in the publicly available 'rooms' denoted by "low/high barge costs 4 rounds" at www.trucksandbarges.nl. Note that the rounds themselves are different in these publicly available games compared to the rounds of the serious gaming session as described in Sect. 4.

Two psychological effects are important with respect to the performance of the players: the anchoring effect and the reference effect [6]. Within our gaming sessions, there is a heavy reliance of players on the information initially offered. Also, if the player already played a round using decision support, this typically affects his/her behavior in subsequent rounds. Similarly, the costs made in the first round will serve as a reference for the performance in later rounds. To filter our these effects as much as possible, all players in all sessions first had to complete 3 weeks in the practice round.

For all groups, we studied various aspects, such as the relation between performance and decision time, and the relation between performance and the number of times the participants consulted the screens with cost information. However, given the limited number of participants and the huge fluctuations in performance, none of these relations were significant ($\alpha = 0.05$). Therefore, we only show the results of the performance of the players and algorithms in terms of costs. These results are all summarized in graphs showing the cumulative distribution function of the human players, scaled relative to the performance of our heuristics and RL policies. With respect to the cumulative distribution function, we first rank the players from good (low costs) to bad (high costs), and plot their scaled performance against the fraction of players that scored as least as well. With respect to the scaled performance, we compare the human performance against the four algorithms from Sect. 4: Heuristic, Myopic (direct cost minimization), RL (reinforcement learning), and RLext (reinforcement learning with extended VFA). We index the performance of the best performing algorithm to 0 and the performance of the worst performing algorithm to 1. In all experiments, RLext resulted in the best performance. The worst algorithmic performance was achieved by either Heuristic or Myopic. A player having a score higher than 1 means that all four algorithms performed better than this player.

6.1 Gaming Session 1: With Students

We first performed an extensive gaming session with 40 students from the master program Industrial Engineering and Management at the University of Twente. All students had to play 3 rounds: *normal*, *support*, and *planning*. We divided the group of students into two: one group first playing a normal round followed by a support round and the other group playing the rounds in reverse order. All groups ended with the planning round. Furthermore, all students had to play both the easy and difficult gaming scenario.

We first assess the difference in performance due to changing the sequence of the normal and support round. Players that first played support had on average 3.7% lower costs in the normal round. This suggests that lessons from the support round might be transferred to the normal round. However, given the huge fluctuations among the limited number of players, this difference is not significant. As only minor differences were observed between normal and support rounds, these results have been aggregated in Fig. 3 under the term *manual* (hence the graph corresponding with manual has twice as many observations as the graph corresponding with rules).

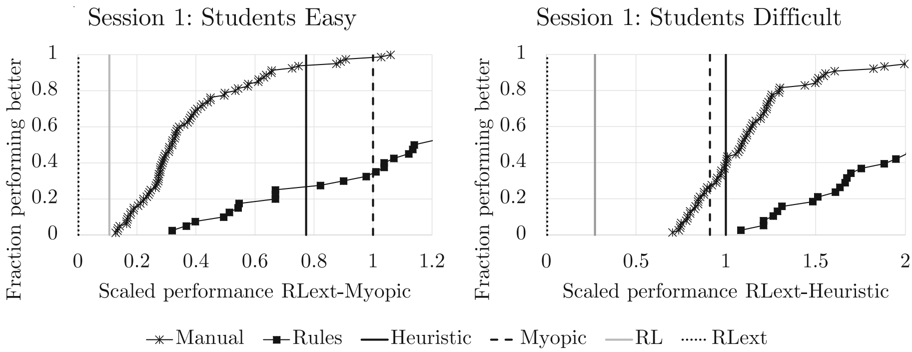

Fig. 3. Cumulative distribution in Session 1 of the scaled human performance relative to RLext (=0) and the worst-performing heuristic (=1). Both RLext and RL consistently outperform human players. In the difficult scenario, players' results considerably decrease.

Figure 3 visualizes the game results. When looking at the left-hand graph (representing the easy scenario), we observe that 98% and 94% of the students outperformed Myopic and Heuristic, respectively. With the algorithm creator, only 33% of the students outperformed the myopic solution method. For the difficult scenario (right-hand graph), we observe that 38% and 26% of the students outperformed the Heuristic and Myopic, respectively. In both scenarios, students had a hard time creating planning rules that improved their manual game play or one of the heuristics.

6.2 Gaming Session 2: With Job Seekers

The second gaming session was performed at a logistics fair organized in the Netherlands. This fair was organized for everyone looking for a job in the logistics sector, specifically aimed at students in pre-vocational secondary schools. At this fair, visitors could, e.g., drive trucks and forklifts, control drones, design a warehouse, stack pallets, but also play our game. As time is limited in such a situation, we let players first practice, then play only one round to measure their performance, and end up with an automated round in which they could see our algorithms in action. The results of 60 players are shown on the left-hand side of Fig. 4. Average performance is worse than for master students, possibly due to a lack of domain knowledge inherent to young job seekers. Best- and worst-case performance are comparable to that of students though.

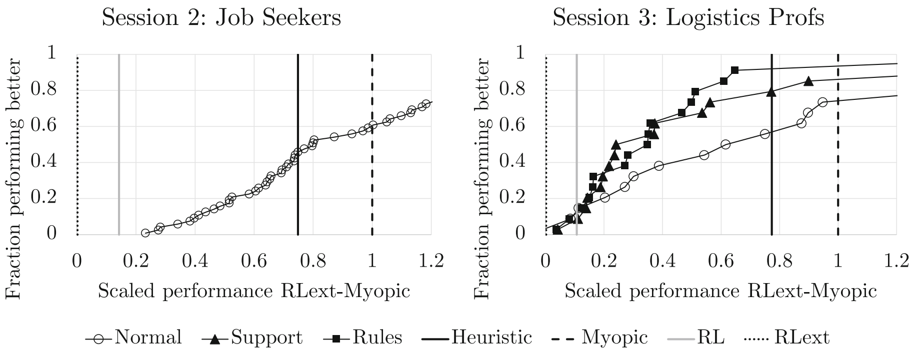

Fig. 4. Cumulative distribution in Sessions 2+3 of the scaled human performance relative to RLext (=0) and the myopic algorithm (=1). Note that some logistics professionals outperform the RL algorithms.

6.3 Gaming Session 3: With Logistics Professionals

Finally, we performed a gaming session at one of the larger logistics companies within The Netherlands. With 17 logistics professionals, we played the same rounds as with the students, but only considering the easy scenario and using the following sequence of rounds for all players: normal, support, and planning. The results are shown on the right-hand side of Fig. 4. In contrast with the results of the students, we now show the results of the two manual rounds (normal and support) separately, as there is a clear increase in performance when using support (but differences are still not significant given the limited number of players). Furthermore, a few players actually outperformed the RL algorithm. Finally, we observe that the logistics professionals were able to create planning rules, using the algorithm creator, that improved their manual game play.

6.4 Interpretation of Algorithmic Performance

We conclude the assessment of our results by briefly reflecting on algorithmic performance. For the easy scenario, the performance of the myopic policy is poor, being outperformed by most human players. This result highlights the challenges of anticipating the future impact of current dispatching decisions. For the difficult scenario, myopic performance drastically improves though. In this scenario, rules-of-thumb no longer work well; evaluating many solutions pays off here. The heuristic performs relatively well for the simple scenario, yet most humans are able to obtain better results. The heuristic embeds some rudimentary logic, but cannot handle exceptions and information about the future. These flaws are aggravated in the difficult scenario. Finally, RL outperforms virtually all players even in the easy scenario; only a handful of logistics professionals can beat its score. It is our only algorithm that explicitly considers downstream costs, which is crucial in anticipatory planning.

7 Conclusions and Further Research

This paper describes an anticipatory planning problem for the dispatching of trucks, barges, and trains, considering uncertainty in future container arrivals. We design several algorithms – a myopic policy, a heuristic and two reinforcement learning strategies – to solve the problem. Furthermore, we develop a serious game called Trucks & Barges to compare the performance of algorithms and human planners. Within the game, players can plan the containers manually both with and without decision support. In addition, the players may create their own planning rules using a graphical algorithm creator.

Test results are obtained through three serious gaming sessions: with students, with job seekers in the logistics sector, and with logistics professionals. The experimental results show that sophisticated heuristics and reinforcement learning on average outperform the human decision makers, but that the top tier of humans comes very close to the algorithmic performance. Both the creation of heuristic rules and of value function approximations in reinforcement learning require considerable domain knowledge. The results of the best logistics professionals in particular highlight the importance of domain knowledge in algorithmic design. Nevertheless, well-designed algorithms outperform most human decision makers. These insights imply that both human expertise and algorithmic developments remain necessary to advance the art of anticipatory planning in the logistics sector.

References

1. Anderson, D., et al.: Human-guided simple search. In: AAAI/IAAI, pp. 209–216 (2000)
2. Bertsekas, D.P.: Dynamic Programming and Optimal Control, vol. 1, 2nd edn. Athena Scientific, Belmont (2017)
3. Buiel, E., et al.: Synchro mania - design and evaluation of a serious game creating a mind shift in transport planning. In: 46th International Simulation and Gaming Association Conference, ISAGA, pp. 1–12 (2015)
4. Drakoulis, R., et al.: A gamified flexible transportation service for on-demand public transport. IEEE Trans. Intell. Transp. Syst. **19**(3), 921–933 (2018)
5. Gigerenzer, G., Todd, P.M.: Fast and frugal heuristics: the adaptive toolbox. In: Simple Heuristics that make us Smart, pp. 3–34. Oxford University Press, Oxford (1999)
6. Kahneman, D.: Thinking, Fast and Slow. Macmillan, New York City (2011)
7. Kefalidou, G., Ormerod, T.C.: The fast and the not-so-frugal: human heuristics for optimization problem solving. In: Proceedings of the Annual Meeting of the Cognitive Science Society, vol. 36 (2014)
8. Keller, N., Katsikopoulos, K.V.: On the role of psychological heuristics in operational research; and a demonstration in military stability operations. Eur. J. Oper. Res. **249**(3), 1063–1073 (2016)
9. Klapp, M.A., Erera, A.L., Toriello, A.: The dynamic dispatch waves problem for same-day delivery. Eur. J. Oper. Res. **271**(2), 519–534 (2018)

10. Kurapati, S., Kourounioti, I., Lukosch, H., Tavasszy, L., Verbraeck, A.: Fostering sustainable transportation operations through corridor management: a simulation gaming approach. Sustainability **10**(2), 1–18 (2018)
11. Meijer, S.A., Mayer, I.S., van Luipen, J., Weitenberg, N.: Gaming rail cargo management: exploring and validating alternative modes of organization. Simul. Gaming **43**(1), 85–101 (2012)
12. Minkoff, A.S.: A Markov decision model and decomposition heuristic for dynamic vehicle dispatching. Oper. Res. **41**(1), 77–90 (1993)
13. Rivera, A.P., Mes, M.: Dynamic multi-period freight consolidation. In: Corman, F., Voß, S., Negenborn, R.R. (eds.) ICCL 2015. LNCS, vol. 9335, pp. 370–385. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24264-4_26
14. Pérez Rivera, A.E., Mes, M.R.K.: Anticipatory freight selection in intermodal long-haul round-trips. Transp. Res. Part E: Logist. Transp. Rev. **105**, 176–194 (2017)
15. Powell, W.B.: Approximate Dynamic Programming: Solving the Curses of Dimensionality. Wiley Series in Probability and Statistics, 2nd edn. Wiley, Hoboken (2011)
16. Raghothama, J., Meijer, S.A.: A review of gaming simulation in transportation. In: Meijer, S.A., Smeds, R. (eds.) ISAGA 2013. LNCS, vol. 8264, pp. 237–244. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-04954-0_28
17. Rossetti, R.J.F., Almeida, J.A.E., Kokkinogenis, Z., Gonçalves, J.: Playing transportation seriously: applications of serious games to artificial transportation systems. IEEE Intell. Syst. **28**(4), 107–112 (2013). https://doi.org/10.1109/MIS.2013.113
18. Samek, W., Montavon, G., Vedaldi, A., Hansen, L.K., Müller, K.-R. (eds.): Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. LNCS (LNAI), vol. 11700. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28954-6
19. Simon, H.A.: Bounded rationality. In: Eatwell, J., Milgate, M., Newman, P. (eds.) Utility and Probability. The New Palgrave, pp. 15–18. Palgrave Macmillan, London (1990). https://doi.org/10.1007/978-1-349-20568-4_5
20. Sterman, J.D.: Modeling managerial behavior: misperceptions of feedback in a dynamic decision making experiment. Manag. Sci. **35**(3), 321–339 (1989)
21. Van Heeswijk, W.J.A., Mes, M.R.K., Schutten, J.M.J.: The delivery dispatching problem with time windows for urban consolidation centers. Transp. Sci. **53**(1), 203–221 (2019)
22. Venkatesh, V.: Creation of favorable user perceptions: exploring the role of intrinsic motivation. MIS Q. **23**(2), 239–260 (1999)
23. Venkatesh, V., Bala, H.: Technology acceptance model 3 and a research agenda on interventions. Decis. Sci. **39**(2), 273–315 (2008)
24. Voccia, S.A., Campbell, A.M., Thomas, B.W.: The same-day delivery problem for online purchases. Transp. Sci. **53**(1), 167–184 (2017)
25. Wiener, J.M., Ehbauer, N.N., Mallot, H.A.: Path planning and optimization in the traveling salesman problem: nearest neighbor vs. region-based strategies. In: Dagstuhl Seminar Proceedings, pp. 1–21 (2007)