Spatio-temporal Analysis of Multi-agent Scheduling Behaviors on Fixed-track Networks

Shivam Agarwal*
University of Duisburg-Essen, Germany

Günter Wallner†
Johannes Kepler University Linz, Austria

Jeremy Watson‡
AICrowd, Switzerland

Fabian Beck§
University of Bamberg, Germany

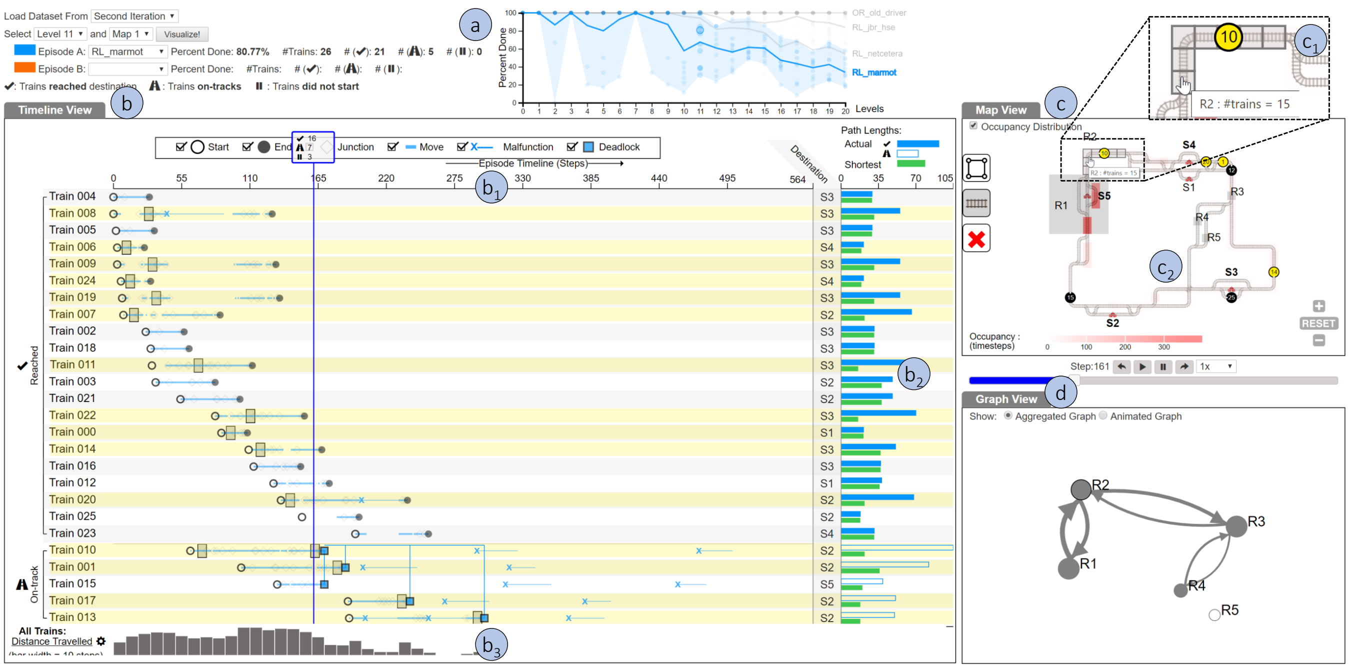

Figure 1: The proposed visualization approach showing trains scheduled using a reinforcement learning approach. The interface consists of: (a) an episode selection panel, (b) a timeline view, (c) a map view, and (d) a graph view. Hovering over a region in the map view highlights in the timeline view the trains that visited the region (yellow background for highlighting, gray rectangles show the visit duration).

Abstract

Multi-agent systems require coordination among the agents to solve a given task. For movement on fixed-track networks, traditional scheduling algorithms have dominated so far, but the interest in autonomous and intelligent agents is growing as they promise to react to unexpected and exceptional situations more robustly. In this paper, we study data from the *Flatland 2020 NeurIPS Competition*, where trains move through a virtual rail network. We developed a timeline-based visualization that provides an overview of all train movements in a simulated episode, clearly hinting at different phases, non-optimal routes, and issues such as deadlocks. This view is complemented with a map view and a graph view, interactively linked through highlighting and synchronous animation. Defining regions of interest in the map builds an analysis graph for detailed inspection. A comparison mode allows contrasting two different episodes regarding the same rail network across all views. We have conducted this application study in close collaboration with the *Flatland* community. Identified analysis goals stem from interviews with key persons of the community, while the approach itself was developed in two iterations based on feedback from experts with diverse backgrounds. This feedback, together with an analysis of the winning submissions from the competition, confirms that the initial analysis goals can be answered.

Keywords: Multi-agent systems, scheduling, visual comparison, spatio-temporal analysis.

1 Introduction

Multi-agent systems solve complex problems by having autonomous entities (agents) interact with each other [10, 17, 22]. According to a recent survey [10], city and build environments form a major application area, for instance, to simulate and anticipate future traffic behavior (e.g., [12, 23]) or to optimize scheduling (e.g., [6, 9]). In such use cases, coordination between agents is of utmost importance to avoid collisions and to optimize route selection. Visualizations that enable the spatio-temporal analysis of multi-agent scheduling and movement can help to understand the agents' actions, their interdependence, and the coordination between them, in turn, helping to optimize agent behavior.

In our work, we focus on visually analyzing movement and scheduling behavior in an environment where the agents are constrained by a fixed-track network. While we study a simulation, in reality, the fixed tracks can be a physical delimiter such as a rail, or a virtual track that the agents must not leave (e.g., robots moving

*e-mail: shivam.agarwal@paluno.uni-due.de
†e-mail: guenter.wallner@jku.at
‡e-mail: jeremy@aicrowd.com
§e-mail: fabian.beck@uni-bamberg.de

in a warehouse). More specifically, we base our approach on the *Flatland* environment [15], a testbed for developing agents that act as trains in rail networks. The goal is to schedule trains to reach their target destinations within minimum travel time. We designed our approach to meet analysis goals elicited in interviews with three domain experts. We developed it in two iterations, where the first version was used to gather feedback from experts with different backgrounds. We then refined the approach and finally used it to analyze data from the *Flatland 2020 NeurIPS Competition*.

Figure 1 shows the developed visualization approach on a selected episode (i.e., one run of the *Flatland* simulation on one fixed-track network using a specific scheduling algorithm). A train's journey is represented in a row of the timeline view (Figure 1b). Important events such as start and end times, junctions crossed by a train, movements, and different issues such as deadlocks are also encoded in the respective rows. The map view (Figure 1c) shows the rail network of the selected episode and allows defining regions of interest. Each region becomes a node and movements between the regions are depicted as links in the node-link graph (Figure 1d). Based on the expert feedback, we extended our approach to visually compare two selected episodes run on the same fixed-track network (Figure 3).

In summary, our contribution is an application study proposing a visual analytics approach with linked views and interactions to support (i) an in-depth exploration of agent coordination behavior in a selected episode and (ii) visual comparison across two selected episodes. We developed the approach in close collaboration with the *Flatland* community and other experts, as well as tested the approach through a realistic application example. The supplementary material [2] includes the prototype of the approach[1], a video, supplementary figures and tables, the video and questionnaire used to collect expert feedback, and their respective responses.

2 Related Work

As indicated by state-of-the-art survey papers [3, 20], many visualizations have been proposed to analyze mobility and transportation. While real-world traffic (e.g., cars on road network) shares similar properties to our scenario, the analysis goals are usually different, for instance, to study traffic congestions [7] or explore traffic attributes [11, 25]. Insights regarding these goals help to modify the network for efficient traffic flow. In contrast, we aim to understand the movement behavior of agents in fixed-track networks to improve their scheduling. Although we are not aware of any visual analytics approach for studying movement in such networks, we find related approaches for visualizing other multi-agent scenarios, visualizations of transit schedules, and visually abstracted movement.

To trace the movement of individual agents, timeline visualizations are common. For instance, Cummings and Mitchell [8] propose a glyph-based timeline visualization for human operators to monitor the live status of multiple unmanned vehicles (robots), where the movement schedule for each vehicle was already planned. Generalizing further, *VizScript* [13] generates timeline visualizations for complex multi-agent systems. However, unlike our approach, it does not convey the effect of important events on other agents, e.g., deadlocks. Most similar to our approach is *MOSAIC Viewer* [5], which proposes a timeline visualization to show the activities of each agent in an individual row, together with a graph representing each agent as a node and communication links between them as edges. However, *MOSAIC Viewer* focuses on investigating the agents' activities when they are not synchronized in their worldviews.

To visually investigate collective movement behavior, heatmaps can be used. For instance, Suarez et al. [21] presented a research platform for simulating large-scale multi-agent combat and navigation policies. The platform comes bundled with several heatmap-based visualizations to help understand the learned policies. In our approach, we also use a heatmap to show the collective movement behavior of a selected scheduling technique as an overview, but allow further exploration with abstraction and linked interactions. We also extend the heatmap to show the differences in movement behavior between two scheduling techniques.

[1] also hosted at: `https://s-agarwl.github.io/fv`

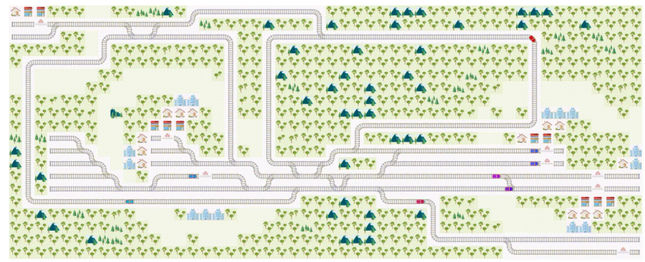

Figure 2: A sample map of the *Flatland* environment.

The Marey graphs, dating back to the 19th century, is a classical example that shows transit schedules by plotting distance against time (cf. [14]). More recently, Palomo et al. [19] proposed a visual analytics system to assist in analyzing transport schedules and detect potential deficiencies, which extends Marey graphs for showing large numbers of trips. Related to this, Xu et al. [28] extended Marey graphs for analyzing manufacturing processes, specifically assembly line performance, also making adaptions for visualizing large-scale data. However, Marey graphs are not applicable in our case: They are appropriate for visualizing the movement with respect to a single line (i.e., one route in the network), but cannot visualize the movement of several trains in a complex network at once.

To reduce visual clutter when showing all movements between places or regions, some visualizations have been proposed to represent origin–destination flows in a more abstract way. For instance, Zeng et al. [29] used an adapted Sankey diagram to represent the flow. Similarly, Andrienko et al. [4] proposed a spatio-temporal abstraction of the origin–destination movement data and a diagram map representation including a glyph-based design of nodes without showing any links. In our approach, we also abstract the movement between regions and represent it as a node-link graph. The abstracted graph representation can also be considered as a form of mobility graph as introduced by von Landesberger et al. [27], that is, a node-link diagram that shows the movement between semantic places. While mobility graphs focus on representing aggregated movement behavior, our aim is to enable a detailed analysis of individual agents.

3 The *Flatland* Environment

Flatland [15] is an open source project and simulates the train scheduling on different maps of rail networks. Each map is a 2D grid consisting of railway tracks on which the trains can travel, and stations as destinations for the trains (Figure 2). Different levels exist that vary in the grid size, the number of stations, and trains to be scheduled. Within each level, different maps are produced that differ in the rail network layout and rate of malfunctions among trains. In a map, each train has a starting position on a track and a station as destination. The trains do not have intermediate stops on their journey. Trains travel at a constant speed of one tile per timestep and cannot move backwards. No two trains can be present on the same tile of the grid at the same timestep. The challenge is to schedule and steer the trains so that they reach their destination in minimum time. An episode in *Flatland* is one run of a scheduling technique on a map. Each episode has a maximum time limit, which scales with the size of the map and after which the episode ends, even if some trains are still on track. Also, trains can randomly experience malfunctions during the episode, which restricts their movement for some time.

In machine learning terminology, each train is modeled as an agent. Hence, it becomes a multi-agent scheduling problem, where agents need to collaborate and come up with an optimized schedule while handling malfunctions at runtime. The environment supports customization of agent observations (i.e., what each train sees in the rail network), which is crucial for agents to make decisions during runtime. To advance machine learning research, two *Flatland* competitions have been organized (in 2019 and 2020 at the AMLD and NeurIPS conferences).

While developing a scheduling technique for *Flatland*, the experts need to analyze the paths followed by the trains and discover issues (e.g., situations leading to a deadlock) to fine-tune the technique and improve its performance. Usually, the experts rely on watching episode playbacks, which is inefficient or can be inconclusive due to several key challenges. First, in *Flatland*, the effect of errors or suboptimal agent decisions are only often visible much later in the episode. Remembering and linking the decisions of each agent to their effect becomes difficult in the playback. Second, understanding agent coordination behavior in *Flatland* is crucial for further improvement of the scheduling technique, which, however, can hardly be monitored in the playback. Currently, the experts rely on performance statistics (e.g., percentage of trains that reached their destination) as a proxy for agent coordination behavior. However, it is insufficient, as they do not help the experts to understand and improve the qualitative aspects of coordination behavior on global and local areas of the network. Third, the experts need to identify unusual agent behavior within the context of the network (e.g., formation of a long queue due to an agent waiting on a single track connecting distant stations). While individual issues like these might surface when watching the playback, it requires stepping back and forth to recontruct their history, whereas other related issues might stay unnoticed. Fourth, a detailed comparison of alterante scheduling approaches, even on the same map, is impractical as two videos are difficult to watch at the same time and, watching them one after the other, requires high cognitive effort to remember relevant details for comparison.

4 Analysis Goals

Before designing our approach, we conducted interviews with *Flatland* experts to understand their information needs regarding the visualization of single episodes. We interviewed three experts (*E1* to *E3*), with each interview taking about 30 minutes. *E1* had a main role in organizing the *Flatland* competitions and provides technical support for participants. *E2* was affiliated to the Swiss Federal Railways (SBB – Schweizerische Bundesbahnen, a *Flatland* partner) and analyzed the scheduling behavior of trains in *Flatland*, as well as substantially contributed to the code base of the environment. *E3* is an artificial intelligence professional affiliated with the German railway company Deutsche Bahn (DB), also a *Flatland* partner, and explored agent-based scheduling techniques in *Flatland*. Based on the interviews, we derived five specific analysis goals (G1 to G5).

First, all experts wanted to obtain an overview of the schedules of individual trains in the episode (*E1*, *E2*, and *E3*). Expert *E1* highlighted the need to analyze junctions crossed by a train (as they are crucial locations for making a decision), occurrence of malfunctions, and deadlock events. *E1* and *E3* also stressed the importance of statistics (e.g., number of trains that reached the destination). The need to focus on trains that did not reach their destination was mentioned by *E2*.

> **G1 – Overview of Schedules:** Get an overview of events and actions (e.g., departure time, junctions crossed, movement, etc.) for each agent. Also, inspect important statistics for an individual episode.

E2 highlighted the need to analyze the use of resources in the rail network, for example, finding out the busiest routes in the network. *E3* used specific environments to judge the scheduling behavior through resource utilization of a scheduling technique where the best-case scenario is known beforehand. The expert mentioned relying on playback, focusing on a lower number of trains, and observing the behavior by following individual trains.

> **G2 – Resource Utilization:** Analyze the utilization of resources in the network: tracks connecting distant places, critical junctions, and areas with a high number of stations or unusual agent behavior.

E2 also mentioned the need to assess the efficiency of train schedules, for instance, how delayed trains were in reaching their destination. Since this assessment needs some reference, the expert usually compared the actual path of the train with the shortest path, assuming there is no other train in the rail network. *E3* also mentioned that the shortest path plays an important role in deciding rewards, which is crucial for reinforcement-learning-based scheduling approaches.

> **G3 – Path Efficiency:** Assess the efficiency of the actual paths taken by agents.

Understanding the cause of issues (e.g., deadlocks, malfunctions, and bottlenecks) is important to improve a scheduling technique. *E2* highlighted the need to understand what has happened in the immediate past to investigate the reasons leading to a deadlock. Adding further, *E2* and *E3* mentioned that it is important to see how other trains reacted to these issues and to be able to observe which areas in the rail network were affected.

> **G4 – Issues:** Investigate the cause and effect of issues, e.g., deadlocks, malfunctions, and bottlenecks.

Finally, *E1* and *E2* highlighted the need to explore scheduling strategies exhibited by the collective and simultaneous movement behavior of a group of trains globally and in local areas. Since *Flatland* promotes experimentation with different scheduling techniques (e.g., reinforcement learning, operations research, hybrid approaches), the exploration of scheduling strategies should be model-agnostic. Such exploration is required to understand whether trains are collaborating by reacting to the actions taken by other trains, or not. The experts gave two examples: 1) using parallel tracks for one-way traffic and 2) trains following each other with minimal gaps.

> **G5 – Scheduling Strategies:** Explore the scheduling strategies through collective and simultaneous movement behavior of a group of agents.

An additional requirement resulted from later expert feedback (cf. Section 6) and was taken into consideration for a revised version of the approach. Participants suggested a comparison between two episodes (different scheduling methods or variants of the same method) on the same network. Such comparison would help developers experiment with new ideas and understand the differences between their approach and past top-performing solutions. It would also help organizers of the competition to explore and report qualitative differences in the scheduling behavior of different submissions.

> **G6 – Comparing Schedules:** Compare agent schedules, resource utilization, efficiency, and strategies between two different episodes.

To better understand the interests of the *Flatland* community and observe the setup of the *Flatland 2020 NeurIPS Competition*, the first author of the paper—having a background in visualization research—regularly attended the weekly community meetings held online for 3 months. Likewise, a person from the *Flatland* community collaborated with us, attended our meetings, and co-authored the paper.

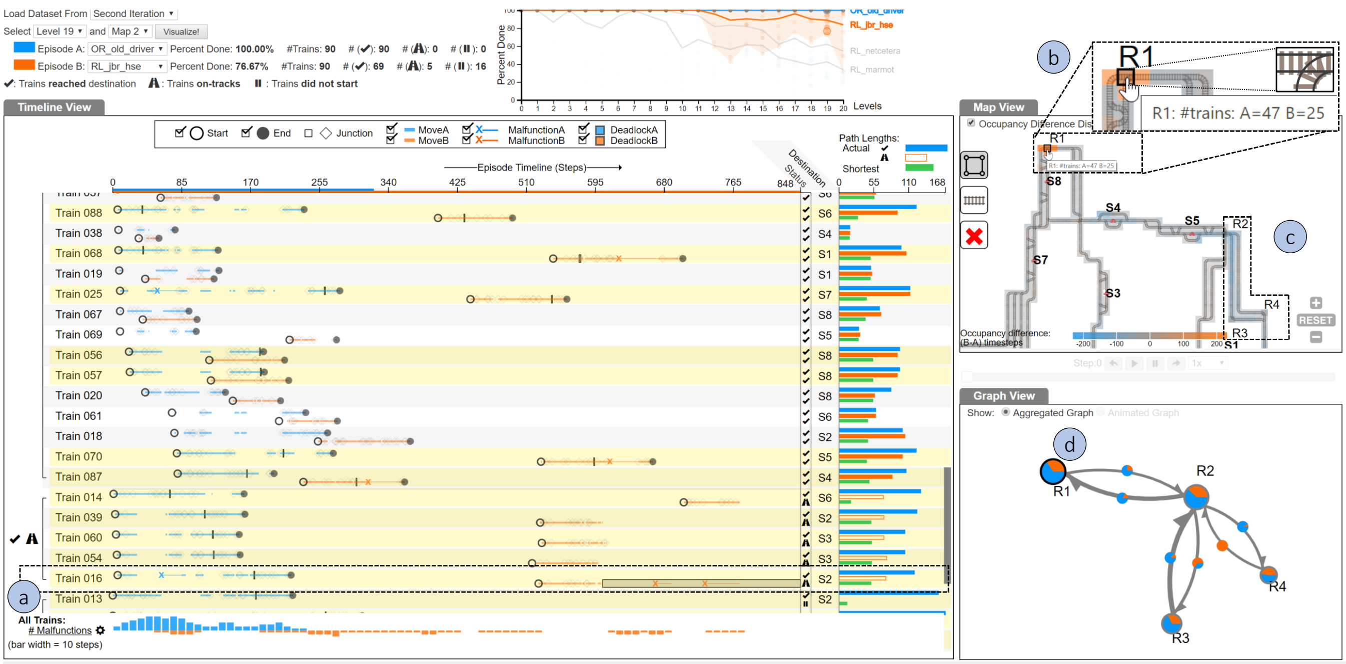

Figure 3: Screenshot of the proposed approach comparing the train schedules produced by an operations research technique by team `old_driver` and a reinforcement learning technique by team `jbr_hse` on a selected *Flatland* episode (`Level 19, Map 2`).

5 Our Visualization Approach

Based on the analysis goals, we propose a visualization approach to help analyze the train movement behavior in *Flatland* episodes. The approach consists of three linked views providing different perspectives on the train movement data. Figure 1 and Figure 3 shows the full interface of the proposed approach.

The approach was developed in two iterations. We implemented the first version during the *Flatland 2020 NeurIPS Competition* and collected feedback from a diverse group of experts. Based on the results, the approach was further extended in the second iteration after the competition ended.

5.1 The Episode Selection Panel

On top of the interface (Figure 1a), a line chart shows the percentage of trains that reached their destination along various test levels for each scheduling technique. Since there are several maps in each level, the line chart shows circles to mark performances on each map for all levels. A level, map, and a scheduling technique for an episode can be selected by clicking on the circles in the line chart or via drop down lists on the left. Once selected, we show statistics about the number of trains in the episode that needed to be scheduled, the percentage of trains that reached their destination, and the number of trains based on their end status (✔ reached destination, on-track, and did not start) (**G1**). For comparing schedules from two episodes on the same map (**G6**), we use two colors—blue (■) and orange (■)—consistently across the interface to identify the unique characteristics of the selected scheduling methods in episode A and episode B, respectively (Figure 3).

5.2 The Timeline View

To provide an overview of an individual episode (**G1**) and compare two episodes (**G6**), we integrate a timeline view (Figure 1b). Positioned at the left, the timeline view visualizes the trains in different rows, with time progressing from left to right (Figure 1b$_1$). The developers usually analyze the trains based on their end status, for instance, assessing path efficiency of the group of trains that reached their destination (✔) vs. exploring the reasons behind the group of trains that started and were still on-track ( ) until the end of the episode. Hence, in the timeline view, the trains are grouped according to their end status. Within each group, trains are ordered based on their starting time.

For each train, we show the important events that occurred during its journey (**G1**). The departure of a train is encoded as a black ring (○), while a filled gray circle (●) denotes arrival. Since trains mainly make route decisions at junctions, we also encode reaching a junction by a diamond shape (◇). Movement is depicted as a colored line (,). Malfunctions are represented by a colored cross with a tail (,), where the length of the tail denotes the duration of the malfunction (**G4**). Aiming to visualize issues in the movement of trains, we use the only direct agent interactions present in the data: head-on collisions. We detect simple cases of deadlocks where trains on a track are moving towards each other, without having their destinations on tiles of the track between them, and no alternate routes to pursue. We represent detected deadlock events (**G4**) as colored squares (■, ■) for individual trains at the position when the train was blocked and connect it to deadlock events of other trains involved in the same deadlock via colored horizontal and vertical lines (cf. last 5 rows in Figure 1b$_1$).

At the right of the timeline, the destination station of each train is shown in a column, together with statistics of the path taken (Figure 1b$_2$). Since the origin of trains in the *Flatland* environment is not a station but an unlabeled location on a track, we do not specify the train origin in the timeline view. Furthermore, to assess the efficiency of a train's route (**G3**), we calculate the length of the actual path taken and compare it with the shortest possible path between the origin and destination, assuming that all tracks are available for movement. The actual path length is shown by a horizontal bar in the color of the respective episode, while the bars for the shortest path length are colored in green (,). For trains that did not reach their destination at the end of the episode, the actual path length might be less than the shortest path length. To avoid confusion, we show the actual path length of trains on track by an unfilled bar (,).

To provide a temporal overview aggregated across all trains (**G5**), we include a histogram at the bottom of the timeline view (Figure 1b$_3$). The height of each bar in the histogram represents the summed value of a selected metric, aggregated across ten timesteps. In case of comparison (**G6**), colored bars extend in opposite vertical directions with a common horizontal axis ( ). Available metrics, which can be selected by clicking on the gear icon ( ), are (i) distance travelled (default selection), number of trains (ii) departing, (iii) arriving, (iv) experienced malfunctions, and (v) number of crossed junctions.

5.3 The Map View

Positioned at the top right of the interface (Figure 1c), the map view provides a spatial perspective of train movement in the rail network. Instead of using the original map representation, which contains a lot of *graphical sugar* (e.g., irrelevant elements such as trees, cf. Figure 2), we decided to design a more abstracted representation focusing on the rail tiles. The stations are labeled (e.g., *S1*), while each train is represented by a numbered circle, with a white dot showing its direction of movement. Since the size of the rail networks in *Flatland* episodes can be large, the map view supports zooming and panning interactions. Movement of trains on the rail network are shown using animations, which can be enabled through playback controls at the bottom of the view. To highlight the utilization of each rail tile (**G2**), we show on demand the occupancy time as a heatmap (Figure 1c) on a reddish sequential scale ( ). When comparing two episodes (**G6**), occupancy difference distribution is shown as a heatmap on a sequential scale from blue to orange ( ).

For analyzing utilization of specific resources in the network (**G2**) and details of train movements (**G5**), it becomes important to focus on specific regions. Since different regions can be of interest during the exploratory analysis, with different strategies of defining regions, we opted for a flexible selection of the regions-of-interest. We provide two modes of the selection in the map view, which can be activated by the two icon buttons at the left of the view. The first mode is a rectangular selection through mouse drag ( ). With this, also, a single tile can be selected as a region by left-clicking. The second mode selects a clicked rail segment between two junctions ( ). The selected regions are assigned a label and are highlighted by semi-transparent gray rectangles (Figure 1c). All selected regions can be cleared by clicking the respective button ( ).

5.4 The Graph View

To analyze movement of agents between selected regions (**G5**), we integrate a graph view in our approach. Positioned at the bottom right of the interface (Figure 1d), it contains a directed node-link diagram, where a node represents a selected region, while movement between regions is represented by the links between the nodes. The size of a node indicates the number of trains that have been in the corresponding region at least once. A train can move between two regions multiple times, and the width of a link represents the number of such transitions across all trains. Regions not traversed by any train are represented by unfilled circles. The node-link diagram is drawn using a force-directed layout, where the initial position of the nodes is set to the position of the selected region in the map view.

Two variants of the graph are selectable. First, the aggregated graph, as shown in Figure 1d, provides an overview across all timesteps highlighting the collective movement of agents in a static visualization. In the comparison mode (**G6**), as shown in Figure 3d, a pie chart is drawn inside each node of the aggregated graph to compare the number of trains in the selected region among the two episodes. Similarly, a pie chart for each link is added (positioned at the middle of each link) in order to compare the number of trains moving between the regions. Second, the animated graph, available in case only a single episode is selected for analysis, shows the trains moving between regions for the current timestep. For instance, an excerpt of an animated graph is shown in Figure 6a. The blue colored node *R3* indicates that there are trains in the corresponding region, while a small circle on the link from *R4* to *R3* indicates a train moving between the regions.

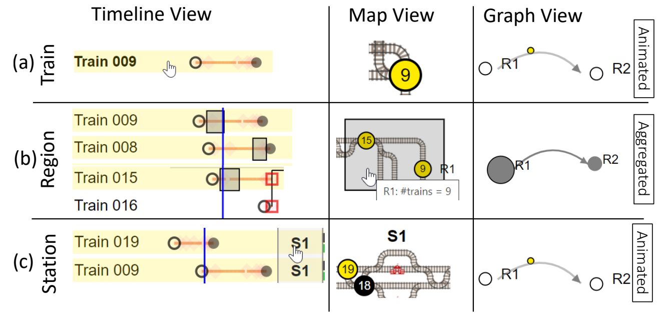

Figure 4: Linked interactions across the timeline, map, and graph view. Hovering highlights (a) an individual train (e.g., *Train 009*), (b) trains that visited the selected region (e.g., *R1*), and (c) trains with a common destination (e.g., *S1*) in all the three views.

5.5 Interactively Linked Views

With selection of a single episode, it can be played with the playback controls below the map view (**G3**, **G4**, and **G5**). This not only moves the trains on the map view and the graph view (if the animated graph is active), but also the dark blue time-slider on the timeline view. Furthermore, we integrate highlighting consistently across the views as illustrated in Figure 4. Hovering over a train in any of the views highlights the corresponding row and circles with a yellow background in all views. For example, in Figure 4a, the row of *Train 009* was hovered in the timeline view, hence, the train was also highlighted in the map and the animated graph. Accordingly, hovering over a region in the map or graph view highlights all the trains that have visited the region. For instance, in Figure 4b, hovering over the rectangular region *R1* reveals that nine trains visited the region. Additionally, a semi-transparent gray rectangle in the timeline shows the periods spent in the region by the corresponding train (**G2**). Hovering over a station label in the timeline or map view highlights the trains heading to this station (Figure 4c). A supplemental video [2] demonstrates these interactions.

5.6 Dataset

In the first version, we included the schedules generated by the two leading reinforcement learning and operations research approaches at the beginning of the competition. The schedules were on four different maps of the *Flatland* environment. Hence, we obtained data of eight episodes that can be selected by a drop-down list in the episode selection panel (Figure 1a). The name of an episode reflects the scheduling technique used: `RL` for reinforcement learning and `OR` for operations research. In the second version, we included the data from the top four winners of the competition across twenty test levels, each containing ten maps [24]. Hence, we added the data of $(4\times20\times10 =)$ 800 episodes.

6 Expert Feedback

We collected expert feedback on the first version of our approach in a questionnaire study providing the tool online. The first version did not contain the comparison features (which were added as a result of the study), the additional dataset from winning submissions, or the line chart showing summary statistics (Supplementary Figure 1 [2]). The survey was advertised on the *Flatland* forum, through social media and personal email, calling specifically for experts in artificial intelligence and visualization. For this study, the tool contained data of eight sample episodes from two scheduling techniques on four maps of the *Flatland* environment.

6.1 Questionnaire

At the start of the questionnaire, demographic information of participants was gathered such as their self-rated expertise in artificial intelligence, information visualization, and operations research (on a 5-point scale anchored by 1 = *no knowledge* and 5 = *expert*), and their background with the *Flatland* environment (cf. Figure 5b). Next, the participants were asked to freely explore and familiarize themselves with the visualization. An optional video explaining the available features and interactions was also provided to assist in getting accustomed with the tool. This was followed with questions regarding insights gained. Participants were invited to name and describe up to three insights they found most interesting. Specifically, we also asked them to elaborate on if they found any differences between reinforcement learning and operations research approaches. Then, for each of the three views, participants were asked to rate how useful and complementary to the other views it was on a 5-point scale anchored by 1 = *strongly disagree* and 5 = *strongly agree* as well as to provide feedback on what they liked and disliked about it. Next, the survey inquired about the system as a whole, asking to rate its helpfulness with respect to the analysis goals (cf. Section 4) using the same scale as above and to reflect what they generally (dis)liked about it. Lastly, we obtained general feedback for which tasks the visualization was deemed helpful and if any important information or features were missing or unnecessary.

6.2 Participants

In total, 12 participants took part. For analysis, we required participants to have good knowledge in at least one of the three fields (artificial intelligence, information visualization, or operations research). Hence, we include only responses from participants who rated themselves as 4 or higher on our expertise scale in at least one of the areas, resulting in 10 participants referred to as P1 to P10 in the following. As shown in Figure 5(a-b), seven participants (P1, P2, P3, P4, P6, P9, and P10) were experienced in artificial intelligence (AI), three (P3, P4, and P5) in operations research, and five (P6, P7, P8, P9, and P10) in information visualization. Six participants had some background with the *Flatland* environment, such as helping in organizing the *Flatland* competitions (P1, P2, P4, P5, and P6), participating in the competitions (P3, P5, and P6), or developing scheduling techniques in the *Flatland* environment (P1, P2, P3, P4, and P5).

6.3 Feedback Analysis Results

A summary of the feedback results can be found in Figure 5 (c-k). In the following, we report the results of feedback analysis along the structure of the questionnaire.

Insights Discovered: Four experts (P1, P2, P3, P6) reported observing the occurrence of deadlocks. Two of them (P1, P6) found it insightful to see the trains linked to a deadlock, and P2 remarked to have been able to investigate situations in which a deadlock occurred. P3 liked "*that the moment of a deadlock is shown at the time it becomes inevitable, rather than when the trains actually stop moving.*" P7 used the histograms at the bottom of the timeline view and observed that the occurrence of malfunctions was spread over an entire episode's length. Regarding the efficiency of paths, P1 reported comparing the lengths of the actual path and the shortest path of the trains. Furthermore, P7 observed that trains with a high number of junctions in their timeline have a much longer actual path length than the shortest path. P2 described to "*see the density of traffic over time.*"

Strategies for Defining Regions: The experts reported selecting rectangular regions (P2), single tiles of the grid (P7, P9), and individual rail segments in different combinations (P7) for further in-depth exploration. The experts also mentioned some of the important areas in the railway network that affect the scheduling. These include, individual stations (P7, P9), dense area with many junctions (P3) or nearby stations (P2), long single railway lines (P3), and single tracks at central positions (P1). An expert (P9) elaborated on using the heatmap to identify the most occupied parts of the network and selected them as regions.

Differences in Scheduling Behavior: Four experts (P6, P7, P8, P9) reported that, unlike the reinforcement learning solution, train schedules from the operations research had no deadlocks. Hence, the latter approach is able to schedule all the trains to their destination station (P6, P7, P8, P9). Highlighting a key difference between the two approaches, two experts (P3, P6) mentioned that the operations research approach shows a clear pre-planning of train paths until their destination. In addition, four experts (P1, P7, P8, P9) mentioned that the starting times in the train schedules of the reinforcement learning approach are spread across the entire episode length. P9 also reported that trains cross fewer junctions in the reinforcement learning approach.

Timeline View: Experts reported that they liked the timeline view as it provides an overview (P7, P10) and contains necessary information such as start/end events and duration of train movements (P8), deadlocks (P1, P6, P8), path lengths (P1, P4, P8), and density (P1), along with the distribution of the trains over time (P2). Two experts (P6, P9) liked the linking of the timeline view with the other views and P9 appreciated the sorting of trains based on their starting times. Experts disliked the inability to zoom in on a specific timespan (P1), reserved white space for trains that did not even start (P7), and the lack of a multi-selection feature for comparison (P8). P4 suggested extending the design to allow comparing schedules from two solutions in the same view, while P10 remarked that the view contains too many details.

Map View: Nine experts (all except P5) rated the statement that *the map view is useful for analysis and complements well the other views* as 4 or higher (Figure 5d). The experts liked the heatmap (P5), the ability to follow individual trains exactly (P2, P7), the capability to define regions of interest (P3, P6), and to see the hovered region on the timeline of trains (P6). P1 stated to like the view because it "*allows to investigate in more detail situations like deadlocks identified in the timeline view*", which was also mentioned by P8. P9 was fond of the abstraction of railways, while P10 appreciated the simplicity of the map view. With respect to shortcomings, two experts (P1, P8) thought that the view is too small to show big rail networks. P3 suggested including a more flexible shape for defining regions in addition to rectangles.

Graph View: Four experts agreed to (rating of 4 or 5), five experts were undecided (rating of 3), and one expert disagreed (rating of 2) with the statement that *the graph view (aggregated + animated) is useful for analysis and complements well the other views* (Figure 5e). Four of the neutral or negative replies (P2, P4, P5, P7) did not contain further details to explain the rating. However, P3 mentioned that it was unclear what was happening in the graph view, while P8 highlighted that deleting one region is currently not possible in the tool. On the other hand, three experts (P1, P2, P8) liked the abstract representation of train movement through graphs, a feature which P3 considered innovative. In addition, P7 mentioned that "*it was good to get the flow and amount of traffic between any two or more regions of tracks.*" Furthermore, the experts liked that the view can be used to investigate frequent routes (P6) or situations in which trains visit regions multiple times (P4). P8 and P9 appreciated the animation in the graph view. P9 reported the inability to select a time range for the aggregated graph.

Analysis Goals: Except for one analysis goal (**G3**), ratings of the others reflect that the experts tend to agree with the statements that the system helps to achieve the respective analysis goals, as shown in Figure 5(f-j). Investigating the responses to understand the relatively

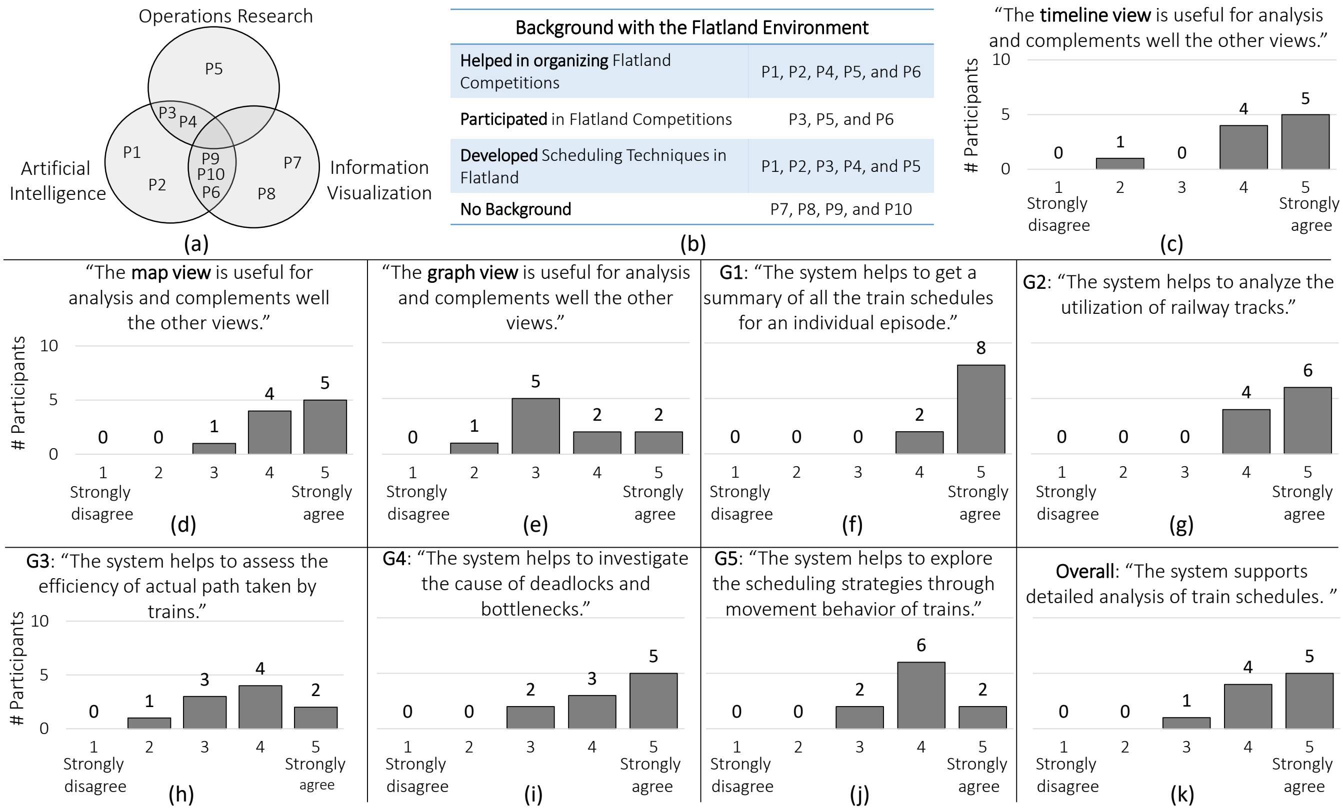

Figure 5: Experience of the experts in the three areas (a) and their background with the *Flatland* environment (b). Expert ratings of the three views of the visualization (c-e), analysis goals (f-j), and the overall system (k).

low ratings for analysis goal **G3** (Figure 5h), P4, who gave a rating of 2, did not provide any details. The experts commented about the small size of the map view (P1, P9) and suggested to include more linking between the timeline and map views (P7).

Overall System: All but one expert agreed with the overall statement that *the system supports detailed analysis of train schedules* (Figure 5k). Experts liked that the system is intuitive (P3, P8), interactive (P8, P10), provides the ability to analyze multiple aspects of the data (P2, P6, P9), and is useful for extracting insights (P1, P4, P7). However, experts also mentioned that comparing different scheduling approaches is difficult (P9) and highlighted that the agent observations (what each train saw; specific to reinforcement learning) are missing (P1, P3). Five experts (P1, P2, P3, P4, P5) commented that the visualization is helpful to diagnose specific situations (e.g., deadlocks) and debug scheduling solutions. Four experts (P4, P8, P9, P10) reported the usefulness of the visualization in analyzing the scheduling strategies and unexpected behaviors. In addition, P1 mentioned that the visualization is useful to experiment with new reinforcement learning solutions, while P7 highlighted its usefulness for improving agent performance. The experts also listed several missing features they would use such as filtering capabilities to analyze only specific trains (P1), the ability to select a time range for analysis (P9), the need of overview to select specific episodes (P9), and especially the ability to compare two scheduling approaches (P4, P6, P7, P8).

6.4 Limitations and Discussion

Due to the online setup of the feedback study, experts explored the tool freely without being monitored by us. However, given the informed answers they provided to the qualitative questions, we do not have reasons to believe that they did not sufficiently engage with the tool to make an informed judgement. Generally, the recruitment through connections to the community and personal invitation might have biased the results as experts potentially replied more positively and compliant, although we tried to counterbalance by asking directly for criticism and options to improve.

The expert feedback indicate that the proposed visualization approach fulfills the analysis goals, while addressing the specific challenges of the *Flatland* environment. However, the experts also suggested valuable features to enrich the analysis of scheduling techniques. Acting on the expert feedback, we extended our approach to include summary statistics and comparison features. Since the experts requested a detailed simultaneous comparisons among two scheduling techniques (P4, P6, P8), we focused on enabling comparison between two episodes in the extension. Since the data logs in *Flatland* do not include agent observations, it was not possible to add this feature in our approach.

7 Application: *Flatland 2020 NeurIPS Competition*

To illustrate one specific use case, we apply our approach to analyze the submissions to the *Flatland NeurIPS 2020 Competition*. We used the second version of the tool, which includes comparison features. The competition was won by an operations research (`OR`) technique by team `old_driver`. The next three ranks were awarded to three reinforcement learning (`RL`) based solutions: 2^{nd} position: `jbr_hse`, 3^{rd}: `netcetera`, and 4^{th}: `marmot`. For evaluation, the competition organizers used different levels with varying grid sizes. Within each level, ten maps with different rail network layouts and rates of malfunctions were used. Each team's submission was evaluated and compared on the mean normalized score to determine the final ranking. Generally, completing more levels with higher percentage of trains that reach their destination in lesser time leads to a higher

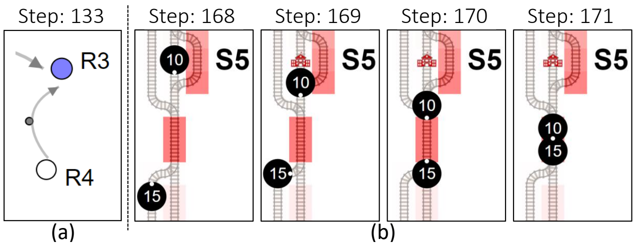

Figure 6: (a) The animated graph shows movement of trains between the selected regions in step 133, while the map view (b) shows the occurrence of a deadlock between two trains (episode `Level 11, Map 1` by team `marmot`).

mean normalized score, among other variables that affect the score, such as, local rewards for each agent (see [15] for details). Next, we present the insights found while analyzing the winning solutions with our approach.

Deadlock Propagation: Selecting an individual episode, Figure 1 shows the train schedules by team `marmot` on `Level 11, Map 1`. Overall statistics at the top shows that ∼81% of trains (21 out of the total 26) reached their destination, with 5 trains being still on-track until the end of the episode (**G1**). From the timeline view (Figure 1b$_1$), showing the group of trains on track at the bottom, we can see that all 5 trains were involved in a deadlock (blue square boxes connected with blue lines). We also infer that the deadlock first occurred between trains *010* and *015* having different destinations (*S2* and *S5* respectively). Focusing on the two trains, using the playback controls, we navigate to the time before the deadlock. The two trains headed towards each other in opposite directions on a single track, leading to a deadlock, as shown in Figure Fig. 6b. Knowing about this ineffective coordination in a specific scenario, suitable techniques can be used to improve the performance, e.g., better agent observations or communication among agents. Later, the deadlock propagated and affected the trains *001*, *017*, and *013*, all heading towards station *S2* (**G4**).

Inefficient Paths: Investigating the path efficiency between distant stations, we select a single railway line between station *S5* and stations *S1* and *S4* as region *R2*, as shown in Figure 1c$_1$ (**G2**). Assessing trains that reached their destination, we focus on the first group of trains (✔ *Reached*) in the timeline view. From Figure 1b$_2$, we observe that seven trains (*008*, *009*, *019*, *007*, *011*, *022*, and *020*) had a much longer actual path length than the shortest possible path length (**G3**). Then, analyzing the trains that did not reach their destination ( ), we see that trains *010* and *001* have followed a much longer path than the shortest path length. The inefficient movement can be explained as both experienced malfunctions that could have altered their orginally intended route twice.

Assessing Parallel Tracks Usage: Parallel tracks have several benefits that can be utilized by the scheduling techniques. For example, considering them as one-way tracks avoids the possibility of a head-on collision, or they can be used as temporary parking spots giving priority to other trains. To assess the parallel track usage, we need aggregated information on the direction and the movement of trains on the parallel tracks. We select three regions: two tiles on each of the parallel tracks (*R4* and *R5*) and a tile on a railway line common to trains using either parallel tracks (*R3*), as shown in Figure 1c$_2$. From the aggregated graph (Figure 1d), we observe that the reinforcement learning approach inefficiently used only one of the two parallel tracks (one with region *R4*) to move trains in both directions.

Comparing Usage of Parallel Tracks: To analyze the differences in parallel track utilization among the two scheduling techniques, we select three single tile regions on a parallel track in the map (Figure 3c). From the pie charts on the links between regions

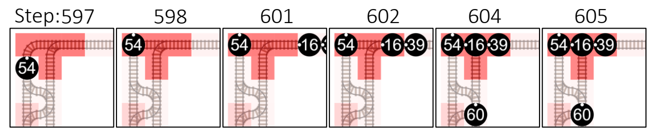

Figure 7: Trains stuck by waiting forever (not a deadlock) in `Level 19, Map 2` scheduled by the `RL` approach of team `jbr_hse`.

R2, *R3*, and *R4* in Figure 3d, we observe that `OR` strictly uses the two parallel tracks to move trains in the two directions. The `RL` technique also demonstrates this, but in few instances it used the left track from *R3* to *R2* (orange slice in pie chart) to move trains in an upward direction rather than the track on the right from *R4* to *R2* (**G2** and **G6**). This was a specific way in which the `RL` approaches improved towards the end of the competition. Observing the data from the first iteration, the leading `RL` submission in the early phase of the competition shows both the parallel tracks being used for moving trains in both directions (Supplementary Figure 2 [2]) (**G5**).

Frozen, Unable to Recover: Next, we compare the winning `OR` and `RL` scheduling approach of the competition. In episode A, we select the `OR`-based approach by team `old_driver`, as episode B we select the `RL` solution by team `jbr_hse`, and `Level 19, Map 2`. From the statistics at the top of Figure 3, we observe that 90 trains had to be scheduled. The `OR` approach was able to successfully schedule all the trains to their destination. However, using the `RL` approach, only 69 trains reached their destination, while 5 trains were still on-track and 16 trains were still waiting to be scheduled before the episode timed out. Focusing on the group of trains in the `RL` approach that were still on-track (✔), we observe that five trains (*014*, *039*, *060*, *054*, and *016*) were not blocked (absence of colored squares and connecting lines), but stood still for a long period of time (white gaps) after showing some movement (orange lines) (**G1**). Continuing the investigation in the map view, from the occupancy difference heatmap, we observe (three) orange colored tiles ( ) on the top of the rail network, indicating that the trains from episode B occupied these tiles for a much larger amount of time than trains in episode A. Among them, to analyze the usage of a region in the rail network (**G2**), we select a tile that had a junction ( ), resulting in a region-of-interest labelled *R1*. Examining usage of the region for individual trains, we hover over the region and observe that one of the five on-track trains (train *016*) spent a lot of time staying on top of the tile, until the episode ended (Figure 3a). This is intriguing as more trains in the `OR` passed through the tile than in the `RL` approach (pie chart inside *R1* in Figure 3d and tooltip in Figure 3b) (**G2** and **G6**). In Figure 7, we see that the trains got delayed (not blocked) because train *060* froze due to obstacles on the two possible paths ahead, bringing three other trains (*016*, *039*, and *054*) to a standstill (**G4**). The observations suggest that the `RL` approach is unable to plan a conflict free path for all the trains in certain scenarios, e.g., when there are obstacles on all possible paths ahead (**G5**).

Recovering from Malfunctions: Overall, the `OR` approach took much less time (colored lines in the episode timeline ticks at the top) (**G1** and **G6**). Considering the fact that in the episode, the `OR` approach witnessed more number of malfunctions in the beginning (histograms at the bottom), it shows that the approach was able to deal with unplanned malfunctions effectively. The actual path lengths of the trains that reached their destination in both episodes seem to be similar (orange and blue bars on the right of the timeline view), demonstrating a similar path efficiency of the two approaches (**G3**). However, for the `OR` approach, we observe huge differences between actual and shortest possible path lengths in some trains with white gaps in their timeline, e.g., train *042*, *048*, and *072* (longer blue bars than the green ones). This suggests that some trains in `OR` had to wait long and had to move far to reach their destination (**G5**).

Failing Cases in RL: We investigated the failing cases of agent coordination among winning RL approaches and documented our findings (Supplementary Table 1 [2]). For analysis, eight episodes were selected in which not all trains reached their destination. The analysis revealed that `jbr_hse` often runs out of time (due to episode timeout), has indefinite waiting times in areas with high number of junctions, and trains often just stay on top of junctions (**G1** and **G2**). In contrast, trains in `netcetera` often simply head towards another train from opposite direction without waiting for them (**G5**). Trains scheduled by `marmot` also exhibit a similar behavior many times. In addition, sometimes, `marmot` moves the trains from their origin towards another train coming from opposite direction on the same track, leading to a deadlock (**G4**).

8 Scalability and Generalizability

As demonstrated, the timeline view is able to represent ∼30 trains without scrolling (a typical number in mid-size networks of *Flatland*). With grouping the trains based on their status and using vertical scrolling in the timeline view, the approach was found useful to analyze episodes with up to 90 trains (Section 7). However, the details in the map view can become too small in case of large rail networks. The approach is limited to comparing two episodes only. One reason is that the timeline groups the trains based on their end status (✔, , ⏸) and their combinations would grow exponentially with the number of episodes to compare.

While our approach has been developed for the *Flatland* environment, we see potential of its use in related scenarios. The proposed timeline representation, interactive region definition, and the graph abstraction can be applied similarly for analyzing agent behavior in other scenarios with movement on constrained route networks. For instance, in analyzing coordination failures in multi-agent driving [18], e.g., identifying accident prone areas, deadlocks, etc. The approach can also be used to understand path planning efficiency, delays, and coordination of airport surface operations (e.g., fuelling, passenger boarding, luggage transit, etc.) through their movement on fixed paths connecting runways and the airport [16].

9 Lessons Learned

Based on our experience gained designing the approach and collected expert feedback, we reflect upon lessons learned that can be helpful for researchers building visual analytics solutions for analyzing multi-agent movement behavior:

Preserve a Static Map of Temporal Behaviors. For analyzing the interdependent behavior from spatio-temporal attributes in multi-agent scenarios, relying on animation does not work well as it demands a high cognitive load. Analysts need to remember a lot of information that changes quickly, e.g., tracking the movement of a group of agents. Also, common spatio-temporal visualizations, e.g., 3D space-time cubes, enable the users to change the point-of-view leading to new insights. However, a change in point-of-view challenges the mental map of the analyst. As a result, during the analysis, it becomes difficult to remember the spatio-temporal attributes of multiple agents simultaneously. We used a fixed timeline to show actions of each agent in rows, grouped the rows based on the agent status at the end of an episode, and ordered them based on their starting times within each group. This helped the users to construct a mental map about multi-agent behavior (e.g., blocked agents early in the episode), while interactions with the interface provided details without changing the overview of their actions.

Interactively Define Spatial Focus and Map it to Time. Combining spatial and temporal information in a single visualization is challenging. However, an in-depth analysis of both is required to understand complex simultaneous behavior of agents. To support this, many spatio-temporal visualizations—including our approach—have separate views for each attribute with simple linked interactions, e.g., brushing and linking of agents. However, to study group behavior in multi-agent scheduling scenarios, they are not enough. We learned two things. First, domain specific encodings and interactions help the analysts to focus on specific regions, e.g., selection of a railway line between two junctions. Second, showing the effect of spatial selection on the temporal dimension helps reveal unexpected insights. For instance, on hovering over the selection of a region-of-interest in the map view, gray semi-transparent rectangular boxes are drawn in the timeline view. This helps in discovering extended stays in a region (e.g., *Train 016* in region *R1* in Figure 3a) or leads to insights such as cyclic movement of agents through a region-of-interest (e.g., *Train 010* through region *R2* in Figure 1).

Abstract Space and Aggregate Multi-Agent Movements. To analyze the multi-agent specific scheduling behaviors, e.g., usage of parallel tracks, experts need to analyze the collective movement of agents through specific regions. We learned that abstracting the exact movement of agents over user-defined regions-of-interest helps to focus on movements between selected regions and to gain insights about the scheduling technique (Section 7). This lesson aligns with the idea of spatial and temporal abstraction proposed by Andrienko et al. [4] to analyze patterns in mass mobility data. Generally, the abstracted and aggregated representations are put first in top-down exploration. However, in multi-agent scheduling behavior analysis, we realized that a bottom-up exploration was first necessary to identify relevant specific agents and regions-of-interest. In a different context, van den Elzen and van Wijk [26] describe a similar bottom-up approach as *Detail to Overview via Selections and Aggregations*.

10 Conclusions and Future Work

We presented a visualization approach to analyze complex spatio-temporal train scheduling behavior on virtual rail networks for the *Flatland* environment. We interviewed three experts from the *Flatland* community to formulate the analysis goals of the visualization. Then, we designed our approach in an iterative process, based on the analysis goals and the feedback of ten experts. The discovered insights confirm that the proposed views give different perspectives on the train movements and the applied scheduling techniques. Finally, we presented lessons learned to help future research visualizing multi-agent movement on fixed-track networks. The close collaboration with the *Flatland* community helped us to approach the key community members, to identify the research challenges, to facilitate development and evaluation of the system. The encouragement and support from the community prompted us to release the tool for all competition participants, during which it was awarded the first prize in the community contributions category [1]. Future work includes integrating the planned modifications in future *Flatland* competitions, e.g., time windows to start the trips and varying speed of trains.

Acknowledgments

We thank Florian Laurent for helping with queries related to the *Flatland NeurIPS 2020 Competition*, the *Flatland* community members, and all experts who contributed to this research. This work is partly funded by MERCUR (project: “Vergleichende Analyse dynamischer Netzwerkstrukturen im Zusammenspiel statistischer und visueller Methoden”).

References

[1] NeurIPS 2020 Flatland Winners, 2020. https://bit.ly/3K7wljR Accessed: March, 2021.

[2] S. Agarwal, G. Wallner, J. Watson, and F. Beck. Supplementary material: Spatio-temporal analysis of multi-agent scheduling behaviors on fixed-track networks, 2022. doi: 10.17605/OSF.IO/R2YZA

[3] G. Andrienko, N. Andrienko, W. Chen, R. Maciejewski, and Y. Zhao. Visual analytics of mobility and transportation: State of the art and

further research directions. *IEEE Transactions on Intelligent Transportation Systems*, 18(8):2232–2249, 2017. doi: 10.1109/TITS.2017.2683539

[4] G. Andrienko, N. Andrienko, G. Fuchs, and J. Wood. Revealing patterns and trends of mass mobility through spatial and temporal abstraction of origin-destination movement data. *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2120–2136, 2017. doi: 10.1109/TVCG.2016.2616404

[5] S. S. Bae, F. Rossi, J. V. Hook, S. Davidoff, and K.-L. Ma. A visual analytics approach to debugging cooperative, autonomous multi-robot systems' worldviews. In *IEEE Conference on Visual Analytics Science and Technology*, pp. 24–35, 2020. doi: 10.1109/VAST50239.2020.00008

[6] R. Basra, K. Lü, G. Rzevski, and P. Skobelev. Resolving scheduling issues of the London underground using a multi-agent system. In *Holonic and Multi-Agent Systems for Manufacturing*, pp. 188–196, 2005. doi: 10.1504/IJISTA.2007.011569

[7] W. Chen, F. Guo, and F.-Y. Wang. A survey of traffic data visualization. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):2970–2984, 2015. doi: 10.1109/TITS.2015.2436897

[8] M. Cummings and P. Mitchell. Managing multiple UAVs through a timeline display. In *Infotech@Aerospace*, 2005. doi: 10.2514/6.2005-7060

[9] A. Cuppari, P. L. Guida, M. Martelli, V. Mascardi, and F. Zini. Prototyping freight trains traffic management using multi-agent systems. In *Proceedings of International Conference on Information Intelligence and Systems*, pp. 646–653, 1999. doi: 10.1109/ICIIS.1999.810360

[10] A. Dorri, S. S. Kanhere, and R. Jurdak. Multi-agent systems: A survey. *IEEE Access*, 6:28573–28593, 2018. doi: 10.1109/ACCESS.2018.2831228

[11] H. Guo, Z. Wang, B. Yu, H. Zhao, and X. Yuan. TripVista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection. In *IEEE Pacific Visualization Symposium*, pp. 163–170, 2011. doi: 10.1109/PACIFICVIS.2011.5742386

[12] K. Hager, J. Rauh, and W. Rid. Agent-based modeling of traffic behavior in growing metropolitan areas. *Transportation Research Procedia*, 10:306–315, 2015. doi: 10.1016/j.trpro.2015.09.080

[13] J. Jin, R. Sanchez, R. T. Maheswaran, and P. Szekely. VizScript: On the creation of efficient visualizations for understanding complex multi-agent systems. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, p. 40–49, 2008. doi: 10.1145/1378773.1378780

[14] E. Mai, M. Backman, and R. Hranac. Visualizing bus schedule adherence and passenger load through Marey graphs. In *18th ITS World Congress*, 2011.

[15] S. Mohanty, E. Nygren, F. Laurent, M. Schneider, C. Scheller, N. Bhattacharya, J. Watson, A. Egli, C. Eichenberger, C. Baumberger, G. Vienken, I. Sturm, G. Sartoretti, and G. Spigler. Flatland-RL: Multi-agent reinforcement learning on trains. *arXiv*, 2020. arxiv.org/abs/2012.05893.

[16] R. Morris, C. Pasareanu, K. Luckow, W. Malik, H. Ma, T. K. S. Kumar, and S. Koenig. Planning, scheduling and monitoring for airport surface operations. In *Workshop on Planning for Hybrid Systems (PlanHS)*, 2016.

[17] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi. Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. *IEEE Transactions on Cybernetics*, 50(9):3826–3839, 2020. doi: 10.1109/TCYB.2020.2977374

[18] A. Pal, J. Philion, Y.-H. Liao, and S. Fidler. Emergent road rules in multi-agent driving environments. In *International Conference on Learning Representations*, 2021.

[19] C. Palomo, Z. Guo, C. T. Silva, and J. Freire. Visually exploring transportation schedules. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):170–179, 2015. doi: 10.1109/tvcg.2015.2467592

[20] T. Sobral, T. Galvao, and J. Borges. Visualization of urban mobility data from intelligent transportation systems. *Sensors*, 19(2):332, 2019. doi: 10.3390/s19020332

[21] J. Suarez, Y. Du, P. Isola, and I. Mordatch. Neural MMO: A massively multiagent game environment for training and evaluating intelligent agents. *arXiv*, 2019. arxiv.org/abs/1903.00784.

[22] K. P. Sycara. Multiagent systems. *AI Magazine*, 19(2):79–92, 1998. doi: 10.1609/aimag.v19i2.1370

[23] D. Teodorović. Transport modeling by multi-agent systems: A swarm intelligence approach. *Transportation Planning and Technology*, 26(4):289–312, 2003. doi: 10.1080/0308106032000154593

[24] The Flatland Community. The dataset of recorded episodes from Flatland 2020 NeurIPS Competition, 2021. `https://www.aicrowd.com/challenges/flatland/dataset_files` Accessed: March, 2021.

[25] C. Tominski, H. Schumann, G. Andrienko, and N. Andrienko. Stacking-based visualization of trajectory attribute data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2565–2574, 2012. doi: 10.1109/TVCG.2012.265

[26] S. van den Elzen and J. J. van Wijk. Multivariate network exploration and presentation: From detail to overview via selections and aggregations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2310–2319, 2014. doi: 10.1109/TVCG.2014.2346441

[27] T. von Landesberger, F. Brodkorb, P. Roskosch, N. V. Andrienko, G. L. Andrienko, and A. Kerren. MobilityGraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):11–20, 2016. doi: 10.1109/TVCG.2015.2468111

[28] P. Xu, H. Mei, L. Ren, and W. Chen. ViDX: Visual diagnostics of assembly line performance in smart factories. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):291–300, 2016. doi: 10.1109/tvcg.2016.2598664

[29] W. Zeng, C.-W. Fu, S. Müller Arisona, A. Erath, and H. Qu. Visualizing waypoints-constrained origin-destination patterns for massive transportation data. *Computer Graphics Forum*, 35(8):95–107, 2016. doi: 10.1111/cgf.12778