MASTERARBEIT | MASTER'S THESIS

Titel | Title

Visual data analysis of time-based transport optimizations - IronFlow-Viz

verfasst von | submitted by

Dragoş-Cristian Barbu

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2024

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 921
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Informatik
Betreut von | Supervisor:	Univ.-Prof. Torsten Möller PhD

Aknowledgements

I would like to express my deepest gratitude to my supervisor Prof. Torsten Möller, for his constant support, feedback, and patience throughout this process. He pushed me to refine my work and helped me stay focused on my goals and for that I am grateful.

I would also like to thank Biljana Roljic and Prof. Karl Franz Dörner, for their support and collaboration in this project.

The financial support by the Austrian Federal Ministry of Labour and Economy, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association is gratefully acknowledged.

Zusammenfassung

Ein großer Aufwand wurde bereits unternommen, um Transporte zu optimieren und die Ergebnisse dieser Optimierungen zu visualisieren. Solche Optimierungen werden durch vorsichtige Routenplanung erreicht, mit dem Ziel, Zeit und Distanz zu reduzieren und den Güterdurchsatz zu erhöhen. Um die beste Route für einen Transport zu finden, setzten Unternehmen auf Fachexperten, während sich der Schwerpunkt in letzter Zeit auf die Entwicklung von Optimierungsalgorithmen verlagert hat. In einem Bereich, in dem jede Sekunde zählt, sind Optimierungen entscheidend, deshalb werden neue Ansätze zur Optimierung oft in Frage gestellt. Die Ergebnisse der Algorithmen variieren und können komplex zu bewerten sein. Um die Gültigkeit und Korrektheit zu demonstrieren, benützen Unternehmen in der Regel ein Visualisierungstool.

In dieser Arbeit wollen wir das Misstrauen gegenüber kurzfristigen Transportoptimierungen zu beseitigen, indem wir eine visuelle Analysetool vorschlagen, die es einem Experten im Transportwesen ermöglicht, historische Daten zu analysieren und geplante zukünftige Routen zu visualisieren. Das Design besteht aus drei Ansichten: absolut, relativ und topologisch, wobei jede eine andere Datenfacette darstellt. Wir zeigen, wie das Vertrauen von Transportplanern in Optimierungsalgorithmen gestärkt werden kann und wie dasselbe Tool zur Weiterentwicklung des Optimierungsalgorithmus genutzt werden kann.

Abstract

An extensive effort has been previously made to optimize transports and visualize optimization results. Such optimizations are reached through careful planning of routes, while trying to reduce time and distance and increase throughput of cargo. In order to come up with the best route for a transport, companies have been previously using domain experts while more recently, the focus has been changed to developing optimization algorithms. In a domain where every second matters, optimizations are crucial, thus finding new ways to optimize is seen with skepticism. Results of algorithms vary and can be complex to evaluate, therefore a visualization tool is generally needed to demonstrate the validity and correctness.

For this thesis we intend to dispel the distrust and skepticism in short transport optimizations by proposing a visual analysis solution, which enables a transport domain expert to analyze historical data and visualize future planned routes. The design is built from three views: absolute, relative, and topological, each showing a different data facet. We show how transport planners' trust in optimization algorithms can be achieved and how the same tool can be used to further develop the optimization algorithm.

Visual data analysis of time-based transport optimizations - IronFlow-Viz

Contents

1 Introduction

The transportation problem is concerned with finding the optimal solution for delivering items from an origin to a destination using the least amount of resources possible. It has been formed and formally analyzed starting in the 1930s [2], and multiple optimization algorithms have been proposed since then. As economic pressures have pushed companies to become more efficient and effective [15], specialized optimization solutions are being developed for individual problems. However, this trend has resulted in increasingly complex and harder-to-understand optimization algorithms, leading to a lack of trust among transport planners who are well-versed in the intricacies of their specific domain.

Transport planners need to validate the underlying optimization algorithm. Hence, they need to analyze proposed routes, transport patterns, and outlier transports. Visual analysis tools can increase planners' trust in optimization algorithms by presenting the results of the algorithms and comparing them to historical routes based on the planners' experience. Consequently, there is a growing need for visual analysis tools to help transport planners understand the optimization process and build trust in its solutions.

These tools can provide new insights that might drive change in workflows, for example, by allowing the user to analyze geospatial patterns. Moreover, maps and other visualizations enable users to monitor and analyze proposed routes and historical or real-time data. Many existing tools focus on a map [9, 12, 18], which is then enhanced through data encoded in different visual channels. However, they are limited in the number of data facets they can present due to the map visualization taking up too much space. Additionally, they have issues depicting changes over time in their visualization without the help of animations, which, according to Fisher [10], have significant drawbacks for users: confusion, less accuracy, and slowness in discovering the information required. In the transport domain, it is imperative to have a clear and concise route overview based on the origin-destination locations and the progression over time: animations usually do not suffice.

As Marey has shown with his graphical train schedule in 1878 [16], location can be encoded into a single linear list of points when geographical coordinates are not required. Building on such a graph, our approach removes the need for a map and encodes the location into a single scalar instead of a pair of coordinates. As such, we can use existing visualization techniques to introduce time without the need for animations. Furthermore, we show events in time in two different ways: through an absolute and a relative timeline. Consequently, the user can compare transports while also being able to follow the events chronologically at a glance.

Our contribution includes a comprehensive exploration of the space of visual analysis tools applicable to planned transports, the development of a design that establishes confidence in the optimization algorithm underlying it, and its evaluation.

Terminology

Table 1 lists key terms and definitions relevant to IronFlow-Viz and the domain analyzed. This table promotes clear and consistent communication about the subject by establishing a common vocabulary. We define each term concisely and clearly to facilitate readers' understanding and show how our design could be extended to the general transportation problem.

Term	Definition
Truck	Depending on the transport domain, the cargo is moved with the help of various motor vehicles. In this case, we are talking about straddle carriers or a more specific term: PHW (German: Portalhubwagen).
Item	In the domain analyzed, we are talking about trucks delivering large metal slabs.
Node	Items are being picked up and delivered from different locations. Generally, we will use the words nodes, origin, and destination.
Transport	The process of moving items from the point when the truck starts picking up items until it has dropped all items, regardless of how many intermediate stops have been made.
Transport leg	The movement from one node to another is called a transport leg, irrespective of how many items have been picked up or dropped between the two nodes.
Transshipment	A transshipment is the process of moving an item from node A to node B by using an intermediate node C, where the item is dropped by one truck and later picked up by another truck.
Shuffling time	The items stored in the company's yard are placed in rows and stacked on top of each other. As such, it takes time for a truck to pick up an item placed on a specific row and in a specific stack. This process is called shuffling.
Throughput	Number of items moved in the time frame of the loaded data set.

Table 1: *Terminology*

2 Design studies and development process

Our work arose from the need to explain the results of a novel optimization algorithm developed for improving the transport of heavy materials in a large industrial storage facility to local personnel with decades of experience in route planning. IronFlow-Viz has been developed adhering to the Nested Model for Visualization Design and Validation [20] and the 9-step Design Study Methodology developed by Sedlmair et al. [17]. First, we conducted two interviews with the logistic experts at the local university. They introduced us to two transport planners from their company partners, whom we interviewed. Hence, there are two different users with different requirements and tasks. On the one hand, we have the planner, who plans the daily routes at the company and will be our main user. On the other hand, we have the researcher and algorithm developers, our secondary users.

The following steps in the Design Study Methodology led us to conduct a requirements analysis and to design multiple low-fidelity prototypes evaluated with our two types of users. The designs were developed in a feedback loop, allowing us to choose the best visualizations for the tool.

Based on the feedback we received from our low-fidelity prototypes, we started developing a web tool based on d3.js. Frequent (and at times short) meetings with the secondary user allowed us to better understand the needs and complexity of the tasks, to notice problems early on, and, therefore, to improve our prototype.

When we reached what we believed to be a robust solution, we presented it to our industrial collaborators. Next, we had two more cycles of development and feedback sessions with the primary user, where we incorporated the changes suggested and fixed bugs accordingly. Real data was used during the implementation, and various borderline cases have been presented to outline the benefits of using a visual analysis tool. In this case, edge cases are scenarios where the optimization algorithm suggested longer routes or delivering items to interim stops.

In our last step, we additionally presented the high-level prototype to visual analysis experts who were seeing the tool for the first time. We used the System Usability Scale (SUS) [11] to understand the work done and identify possible improvements.

3 Requirements Analysis

During our initial interviews, we gathered, analyzed, and summarized the requirements of the two user types. The result of this requirement analysis is outlined here.

- Main user: Planner
  - Use a compact view to see where a transport is heading
  - Understand the output and quality of the optimization algorithm
  - Learn how a transshipment increases the overall throughput of items
- Secondary user: Researcher
  - Analyze periods of time where the throughput is low
  - Identify patterns
  - Further improve the optimization
  - Use statistical data to explain output

To better express the requirements, we have identified a set of questions the users ask to gain insight into the optimization results:

Q1 What items must be picked up and delivered in the next hour?

Q2 What is the route for this truck?

Q3 Are the trucks under- or over-loaded?

Q4 Why is there a transshipment?

Q5 What is the overall throughput, and how does it compare to historical data?

Q6 Which are the most used nodes?

Q7 Are there multiple items that need to be delivered to the same destination from the same origin?

Q8 Why is this truck waiting?

Q9 How long does an item stay in an area after it is transshipped?

4 Data and task abstraction

Our data set is composed of three main dimensions: trucks, nodes, and items, which are connected by transport routes. Therefore, the input consists of transports with the following attributes: origin, destination, and the item location in the node; individual item weights; truck information; time required: transporting, waiting, and shuffling. Regarding the origin of the data set, transport routes can be the output of the optimization algorithm developed by the researchers or historical data, which the planner has suggested based on their experience.

Analyzing the requirements above, we can outline the following high-level tasks, which we tie to the set of questions to emphasize their connection:

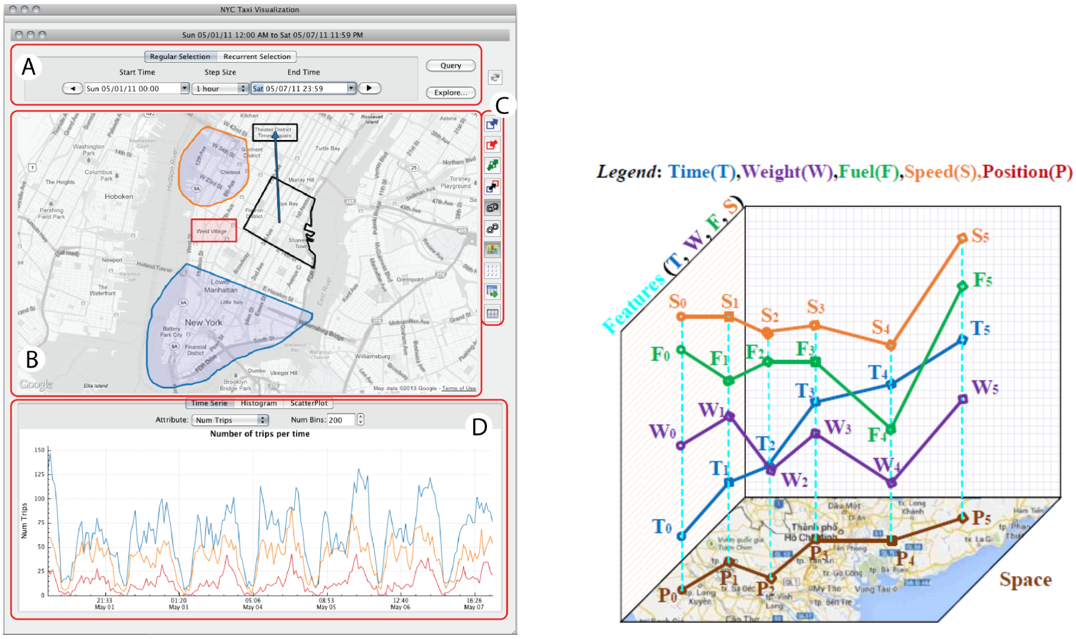
T1 **Route planning:** The main task of the primary user is to plan the transport routes. (Q1,Q2)

T2 **Monitoring transports:** It is necessary to track the shipment both in geo-spatial coordinates and regarding the status of the cargo and transport means (Q3, Q8).

T3 **Pattern recognition:** Identifying patterns allows the user to understand better the characteristics of items delivered between two nodes and allows them to make changes to increase throughput. (Q5, Q6, Q7)

T4 **Incident investigation:** The user might want to know what incidents lead to congestion in traffic. The first step in such an investigation is finding the outlier and then comparing the characteristics of data around that incident and comparing it to normal behavior. (Q8)

T5 **Origin-Destination analysis:** To have an overview of a node, a common task is to analyze incoming and outgoing transports. Compared to other tasks, in this one, the user is specifically interested in exploring the route as a trajectory, meaning the direction is also essential (Q4, Q7)



(a) TaxiVis [9]

(b) Space-Feature-Time cube [14]

Figure 1: *Route planning and monitoring tools*

T6 Transshipment analysis and alternatives: Transporting a single item on one transport is not optimal. As such, the users search for alternative transports while the optimization algorithm offers transshipment solutions. (Q4,Q9)

5 Related work

Visual analysis is an overarching name for tools based on visualizations, enabling users to analyze large data sets. They offer interactivity to present different facets of the data and encode information through shapes, colors, positions, etc. Trying to understand all the aspects of a large dataset is a difficult task, even for domain experts. Therefore, visual analysis techniques have been developed to help them. The following lines will discuss some techniques and solutions for transport problems.

5.1 Route planning and monitoring

The first category consists of tools trying to find the best route for a specific transport and monitoring it to see if everything is working as intended. What best means depends on each use case. Regarding our tasks, these tools focus (only) on T1 and T2.

The most apparent approach to visualization is an interactive map that can be moved on the x and y-axis. On top of it, various information can be drawn, for example, heatmaps as shown in SmartAdP [12] and TaxiVis [9], or routes and aggregated trips like in the tool developed by Riveiro et al. [18]. Adding temporal changes is generally challenging to accommodate.

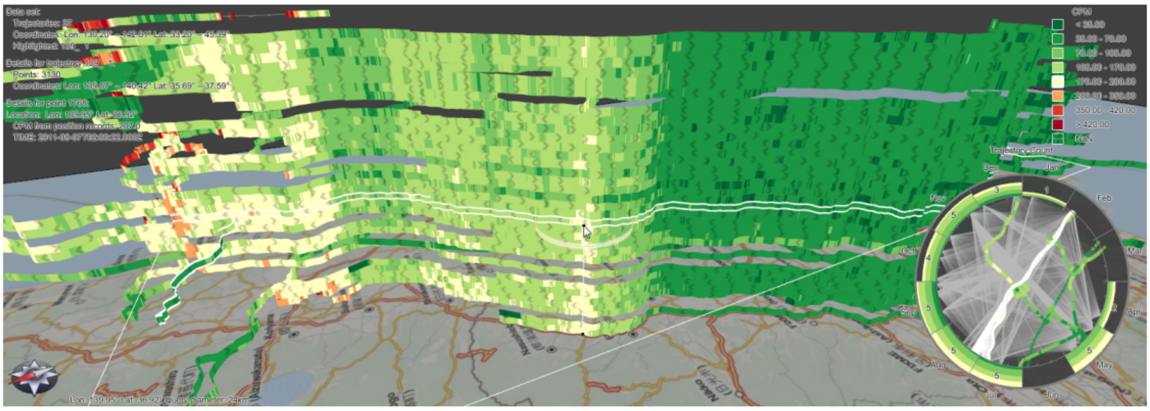


Figure 2: *Wall of color-coded bands. [7]*

One clever way to deal with changes over time is the space-time cube. It was introduced in the 1960s and is a 3-dimensional cube that consists of a map on the x and y coordinates while the z-axis handles the temporal feature. Kraak [14] revisited the topic and implemented a visualization tool to allow interactivity on the space-time cube. Furthermore, it focuses on filtering and querying the dataset to change the cube's size and rotation, allowing the user to analyze the data better. The downside of the 3D visualization is that it only allows a small number of transports to be analyzed simultaneously.

Tominski et al. [7] are tackling the issue of multiple routes in a 3d space-time cube by grouping all the trips with similar origins and destinations and stacking them on top of each other (Figure 2). The result is a wall of bands color-coded according to the selected feature. In this solution, the user has to choose a set of locations, and in the visualization, they will see all the trips that use the specified points. From a temporal point of view, the stacking of bands represents the order in which the trips have taken place, be it for the start, the end, or an arbitrary common time event of the trip. Multiple arrow shapes are drawn on top of the band to specify the direction of the transport. As the transports are aggregated, analyzing a single transport and proposing alternatives as required by T6 is hard.

Wunderlich et al. [13] developed a visual analysis tool that enables the user to plan a train trip considering delays. They started from Marey's graph and ended up with a simpler view similar to a Gantt chart because the user is not interested in the interim stops but in the time needed between switching trains, which is generally affected by delays. The tool focuses on a single transport for which it proposes alternatives (T1 and T6) but does not offer solutions for any other tasks.

5.2 Incident investigation

While route planning tools look to the future for cost optimizations, other visualizations focus on the past to analyze an exact moment to see what went wrong or what went well (T4). When looking at the past, we often try to investigate incidents and why they happen. In the transport field, an incident might cause a delay and create a chain reaction, slowing down other transports.

Incidents have different levels of consequence, from slowing down and delaying transport to loss of human life. Riveiro et al. [18] created a visual analysis tool (TripMiner) to identify crashes in

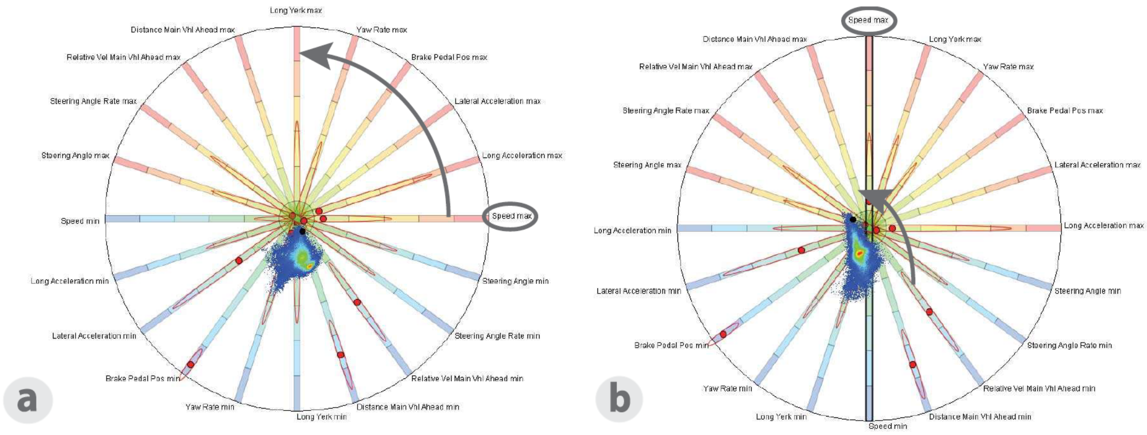


Figure 3: *Driving patterns defined by 20 features [18]*

which trucks were involved and to analyze the moments leading up to them. It uses a clustering algorithm to identify specific driving patterns and try to find a normal behavior. Next, the user can compare a particular trip against the standard trip cluster and check if the trip has any deviations from the expected behavior and how this influences the transport. The main visualization is a radial view based on RadViz [19], which displays the clustering features and allows the user to analyze their influence on the algorithm by swapping the segments around and analyzing changes in the scatter plot to see if the swapped feature has any impact on the outcome.

5.3 Discovering patterns

The last group consists of visualizations which enable the user to find patterns that would otherwise be missed. Knowing that many transport trips are happening between points A and B represents a simple pattern that can be identified from the data. However, transport systems generally work with big datasets and complex data models. Consequently, a visualization tool is superior. Furthermore, patterns could occur at places the user does not expect so that the domain expert might miss it, while a visualization tool would highlight it.

Traffic analysis tools have been developed to better overview urban transport to optimize planning or reduce traffic. This tool category falls under the term Intelligent Transport Systems (ITS). Taxis are a type of transport that is always in movement and greatly influences congestion, especially in big cities like New York. TaxiVis [9] is a visual analysis tool trying to find patterns in taxi routes across time and space. TaxiVis enables the user to query the dataset on multiple dimensions (filters include but are not limited to distance traveled, pickup, dropoff, and trip duration) but focuses on comparing spatiotemporal slices. One visual analytics technique heavily used in visualizing large datasets is aggregation because it creates a much clearer view and highlights patterns. Ferreira et al. [9] is working with over $500,000$ taxi trips per day over two years. Consequently, they use aggregation techniques such as heatmaps or simply combining locations into neighborhoods and comparing the areas. Time slice comparisons are highlighted by analyzing the influence of Hurricane Irene (2011) and Hurricane Sandy (2012) over the number of taxi trips in the immediate time frame. Focused on aggregated data, the tool cannot analyze individual trips that T1, T2, or T6

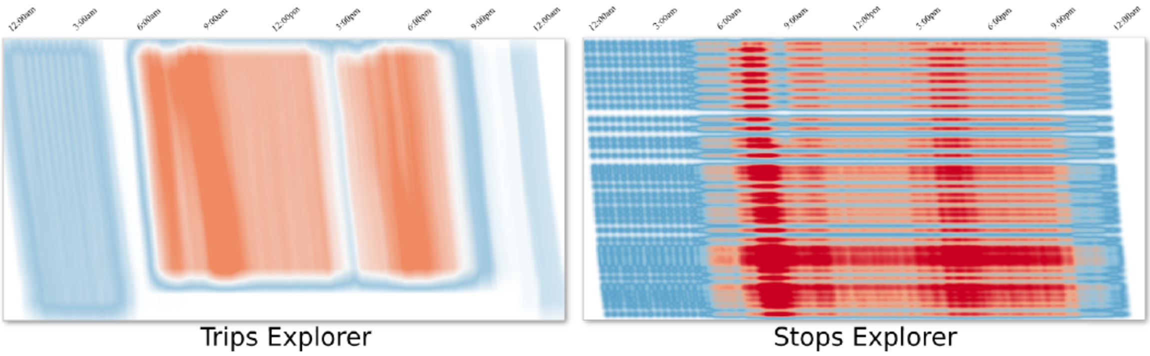


Figure 4: *Finding busy time frames in metro stations [6]*

require.

Further development of Marey's graph has been done by Palomo et al. [6] to identify patterns in public transport trips and analyze delays caused by the number of passengers. The visualization developed, TR-EX, consists of two views next to each other, one displaying trip data while the other stop data, which have a common y-axis encoding the distance between the stops of a New York City subway line. A Marey's graph built from the dataset is overplotted since there are so many subway trips in one day. Therefore, the tool uses an algorithm to reduce the clutter. The algorithm used is Kernel Density Estimation. It contains a bandwidth parameter that the user can update to smoothen out the effect of KDE and thus reveal fine or high-level details.

Similarly, pattern discovery can identify hotspots in traffic to find places where many traffic participants travel and exploit the number of people for one's gain. SmartAdP [12] is a tool that takes advantage of crowded areas to present suitable locations for placing billboards and says billboards are one of the most common ways to advertise out in the open. It uses taxi trajectories, a data mining algorithm, and visualizations to display possible solutions and their comparison to identify these places. The visualization consists of a map, a custom glyph, and a solution explorer. The map highlights patterns of high traffic and presents a custom glyph on the proposed locations. The glyph is a radial view similar to a speedometer, showing billboard-related data and average taxi speed. Lastly, the solutions can be compared in histograms and statistical information table.

TaxiVis and TR-EX excel at analyzing origin-destination routes (T5) and identifying patterns (T3). Still, extending them with route planning or alternative propositions would be hard because they focus on looking to the past and not the future. SmartAdp focuses on alternative propositions, but its interest lies in location, not transport.

5.4 Data and Task abstraction in related work

To compare the presented solutions, we show the datasets and which tasks they fulfill in table 2 and 3. We also add IronFlow-Viz to the tables to express the similarity with other tools and show that only IronFlow-Viz manages to solve the tasks developed from the requirements analysis.

Traffic and transport data can come from direct and indirect sources, e.g., a company could be monitoring its own transport fleet, or a mobile phone dataset could be preprocessed to analyze which metro stations are busy. Regarding the data source, we will group the datasets into three categories for a better overview:

Solution	D1. Devices in motion	D2. Schedules	D3 Event-based tracking
AllAboard [5]	Mobile cell phone	Bus schedule	
SmartAdp [12]	Taxi GPS		
TaxiVis [9]			Taxi trajectories
Tominski et al [7]	GPS data from Tokyo - Fukushima transports		Vessel traffic in Brest
TripMiner [18]			Sensors of freight transports
TR-EX [6]		Metro schedule	Metro trips
Wunderlich et. al. [13]		Train schedule	Train trips
IronFlow-Viz		Item slab orders	Pickup/dropoff events

Table 2: *Data abstraction in related works*

D1 Devices in motion: Nowadays, IoT devices are used to generate all kinds of data and thus can be used to report real-time attributes of a transport like speed, weight, etc. Also, real-time location can be tracked by GPS coordinates.

D2 Schedules: One common dataset group in the transport industry is data related to schedules and planning of trips. Analyzing a set schedule against the time spent on a trip can reveal delays and deviations in planned costs.

D3 Event based tracking: Lastly, data could be generated for event tracking. This data group encompasses data that can be generated by sensors or human input. It simply tracks events that happen at a specific time: the truck arrives at its destination, the ship finishes loading cargo, etc.

We have previously classified the tools by their focus, but when looking at the tasks previously defined, we can see that they fulfill more than just one and generally have a focus on one of the tasks while also partially fulfilling some other. It is important to note that from the presented solutions, only Wunderlich et al. [13] developed their tool for a general user without prior knowledge. In contrast, the others are developed for domain experts to assist them in their work. Furthermore, it is the only solution that specifically mentions transshipments. SmartAdP only handles the second part of T6, the alternative suggestion. For our researchers handling transshipments is important because it is one of the main ways the algorithm optimizes the solution and improves the total throughput.

Fully supported	
Partially supported	
Not supported	

Solution	T1	T2	T3	T4	T5	T6
AllAboard [5]						
SmartAdp [12]						
TaxiVis [9]						
Tominski et al [7]						
TripMiner [18]						
TR-EX [6]						
Wunderlich et. al. [13]						
IronFlow-Viz						

Table 3: *Tasks: T1. Trip planning, T2. Real-time monitoring, T3. Pattern recognition, T4. Incident investigation, T5. Origin-Destination analysis T6. Transshipment analysis and alternatives*

6 Prototyping

Throughout this chapter, we will present the progress of the prototype developed and explain the visualizations that compose IronFlow-Viz. We will briefly present two paper prototypes and the final high-fidelity prototype at length, a combined result of the mock-ups.

6.1 Paper prototypes

6.1.1 Paper prototype 1

The first mock-up contains three main views: absolute view, topological view, and stats view. Additionally, in Figure 5, you can see a header with helper visualizations that enable filtering and sorting in the main three views.

The absolute view is based on Marey's train schedule [16] and displays transports between nodes. The y-axis represents node locations, while the x-axis shows the time. Encoded with color, we have the truck IDs, while the width of the lines represents the number of items on a transport leg. In this view, you can clearly see what trucks do at specific times and follow their route throughout the selected timeline. The view's main purpose is to solve the T1 and T2 tasks to have a clear view of individual transports.

The topological view represents an abstraction of the company's yard where we have the nodes encoded in dots and the amount of transport between two nodes as the width of the line between

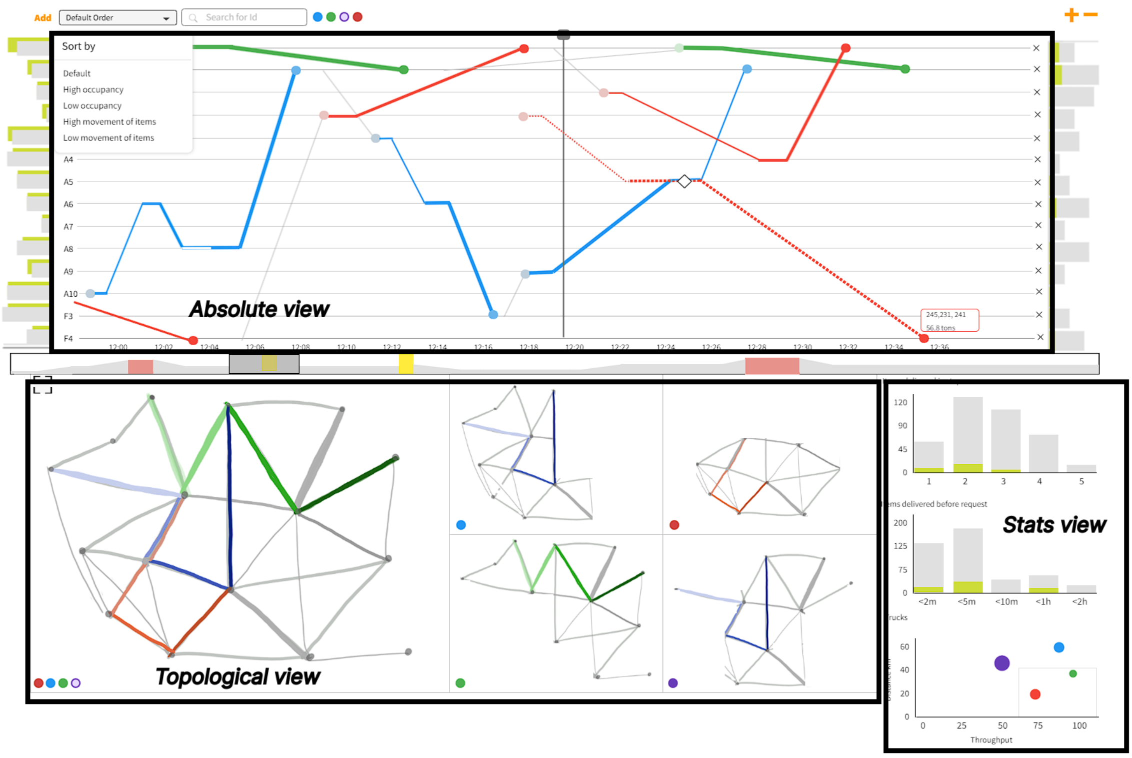


Figure 5: *Paper prototype 1*

said nodes. The color again represents the truck id. This view enables the user to identify patterns (T3) and to find locations that are used a lot or quite rarely (T5). Also, in the "small multiples" visualization [8], we can see where each truck spends most of its time.

Lastly, the stats view comprises two histograms and one scatter plot. The histograms show the number of transport legs required for an item to arrive at the destination and the time, respectively. The scatter plot shows information about trucks and the amount of work done. With these visualizations, we highlight outliers so that incidents can be more easily analyzed (T4). The final version of the scatter plot is different and will be described in more detail in the web interface chapter.

To bind the main views together, we have nodes sorting and filtering at the top-left of the view. Then we have a search for finding items, and on the top-right, we have a zoom feature that increases and decreases the timeframe displayed. To navigate through time, the user can utilize the scrollbar between the absolute and topological views. In the scrollbar, the total weight of transports at a specific time is encoded, where red rectangles represent critical points in time while yellow rectangles indicate transshipments (T6).

6.1.2 Paper prototype 2

The second mock-up is built from four views: heatmap, relative view, UpSet view, and stats view. Overall, the focus is on analyzing past data and less on planning and monitoring routes.

The relative view tries to solve tasks similar to the absolute view from the first prototype but goes about it from a different perspective. Instead of focusing on time, it focuses on duration so

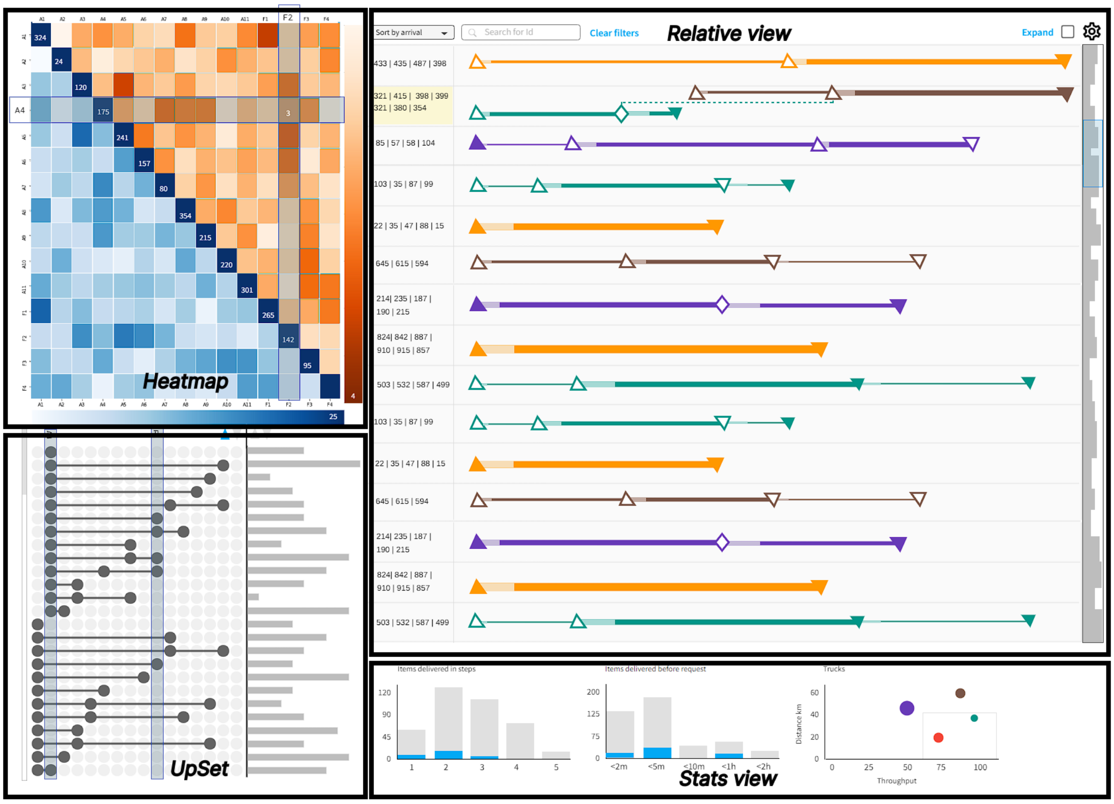


Figure 6: *Paper prototype 2*

that transports can be more easily compared. It is similar to a table where each row contains one transport. On the left of each row, we have the IDs of the items delivered, and on the right, we have a transport as a line visualization. Each transport leg in the transport is shown as a line with a width directly proportional to the items moved and a length encoding the time needed to drive from one node to another. At the start and end of each transport leg, we have a symbol: a triangle pointing up or down showing if items were picked up or dropped down, or a rhombus indicating that there was a pickup and a drop-down of items at this node. In the case of transshipments, these are shown together on one row and are connected by a dotted line. Similarly to the first mock-up, the time can be scrolled through with a scrollbar containing a histogram of transport weights. We call this the relative view because its purpose is to compare transports to each other from a time perspective. Consequently, we emphasize the duration of transports over the absolute times of pickup and delivery. The exact times can be found by hovering the mouse over the triangles. The table is, by default, sorted by the start time, but this can be changed with the sorting options on top.

The heatmap is focused on nodes and acts as a node-by-node matrix. We have the number of items on the diagonal at the specific node. On the orange side, the matrix shows the maximum number of stops on a transport between the two nodes: the darker the hue, the more stops a transport has made. Lastly, on the blue side, we have the total number of trips between the nodes in the cell: similarly to the orange side, the darker the hue, the more transports exist between the nodes. As an interaction, the heatmap displays the exact number on the orange or blue cell while hovering over it. On click, the heatmap filters the rest of the views only to display the transports between the two selected nodes. From a task perspective, the heatmap is ideal for origin-destination analysis (T5) and pattern recognition (T3).

The last two views are the stats view, similar to the one in prototype 1, and the UpSet view. The UpSet view is presented as a visualization of intersecting sets and was developed by Lex et al. [1]. In this case, it identifies all the variations of connections between the nodes by encoding the nodes into dots and the connection as a line between them. On the right side of the plot, a histogram shows how often the specific configuration happens.

6.1.3 Merging the paper prototypes

After introducing the users to the two prototypes, we gathered feedback and merged the two versions. We tried multiple variants of displaying the needed visualizations in one view, and the one the users found most usable was implemented. The final paper prototype comprises the absolute, relative, and stats views on the first page, while the topological view is hidden behind a click action. To see the topological view, the user has to click on the button below the scrollbar, and the topological one will replace the relative view. The main change is that we rotated the relative view 90 degrees to align with the timeline, and it can now be scrolled together with it. This change enables the user to simultaneously work with the relative and absolute view to analyze and compare the transports from a geospatial point of view.

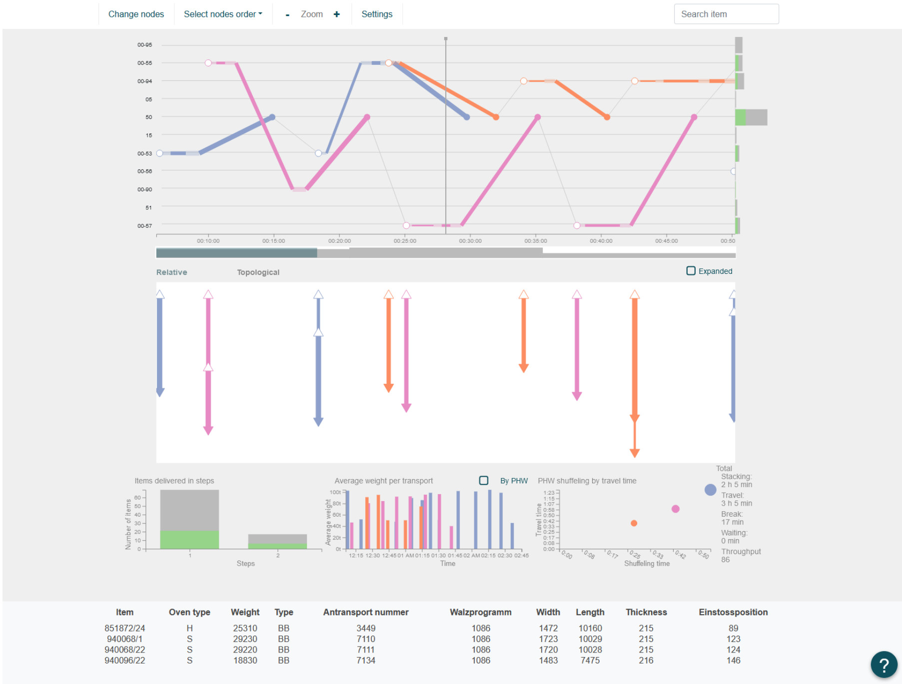


Figure 7: *High fidelity prototype*

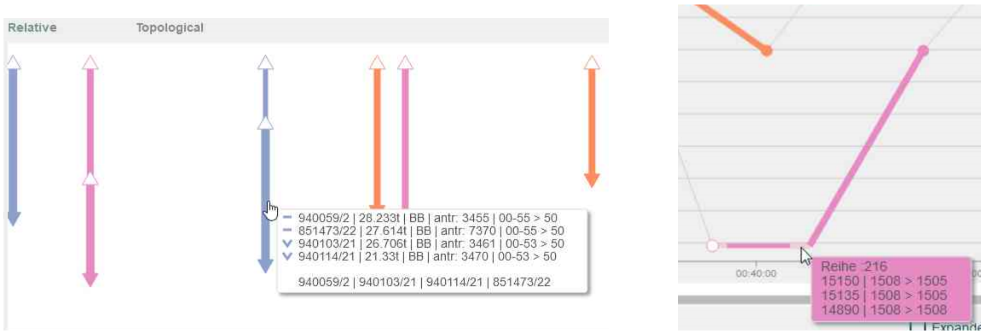
6.2 High fidelity prototype

Building on the merged paper prototype, we developed a web application that serves as the high-fidelity prototype of our design. As our goal was to prototype the application rapidly, we opted to create a single-page application using Typescript and Javascript, with support from the d3.js library. When the web application loads, it automatically loads the dataset from a specified folder, which can be updated in the source code. Any .csv file in that folder will be loaded as well. The application requires the data to be in a specific format similar to the output of the optimization algorithm but enhanced with data from the company's system. The tool's input is simply a file containing travel routes, so users can easily switch between viewing optimized and historical data.

Since we have already described the absolute, relative, and topological views earlier, we will not revisit them in this section. Instead, we will focus on introducing the new components of the web application and how they interact with the rest of the design.

As shown in Figure 7, the web app expands on the paper prototype by providing additional features. On the top, we have the settings and search bar. Next, we have the absolute and relative views, and displayed below are the statistical views. Lastly, at the bottom, we have a table displaying information about a list of items. The items in the table change when the user clicks on any transport or transport leg in the other views. Furthermore, the user can click on the bar charts, and the table displays information regarding the items in the selected bar.

We have implemented tooltips to enable a quick view of the information about items in the transport legs displayed in the relative and absolute view. A tooltip is displayed on hover, similar to the ones in Figure 8. In the first scenario (Figure 8a), the user hovers over the thicker section of the secondary blue transport line. Since the transport contains three triangles, the truck has visited three nodes. While at the first two nodes, the truck picked up items and then traveled to



(a) Tooltip 1

(b) Tooltip 2

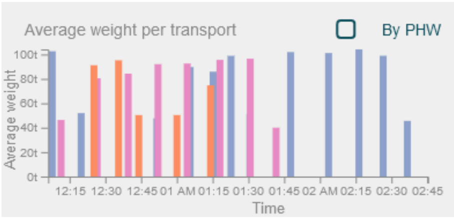
Figure 8

the last node, where it dropped the carried items. The tooltip displays four rows corresponding to the items carried and an additional row displaying the IDs of the items. Each item row is preceded by a symbol: an arrow up, an arrow down, or a line. This symbolizes the action the item has gone through in this transport leg. In the case at hand for the first two items, the symbol is a line, indicating that they were picked up at the start of this transport leg and dropped off at the end. The next two items have an arrow down, meaning they were picked up in a different transport leg but are dropped at the end of this one. An additional type of tooltip has been implemented to show the shuffling action and can be seen in Figure 8b.

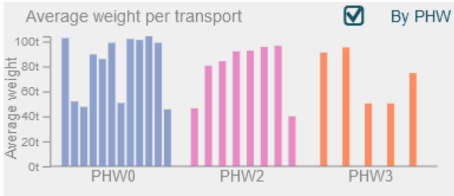
As previously noted, we have included a header that allows the user to filter and sort nodes, search items, and modify settings. The actions are explained below:

- A list of saved node orders is available under a dropdown and can be expanded by pressing the "Change nodes" button. This will open a dialog where the user can add his preferred order and filter out the unnecessary nodes.
- At the end of the header, an input form allows the user to search for items by ID. While inputting numbers, a list of filtered IDs is displayed. Clicking on one of the IDs will move the timeline to the appropriate location and highlight the pickup time and location of the item.
- A settings button opens a dialog allowing the user to change:
  - truck colors
  - the amount of information displayed in tooltips
  - the dataset
  - the language

Regarding the statistical visualizations, one noteworthy change is that we have removed the bar chart displaying the time to the destination of items, as we received feedback that it does not bring enough value to the user. In its place, we introduced an interactive bar chart having a timeline or a grouping of transports by truck on the x-axis if the "By PHW" checkbox is checked. The height of the bar displays the average weight moved per transport. This means a truck that is full on its first transport leg and half full on the second will display a bar chart of 75% height but in tons, not in percentages. This enables the user to identify transports where the truck did not work at full capacity. An example is shown in figure 9.



(a) Time on X axis



(b) Grouping by PHW on X axis

Figure 9

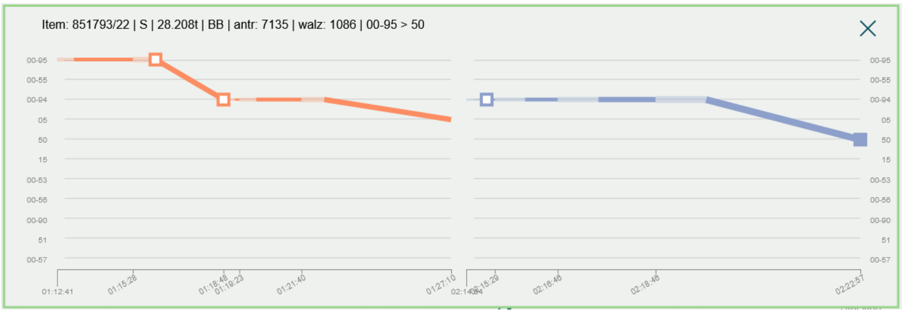


Figure 10: *Transshipment*

We initially intended to show transshipments as dotted lines in the relative and absolute view, but we quickly realized that they are of great importance and require the user's full attention. As such, we highlight the transshipment in the absolute view as a square, and when the user clicks on it, we display a dialog containing all transports on which the transshipped item is moved. As seen in figure 10, item 851793/22 is moved first with the orange truck and dropped at 00-94, from where the blue truck later picks it up. The transshipment view allows the user to quickly analyze the route the item has been transported on by having all the necessary details at a glance, including the other items on the transport (with the help of the popups and items table on hover/click over a line).

7 Formative evaluation

IronFlow-Viz has been developed and evaluated in a constant feedback loop, where the domain experts can express their ideas and problems with the tool. Throughout the following lines, we will present some of our testers' more important comments.

7.1 Item details - popups and table

One might think that the popups displayed contain too much information. Still, throughout our interviews with our domain experts, we received requests to fully inform the domain users of the items being transported. Such information, for example, would be the "Antrasportnummer", a domain-specific identification that encodes information regarding the item's creation. With a similar mindset, we duplicated the item details information in the bottom table. This enables the user

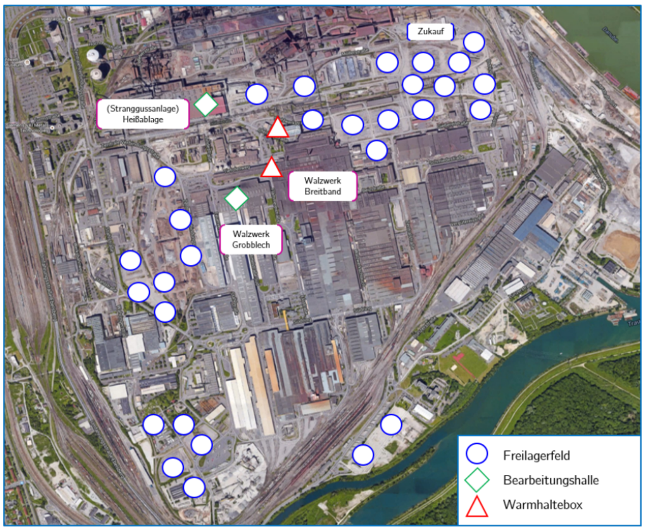


Figure 11: *Factory yard map*

to compare items with similar features, depending on where the user has clicked (e.g., Items transported in one step or items transported together). These changes allowed the planner to better understand the algorithm's output and how items are grouped into one transport.

7.2 Topological view

As stated before, users generally expect a map when visualizing the routes of a transport. This was no different with our domain users, but just because they were expecting it does not mean they needed it. In figure 11, you can see a map of the yard of the company of our domain experts. Nodes are represented by circles, triangles, or rhombuses based on their type. It would be difficult to represent the routes of multiple trucks on such a map because they would cross each other. Furthermore, encoding additional information on the route requires the map to be zoomed in to have a clear view of the lines. Just because it is difficult does not mean we should not have used a map, but it was not required as mentioned. More specifically, knowing the exact location of nodes and trucks is not required. Consequently, we introduced a topological view and encoded the information from the map similarly to a metro map. Lastly, the requirement of having a geographical view was of less importance than the time analysis solved by the relative view. We followed Schneiderman's Mantra [3] and only displayed such details on demand, meaning the user has to click a button to navigate to the topological view actively.

7.3 Transshipment view

In our initial design, we tried to integrate transshipments directly into the absolute and relative view, but during our feedback loops, we realized it was not a good idea. This was first because it was cluttering the views when using real domain data, and it was difficult to follow the two transport routes through time when the said transports were happening more than 1 hour apart. Secondly, we found out that it was important for the user to identify why an item was transshipped, and consequently, it required a view where the transshipment was the central focus. Consequently, we decided to highlight the transshipment in the absolute view by introducing a new symbol: a square. When hovering over the square, the transshipment view appears as a dialog, while a click keeps the view in place. This behavior allows the user to navigate through time in the absolute view while still having the transshipment in focus. Users stated the dialog was an important step in understanding transshipments, as it enabled them to follow the trip of an item from origin to destination.

7.4 Colors and settings

In response to user feedback, a new feature has been added to address concerns regarding the color scheme in the visualization tool. Some users expressed that the colors used in the tool were occasionally perceived as dull or less visually engaging. The feature consists of a dialog, which allows users to make customized color selections. Additionally, the dialog includes other useful features such as a language selector, dataset selector, and item details filter. The language selector lets users switch between different language options to suit their preferences or language proficiency. The dataset selector allows users to change between historical data and optimized planned routes without the need to restart the application. As mentioned previously, the item details filter changes the amount of data displayed in tooltips.

During our interviews, the users could not agree on a color scheme and had a diversified view of the tool's user experience in regard to settings. Implementing the dialog satisfied the users, as each individual could set their preferences.

7.5 Dialog for nodes order

Another difference in the preferences of individual users is that they are interested in different nodes. For example, some users are more interested in seeing data about facilities than of storage areas. Also, some storage areas are more important than others, and they also differ in size, meaning more items can be stored in a location, which leads to more transport in that area. Consequently, some users would like such nodes of interest at the top of the absolute view, while others prefer them in the middle. To fix this issue, we added predefined filtering and sorting in the header and allowed the user to open a dialog to set a list of nodes according to his preferences.

7.6 Researchers and planners feedback

While designing and developing IronFlow-Viz, we have contacted researchers biweekly and asked for their feedback. They expressed interest in identifying outlier transports and figuring out periods of low throughput. As such, they think that the statistical views helped them quickly find outlier transports and trucks, while the relative and absolute views helped them better analyze the periods where a low number of items were moved. Meetings with domain experts occurred at larger time intervals than with researchers. Planners mentioned that they had difficulties understanding the paper prototypes because of the synthetic data behind them. When the high-fidelity prototype was introduced, and real-world data was used, domain experts expressed enthusiasm about the tool and said, "everything makes sense now". When asked directly if they now trust the solution

Task	Question
T1 and T2	How many trucks are currently running? What can you say about item 4495? Describe pickup, destination and time. What type of item is it? Weight?
T3	Are items usually transported directly?
T4	Which is the longest transport? Which truck do you think travels the most? Can you say exactly? Are there any truck outliers?
T5	Which is the busiest node? How many trips are there between the facilities? How many items are delivered between “00-91” and “50”?
T6	Identify a transshipment and describe it.

Table 4: *Questions and instructions*

offered by the algorithm, they said yes, because they can now validate the result by analyzing the context in which the transport took place.

8 Summative evaluation

IronFlow-Viz incorporates many domain-specific terms and ideas, making the user interface look complex at first glance. Hence, we decided to do a usability test with people outside the transportation domain who thoroughly understand user-centered design. The testers selected have expertise in either user interface design or visual analysis. We also added two domain experts with whom we worked during development. Except for the two, it was important for the testers to have not seen IronFlow-Viz before. As such, we spent around 10-15 minutes presenting the visual analysis tool and then asked the testers to complete several tasks. Finally, the testers answered the SUS questionnaire.

The users were assigned a list of instructions and questions as shown in table 4. We have derived these questions from the tasks previously presented. Therefore, we have grouped the instructions by the task they represent. During the completion of the assignment, the testers were observed and helped if needed. Any hurdles were noted, and feedback was gathered.

System usability scale results

Table 5 shows the individual results of the SUS. Out of all the testers, we can see that only TS1 has a relatively low score, while the average of all the scores accounts for 75.83 out of 100. Considering that IronFlow-Viz is a specialized tool, we can say that it scores well on the system usability scale. According to Bangor et al., [4], a score above 70 demonstrates that the tool has good usability, while values below 60 recommend the tool be reanalyzed and fixed.

TS1	TS2	Company	TS4	TS5	TS6
57.5	65	72.5	80	90	90

Table 5: *SUS Results*

9 Conclusion and future work

We proposed a design for improving transport planners' trust in algorithm optimizations of route planning, incident investigation, and pattern discovery. We surveyed existing solutions and analyzed the tasks of transport planners and researchers. We presented possible solutions and explained how planners' trust can be achieved by introducing a visual analysis tool.

IronFlow-Viz could be further developed to handle dynamic data and show the current state compared to the planned routes. Another direction is allowing the user to modify the optimization algorithm's parameters so that the researcher can analyze the changes and adjust the results.

References

[1] Lex Alexander, Gehlenborg Nils, Strobelt Hendrik, Vuillemot Romain, and Pfister Hanspeter. Upset: Visualization of intersecting sets. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1983–1992, 2014.

[2] Schrijver Alexander. On the history of the transportation and maximum flow problems. *Mathematical Programming*, 91, 05 2004.

[3] Shneiderman B. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[4] Aaron Bangor, Philip T. Kortum, and James T. Miller. An empirical evaluation of the system usability scale. *International Journal of Human–Computer Interaction*, 24(6):574–594, 2008.

[5] Michele Berlingerio, Francesco Calabrese, Giusy Di Lorenzo, Rahul Nair, Fabio Pinelli, and Marco Luca Sbodio. Allaboard: A system for exploring urban mobility and optimizing public transport using cellphone data. *Advanced Information Systems Engineering*, page 663–666, 2013.

[6] Palomo Cesar, Guo Zhan, Silva Cláudio T., and Freire Juliana. Visually exploring transportation schedules. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):170–179, 2016.

[7] Tominski Christian, Schumann Heidrun, Andrienko Gennady, and Andrienko Natalia. Stacking-based visualization of trajectory attribute data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2565–2574, 2012.

[8] Tufte Edward. *Envisioning Information*, page 67. Graphics Press, 1990.

[9] Nivan Ferreira, Jorge Poco, Huy T. Vo, Juliana Freire, and Cl T. Silva. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2149 – 2158, December 2013.

[10] Danyel Fisher. *Beautiful Visualization by Julie Steele, Noah Iliinsky*, chapter Chapter 19. Animation for Visualization: Opportunities and Drawbacks. O'Reilly Media, Inc., 2010.

[11] Brooke John. Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189, 11 1995.

[12] Dongyu Liu, Di Weng, Yuhong Li, Jie Bao, Yu Zheng, Huamin Qu, and Yingcai Wu. Smartadp: Visual analytics of large-scale taxi trajectories for selecting billboard locations. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):1 – 10, January 2017.

[13] Wunderlich Marcel, Ballweg K., Fuchs Georg, and Landesberger Tatiana. Visualization of delay uncertainty and its impact on train trip planning: A design study. *Computer Graphics Forum*, 36:317–328, 06 2017.

[14] Kraak Menno-Jan. The space-time cube revisited from a geovisualization perspective. *Proc 21st Int Cartogr Conf*, 07 2008.

[15] Speranza M.Grazia. Trends in transportation and logistics. *European Journal of Operational Research*, 264, 08 2016.

[16] Friendly Michael and Wainer Howard. *A History of Data Visualization and Graphic Communication*, chapter Chapter 1: In the Beginning... Harvard University Press, 04 2019.

[17] Sedlmair Michael, Meyer Miriah, and Munzner Tamara. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, 2012.

[18] Maria Riveiro, Mikael Lebram, and Marcus Elmer. Anomaly detection for road traffic: A visual analytics framework. *IEEE Transactions on Intelligent Transportation Systems*, 18(8):2260 – 2270, August 2017.

[19] J. Sharko, G. Grinstein, and K.A. Marx. Vectorized radviz and its application to multiple cluster datasets. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1444–1427, 2008.

[20] Munzner Tamara. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, 2009.