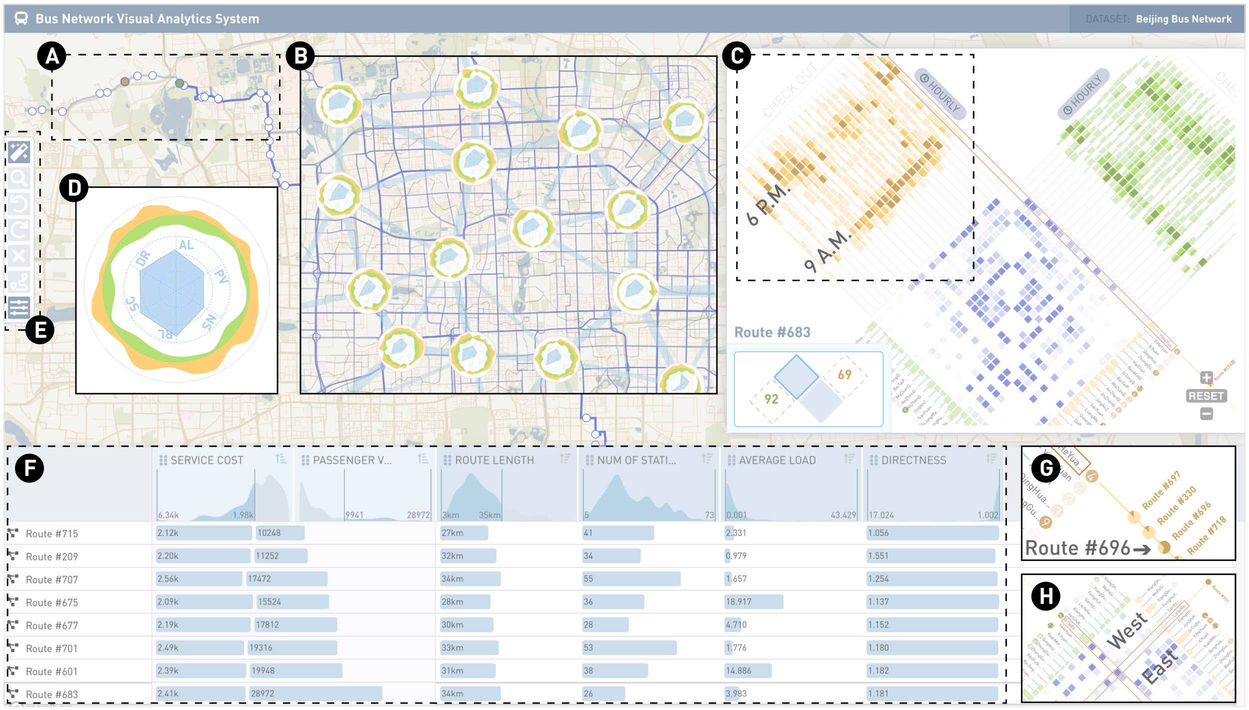
Towards Better Bus Networks: A Visual Analytics Approach

Di Weng, Chengbo Zheng, Zikun Deng, Mingze Ma, Jie Bao, Yu Zheng, Mingliang Xu, Yingcai Wu



	SERVICE COST	PASSENGER V...	ROUTE LENGTH	NUM OF STATI...	AVERAGE LOAD	DIRECTNESS
	6.34k 1.98k	9941 28972	3km 35km	5 73	0.001 43.429	17.024 1.002
Route #715	2.12k	10248	27km	41	2.331	1.056
Route #209	2.20k	11252	32km	34	0.979	1.551
Route #707	2.56k	17472	34km	55	1.657	1.254
Route #675	2.09k	15524	28km	36	18.917	1.137
Route #677	2.19k	17812	30km	28	4.710	1.152
Route #701	2.49k	19316	33km	53	1.776	1.180
Route #601	2.39k	19948	31km	38	14.886	1.182
Route #683	2.41k	28972	34km	26	3.983	1.181

Fig. 1. The interface of BNVA. A, C, G, H) The pattern indicated by the transfers from Route #683 to #696 suggests that Route #683 can be extended farther into the west. B) The network-level analysis, comprising the map, route, and aggregation layers, facilitates the exploration of network performance. D) The zone glyph summarizes the statistics and passenger flows of a transportation zone. E) The toolbox provides fine-grained controls for the model. F) The route ranking view depicts the performance criteria of the routes.

Abstract—Bus routes are typically updated every 3–5 years to meet constantly changing travel demands. However, identifying deficient bus routes and finding their optimal replacements remain challenging due to the difficulties in analyzing a complex bus network and the large solution space comprising alternative routes. Most of the automated approaches cannot produce satisfactory results in real-world settings without laborious inspection and evaluation of the candidates. The limitations observed in these approaches motivate us to collaborate with domain experts and propose a visual analytics solution for the performance analysis and incremental planning of bus routes based on an existing bus network. Developing such a solution involves three major challenges, namely, a) the in-depth analysis of complex bus route networks, b) the interactive generation of improved route candidates, and c) the effective evaluation of alternative bus routes. For challenge a, we employ an overview-to-detail approach by dividing the analysis of a complex bus network into three levels to facilitate the efficient identification of deficient routes. For challenge b, we improve a route generation model and interpret the performance of the generation with tailored visualizations. For challenge c, we incorporate a conflict resolution strategy in the progressive decision-making process to assist users in evaluating the alternative routes and finding the most optimal one. The proposed system is evaluated with two usage scenarios based on real-world data and received positive feedback from the experts.

Index Terms—Bus route planning, spatial decision-making, urban data visual analytics.

1 Introduction

In the last few centuries, the public transit bus service has become one of the most widely deployed public transportation backbones worldwide because of its flexibility and affordability. Bus routes are typically updated every 3–5 years because the travel demand of bus riders is constantly changing [44]. However, revising bus routes remains difficult because a) it is hard to analyze the performance of a huge and complex bus network to determine which routes are problematic, and b) a large solution space constituted by many factors involved in the planning process [61] needs to be extensively evaluated.

Most bus networks are planned and updated manually or by analyzing small datasets based on planners' knowledge [22, 47, 48]. Such methods can be laborious and time-consuming. To identify feasible bus routes, data-driven planning methods [30, 46, 55], including the

- *D. Weng, Z. Deng, and Y. Wu are with State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China and Zhejiang Lab, Hangzhou, China. E-mail: {dweng, zikun_rain, ycwu}@zju.edu.cn. Y. Wu is the corresponding author.*
- *C. Zheng and M. Ma are with Zhejiang Lab, Hangzhou, China. Email: chbozheng@gmail.com, mamzaug@foxmail.com.*
- *Jie Bao and Yu Zheng are with JD Intelligent Cities Research, Beijing, China and JD Intelligent Cities Business Unit, JD Digits, Beijing, China. E-mail: baojie@jd.com, msyuzheng@outlook.com.*
- *Mingliang Xu is with School of Information Engineering, Zhengzhou University, Zhengzhou, China and Henan Institute of Advanced Technology, Zhengzhou University, Zhengzhou, China. E-mail: iexumingliang@zzu.edu.cn.*

mathematical and heuristic ones, have been proposed to extract the routes based on predefined criteria. However, most of these methods act as a black box, generating an optimized result based on the given input data and parameters without explanations. Therefore, verifying the quality of the generated routes or adjusting the parameters to find an improved solution is difficult for domain experts. The interpretability of route generation has been further studied by Chen et al. [17] and Weng et al. [63]. Rather than producing a potentially unsatisfactory route, they propose to generate a set of *Pareto-optimal transit routes*, where none of the routes is better than any other route for all given criteria. However, experts are still required to manually compare among hundreds of routes and determine the most optimal one.

In this study, we collaborated with the experts from the transportation and urban computing domains to develop a solution that facilitates efficient analysis and incremental planning of bus routes. Inspired by the urban data analytics studies [40,64], we employ a visual analytics approach to integrate domain knowledge with the computational power of the Pareto-optimal route extraction method [63], thereby enabling the experts to interactively interpret the performance of complex bus networks and evaluate the alternative routes generated in real-time. Developing such a solution poses the following three challenges:

In-depth analysis of complex bus route networks. The performance of bus route networks must be extensively evaluated first to detect the ineffective parts that need to be replanned. However, bus route networks that operate in a large city may comprise a hierarchy of thousands of bus routes and stations, producing millions of trip records per month. Analyzing such a complex dataset for deficient routes poses challenges in the scalability and effectiveness of the proposed solution.

Interactive generation of improved route candidates. The Pareto-optimal route search model [63] can be used to generate potential route alternatives to the detected deficient routes. However, the constraints provided by the model are insufficient for many practical scenarios, for example, planning the routes via a few key stops. Moreover, since the model is progressive and time-consuming, the experts need to know when the results are sufficiently good to stop the search.

Effective evaluation of alternative bus routes. A large number of route alternatives will be generated with the model. These alternatives can be diverse and complicated. The experts request that they should be kept in the loop to make informed and transparent decisions. However, the volume, diversity, and complexity of the generated candidates prohibit the experts from effectively evaluating these candidates based on topologies or performance criteria to identify the most optimal one.

To address these challenges, we propose BNVA (Bus Network Visual Analytics system), a novel visual analytics system that helps bus route planners analyze and improve the performance of bus route networks. For the first challenge, we adopt a hierarchical exploration approach comprising network, route, and stop levels and propose a novel matrix view to facilitate the analysis of passenger flows and transfers; for the second challenge, we extend the model to support various constraints and integrate tailored visualizations to depict the performance of the generated routes in real-time; for the third challenge, we develop a progressive decision-making strategy based on route aggregation to help users evaluate the topologies and criteria of alternative routes. The contributions of this study are as follows:

- We characterize the user requirements in analyzing and improving the performance of bus route networks;
- We extend the Pareto-optimal route generation model to support additional constraints from practical scenarios;
- We develop a novel visual analytics system for bus route planning, featuring a route matrix view for passenger flow analysis and a conflict resolution strategy for progressive route decision-making;
- We evaluate our approach with two usage scenarios based on a real-world dataset and received positive feedback from experts.

2 Related Work

This section summarizes the transit network design and urban data visual analytics studies that are closely related to the present study.

2.1 Transit Network Design

Transit network design is a huge and complex research topic in transportation research. Guihaire et al. [30] divided the process of transit network design and scheduling into three parts: transit network design problem (TNDP), transit network frequencies setting problem, and transit network timetabling problem. This section mainly focuses on the related studies on the TNDP.

Many data-driven methods have been developed to generate transit networks automatically based on public travel demand, which can be determined from the survey [60,70], transportation [17], and telecommunication [56] data. These methods can be categorized into mathematical and heuristic methods. Mathematical methods [29,45,51] generate the transit line configuration with numeric optimization approaches, including mixed-integer linear programming [33]. Heuristic methods tackle the problem based on certain heuristics with the greedy approach [46], heuristic search, and genetic algorithms (GAs), such as tabu search [75], GA variants [16,68], simulated annealing [21], and ant colony algorithm [70]. However, most of these methods only generate an optimized transit line configuration balancing multiple performance criteria with the given parameters. To expand the decision-making scope with alternative solutions, Chen et al. [17] applied a random search approach in finding a set of transit routes that do not dominate each other in terms of two criteria, namely, route time and passenger flow, with the given origin and destination stops. Weng et al. [63] elaborated on the definition of Pareto-optimal transit routes and proposed an efficient route extraction method based on the Monte-Carlo tree search.

Nevertheless, the aforementioned automated methods may generate unsatisfactory results and typically require users to laboriously analyze the results and adjust the input parameters accordingly. Several preliminary efforts [39,53] in providing interactive solutions for the design procedure have been observed in the wild, but these solutions do not involve model-assisted visual planning of transit routes. To the best of our knowledge, this study is the first step toward the in-depth visual analytics of interactive bus route analysis and adjustment.

2.2 Visual Analytics of Urban Data

Large-scale urban data, such as human mobility, social network, geographical, and environmental data, have been collected across cities in recent years with the advancement in data sensing and management technologies. These urban data have made efficient data-driven solutions [78] possible for various urban problems, such as bike lane planning [12], ambulance location selection [38], and travel time estimation [62]. However, most of these methods cannot automatically produce satisfactory results [40] without domain experts in the loop because of the complicated nature of urban problems.

To facilitate the analysis of urban data, many studies [79] have employed visual analytics that enables users to interactively obtain patterns and insights from complex datasets, typically with the aid of efficient computational models. These studies mainly involve the visualization of time [1] and location properties. Time properties can be depicted with axis-based design [76], relative time encoding [65], and dynamic visual representations [25]. The visualization of location properties can be categorized into the point-, region-, and line-based techniques [79]. Point-based techniques [7,23,67] represent the locations with the points in their spatial contexts. Region-based techniques [2,72] render the aggregated location data based on certain spatial divisions. Line-based techniques [9,41] encode the locations based on road or traffic networks with line-based representations. Besides, other properties involved in the urban visual analytics studies, such as numbers and texts, can be presented with their corresponding visualization techniques [37,50,69]. Furthermore, various techniques, including spatiotemporal [26,59] and multivariate [35,43] visualizations, have been developed for the visualization of multiple properties combined.

With the aforementioned visualization techniques, urban visual analytics studies have explored various data sources, such as human mobility [6,40,52], social media [15,42], environmental [19,27,66], and transportation data [32,54,71]. Andrienko et al. [4] comprehensively summarized the prior studies on transportation data and categorized these studies into three directions, namely, transportation infrastructure [5, 8], transportation user behaviors [10, 13], and transportation modeling and planning [3, 57]. Transportation data can be also used in studying place connectedness combined with novel visual analytics techniques [11]. However, whether visual analytics can facilitate the efficient planning of bus transit routes remains to be elucidated. In this study, we tackle this problem by combining the visualization techniques for the time and line-based location properties to provide an overview of bus network performance and enable users to interactively evaluate alternative bus routes.

3 Background

In this section, we show the formulation of the bus route planning problem and summarize the identified domain requirements.

3.1 Problem Formulation

In the past year, we collaborated with four domain experts (EA-ED) working on a bus network improvement project for a large city with over 7 million residents. EA and EB are urban computing experts who have been studying sophisticated data-driven solutions for urban problems in the last few decades. EC is a data scientist in charge of developing a route optimization model for the bus network improvement project, and ED is the project manager of this project. The goal is to find an improved bus route network that meets the travel demand of citizens.

An extensive literature review [30] reveals that most data-driven methods focus on creating a new optimized route network from scratch. However, such methods are impractical in our scenario because re-planning the entire network will significantly disrupt commuters' daily routines. By working with the domain experts, we constructed a three-stage iterative workflow for analyzing and improving bus networks:

Stage Exploration: At this stage, the experts aim to explore the performance of the bus network and determine ineffective routes. The ineffective routes may exhibit several characteristics, such as *low passenger flows* (few passengers will take these routes, leading to the waste of public transportation resources), *high service cost* (these routes are costly to maintain mainly due to long service time and frequent scheduling), and *poor route directness* (the transit distances on the buses are considerably longer than the road distances). The analysis of the existing bus network is based on the historical bus trip records.

Stage Manipulation: After an ineffective route is determined, the experts would like to obtain its replacement routes. The Pareto-optimal route search method [63] is selected because of its capabilities in generating alternative routes on the approximated Pareto front based on multiple performance criteria. However, the original method is progressive without indicating the quality of the current results in real-time. Therefore, this stage aims to allow users to intuitively specify route generation constraints and determine when the generated routes are sufficiently good to terminate the search process.

Stage Evaluation: Given a set of Pareto-optimal route alternatives, the experts aim to determine the improvement of the generated routes, make trade-offs between the performance criteria, and determine which alternative route is the most optimal. Considering that each alternative route is a graph that shares some nodes with other routes, analyzing the similarities and differences between these routes with hundreds of the routes generated becomes increasingly challenging. Hence, this stage aims to facilitate intuitive multi-criteria comparison and decision-making of alternative routes with tailored visualizations.

3.2 Data Description

The proposed system is based on three types of data, namely, bus stop, route, and trip data, collected from bus networks.

- **Bus stop data** comprise the bus stops in a city. Each stop is defined by its ID, name, and coordinates.
- **Bus route data** comprise the bus routes, where each of them is identified by its ID and a stop sequence.
- **Bus trip data** contain a series of bus fare card records, where each of them comprises a card ID, a tap-on timestamp, and the route and stops where the fare card was tapped on and off. However, the tap-off timestamps are not present in the dataset because of sensor errors. This timestamp was inferred either by using the tap-on timestamps at the destination stops if such transfer records exist, or based on driving time along the bus route at 20 km/h plus 2 min spent at every stop, as suggested by our domain experts.

3.3 Requirement Analysis

To develop a feasible and practical approach for analyzing and improving bus networks with visual analytics, we followed the nested model for visualization design and validation [49] to characterize the domain problem and identify the low-level analytical requirements. Based on the three-stage workflow, we conducted literature reviews on the related studies, informal interviews with the experts, and brainstorming sessions with other visualization researchers to iteratively refine the workflow and compile the following analytical requirements.

Stage Exploration (P).

P1: Obtain the spatial overview of the bus network and its performance. The experts requested to see a map-based overview of the bus network similar to other GIS software [20, 53]. Such an overview should help users to rapidly orient themselves in the spatial context and grasp the spatial distribution of routes. The overview should also include the visualization of the network performance to guide users in performing a drill-down analysis on the potentially ineffective routes.

P2: Analyze the passenger flows of bus routes to find weaknesses. The movement of passengers through the bus network provides key insights for the experts to evaluate the performance of the routes. For example, some parts of a route might be non-functional if few passengers were getting on or off in these parts. Therefore, intuitive visualization of passenger flows is highly demanded to facilitate the identification of deficient routes. Moreover, such a visualization should also enable the experts to analyze the transfers among multiple bus routes, which may reveal patterns to improve route design.

Stage Manipulation (M).

M1: Generate a set of alternative routes based on the constraints. The experts prefer to interact with the model and generate the Pareto-optimal routes as the potential replacements of the selected ineffective route. To minimize the disruption caused by route changes, the experts may preserve several primary stops from the original route. Moreover, certain constraints, such as the construction cost and maximum route length, may be imposed on route generation. An easy-to-use interface is required for translating these constraints into the complex parameters required by the model.

M2: Inspect the quality of the generated routes in real time. Considering that the model is progressive and does not stop automatically, the experts need to know when the results are sufficiently good to stop the route generation process. Hence, tailored visualizations are required to depict the current status of the generation process in real-time and provide the early quality preview of the alternative routes as the process continues. Moreover, such visualizations should allow some undesired routes to be removed from the search space to interactively guide and accelerate the search process.

Stage Evaluation (L).

L1: Compare the generated routes based on topologies. To help the experts identify the most promising ones from the generated routes, the proposed system must reveal the topological similarities and differences among these routes. The experts may want to know: *What stops do these two routes share? Which pair of consecutive stops is the most frequently selected? How much does a route deviate from another one by taking a detour?* Integrating topological information can help experts estimate the performance of these routes and eliminate the undesired ones that share similar characteristics.

L2: Compare the generated routes based on multiple criteria. The most promising alternative route should also be determined in terms of the performance criteria. However, experts may treat each criterion differently under different circumstances. For example, the distances between the stops in a suburban bus route will be considerably longer than those between the stops in a city bus route. To facilitate a judicious decision-making process, the system should enable the experts to inspect the criteria of the generated routes and identify the most optimal

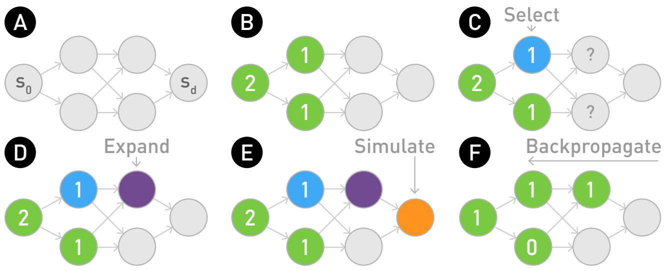


Fig. 2. A) A station graph generated between stations s_0 and s_d. B) An example of the state tree (green nodes represent the explored states, and the numbers of the discovered Pareto-optimal routes are shown in the nodes). C) The most promising state (blue) is selected at the selection stage. D) The most promising neighboring station (purple) is expanded at the expansion stage. E) The simulation begins from the expanded station to s_d. The involved stations are marked in purple. F) The route obtained from the simulation is tested against the current set of Pareto-optimal routes, and the numbers in the nodes are updated.

one efficiently with tailored ranking models.

4 Model

In this section, we introduce the route generation method that supports the visual analytics of bus route planning and propose the modification and improvements to tailor this method for the interactive analysis.

4.1 Pareto-Optimal Route Search

This subsection summarizes the core idea of Weng et al.'s method [63] that searches Pareto-optimal transit routes based on the Monte-Carlo search tree. The Monte-Carlo search tree [14] is studied to search the best next move in a game. Starting from a given game state, the search repeats four stages, namely, selection, expansion, simulation, and backpropagation. First, the most promising state is selected. Then, a new state is created based on the estimated best next move of the selected state. Next, the game result is obtained via simulation. Finally, the estimated value of the states in the tree is updated accordingly.

Given an origin and a destination station s_o and s_d in a set of stations $S = \{s_1, s_2, \ldots\}$, a feasible transit route between the origin and destination stations is denoted as $R = \{r_1, r_2, \ldots, r_n\} \in P_{od}$, where $r_1 = o$, $r_n = d$, $s_{r_i} \in S$, and P_{od} comprises all routes between s_o and s_d. A set of criteria $C = \{c_1, c_2, \ldots\}$ can be defined to measure the performance of a route R. We say R_i *dominates* R_j if a) R_i is better than or equal to R_j with respect to every criterion $c \in C$ and b) R_i is strictly better than R_j with respect to at least one criterion. The goal is to find an optimal Pareto-set of feasible transit routes $P = \{R_1, R_2, \ldots\} \subseteq P_{od}$, where R_i is *not* dominated by R_j for any $R_i, R_j \in P$.

Initially, a directed acyclic station graph $G_{od} = (S, E_{od})$ (Fig. 2A) is constructed between the origin and destination stations s_o and s_d with the approach proposed by Chen et al. [17], where the edges E_{od} among the station nodes S are given based on heuristic criteria. A search strategy is then proposed based on the following five stages.

Initialization. A state tree is initialized with a root node representing the origin station. A set of discovered Pareto-optimal transit routes P_s is maintained throughout the search process. Fig. 2B shows an example state tree. The numbers in the nodes indicate the numbers of the discovered Pareto-optimal transit routes involving the nodes.

Selection. This stage aims to find the most promising node m_p. Starting from the root node, the search process recursively descends in the state tree by selecting a child node with the maximum upper confidence bound [36], balancing between the exploration of the less-visited and high-valued nodes. The selection stops if any neighboring station of the current node has not been added to the state tree. The selected node m_p is marked in blue in Fig. 2C.

Expansion. This stage seeks to identify the most promising neighboring station of m_p that is not in the state tree. All routes belonging to a node in the state tree begin with a *route prefix* $R^k = \{r_1, \ldots, r_k\}$ and constitute a *route subspace* $P_{od}(R^k)$. The search process estimates the gain Δc_i with respect to every criterion $c_i \in C$ in choosing a neighboring station s_{e_i} and its associated route subspace $P_{od}(R^k \cup \{e_i\})$ based on heuristic approaches. The details of these approaches can be found in the original paper. Thereafter, the gains are normalized and averaged for each neighboring station, and a neighboring station is randomly chosen based on the distribution of the gains. This station, marked in purple in Fig. 2D, is inserted into the state tree as a child node of m_p.

Simulation. Starting from the newly inserted node, the search process recursively performs a similar selection strategy to that in the expansion stage until the destination station is reached (the simulation nodes are marked in orange in Fig. 2E). Then, the obtained route is tested against the Pareto-optimal transit route set P_s.

Backpropagation (Fig. 2F). If the obtained route is Pareto-optimal, P_s will be updated to remove the routes that are no longer Pareto-optimal. The search process returns to the selection stage or terminates when the time limit is exceeded or the station graph is exhausted.

4.2 Model Optimization

This subsection introduces the optimization and improvements we have implemented for the Pareto-optimal route search model to make it suitable for the interactive analysis.

4.2.1 Efficient Computation of Route Directness

Route directness [74], which measures how much the transit network distance between each pair of stations differs from the corresponding shortest road network distance, is a crucial factor to consider in the bus route planning process. High route directness typically indicates that buses take minimal detours, and passengers will spend minimal time in traveling on this route. In the expansion stage, Weng et al. [63] proposed an averaging heuristic that estimates the route directness I_R of a route subspace $P_{od}(R^k)$ constituted by all routes starting with prefix $R^k = \{r_0, r_1, \ldots, r_k\}$. However, computing such a heuristic can be time-consuming because the time complexities are estimated to be $O(|S|^2)$ for the expansion stage and $O(|S|^3)$ for the simulation stage. To accelerate the computation, we compute I_R with the following formula:

$$I_R(P_{od}(R^k)) = \frac{1}{N_{r_k d}}((\sum_{u \in R^k} A(u, r_k)) + B(r_k)),$$

where $A(p,q) = \sum_{v=\eta(s_q)}^{\eta(s_d)} \frac{\delta_{pv}}{D_{pv}}$, and $B(q) = \sum_{v=\eta(s_q)}^{\eta(s_d)} \sum_{u=\eta(s_q)+1}^{v-1} \frac{\delta_{uv}}{D_{uv}}$. $\eta(s_i)$ denotes the topologically sorted index of station s_i in the station graph, N_{uv} is the number of possible paths between stations s_u and s_v, and δ_{uv} and D_{uv} are the transit network and shortest road distances, respectively, between stations s_u and s_v. $A(p,q)$ and $B(q)$ can be efficiently precomputed for every $s_p, s_q \in S$ because they are unrelated to route prefix R^k, resulting in low time complexities $O(|S|)$ and $O(|S|^2)$ at the expansion and simulation stages, respectively.

4.2.2 Translation of Generation Constraints

To effectively integrate the route generation model with the proposed system, the experts requested us to implement practical route constraints, including multi-stop anchoring, construction cost, and service cost, in addition to route service time, demand satisfaction, and route directness defined in the original model.

Multi-stop anchoring. The model only allows users to define the origin and destination stops. However, experts prefer to anchor several primary stops of the original route to minimize the impact on route network structure and riders' daily routines caused by route changes. Moreover, such functionality is particularly useful when only a part of a route needs to be adjusted. To achieve multi-stop anchoring, we reformulate the input of the model as a set of stop sets $\mathbb{S} = \{S_1 = \{s_{11}, s_{12}, \ldots\}, S_2, \ldots\}$ and construct a station graph by combining the subgraphs generated in the following procedure. For each stop set $S_i \in \mathbb{S}$, the stops in this set will be sequentially connected as a station subgraph; and for each pair of consecutive stop sets $S_i, S_{i+1} \in \mathbb{S}$, a station subgraph will be generated between the last stop in S_i and the

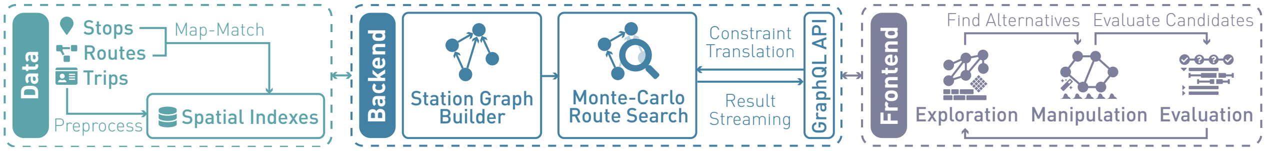


Fig. 3. The system architecture of BNVA. BNVA comprises three parts, including data storage, backend, and frontend. Data storage preprocesses the data and indexes the data with a spatial database. Backend builds station graphs and answers route generation requests. Frontend combines the exploration, manipulation, and evaluation interfaces to establish an iterative workflow for analyzing and improving the performance of bus networks.

first stop in S_{i+1}. After the full station graph is built, we test each node in this graph against the criteria proposed by Chen et al. [17] and remove the nodes that neither belong to any stop set in $\mathbb{S}$ nor satisfy the criteria. Therefore, the Pareto-optimal routes generated on this graph contain all the stops in their order given in $\mathbb{S}$.

Construction cost. Integrating the construction cost enables experts to specify the cost of establishing a bus route as the estimated cost per stop C_s times the number of stops in this route. We propose to add a new construction cost criterion C_R that measures the average construction cost of a route subspace $P_{od}(R^k)$ with route prefix $R^k = \{r_1, ..., r_k\}$. The number of possible routes from a stop s_i to destination stop s_d, denoted by n_i can be computed by traversing in the station graph from s_i to s_d. Iterating from s_d back to s_{r_k}, we have: $c_i = \frac{1}{n_i} \sum_{j \in E_i} n_j (c_j + C_s)$, where E_i comprises the indices of the neighboring stops of s_i. Hence, we have $C_R(P_{od}(R^k)) = c_{r_k}$. The computation of this formula is efficient because the time complexity is estimated to be $O(|S|)$.

Service cost. Service cost indicates the cost of scheduling and operating a bus route. The estimation of service cost is based on several factors including headway H (the time interval between two consecutive buses), service span T_s (the period when the route is operating), one-way route service time T_R, crew wage per hour C_w, fuel cost per kilometer C_f, and maintenance cost per kilometer C_b. We choose $v = 20$km/h to be the average speed of buses, as recommended by the experts. Service cost C_v can be derived and estimated from the route service time criterion T_R: $C_v = \frac{T_s}{H}(2T_R C_w + 2T_R v(C_b + C_f))$.

4.2.3 Parallelized Pareto-Optimal Route Search

We accelerate the route generation model by exploiting multiple cores on modern workstations and parallelizing the search process.

In the expansion stage, rather than choosing a neighboring station randomly based on gain distributions, we sort the neighboring stations by their average gains and select the largest-k stations, where k is the maximum number of threads to use specified by the user. Multiple simulation stages, which still use the original random strategy, are spawned in parallel for these stations. The generated candidate routes are then collected and tested against the Pareto-optimal transit route set in batch at the backpropagation stage.

4.2.4 Model Interactivity

We propose the following modifications to improve the interactivity and responsiveness of the aforementioned model and provide better integration with the visual analytics interface.

Interactive search. Based on the current route generation result, the users may find some stations unnecessarily included in the routes and prefer to manipulate the search space by removing such stations from the station graph. Moreover, the model should support adding stations that are not included in the station graph. The model processes such requests by first regenerating the station graph incrementally and then altering the state tree and the route set to reflect the changes.

Route pruning. The route generation model may produce a sheer volume of Pareto-optimal routes because of the increasing number of performance criteria. However, many routes can be unsatisfactory because they may perform well in some criteria but extremely poor in other ones. To eliminate these routes, we allow users to specify the desired value range for each criterion. As the search process computes the estimation heuristics (e.g., route directness $I_R(P_{od}(R^k))$), we use a similar heuristic strategy to estimate the minimum and maximum criterion values for each neighboring station. A station will not be chosen if its estimated value range does not overlap with the specified one. Such a pruning strategy not only integrates criterion filters into the model but also improves the efficiency of route generation.

5 System Architecture

BNVA is a web-based visual analytics application constituted by three parts, namely, data storage, backend, and frontend (Fig. 3). The data storage preprocesses data sets and indexes them spatially in the database. The backend handles route generation requests and exposes the internal states and APIs of the generation model to the frontend. The frontend comprises three visual interfaces, namely, the exploration, manipulation, and evaluation interfaces. The exploration interface facilitates the performance analysis of the existing bus network, the manipulation interface enables users to interact with the progressive model, and the evaluation interface assists users in comparing among the candidate routes based on topologies and performance criteria.

6 Visual Design

To facilitate efficient exploration and improvement of bus route networks, BNVA comprises three visual interfaces, namely, the exploration (P1, P2), manipulation (M1, M2), and evaluation (L1, L2) interfaces. The visual information-seeking mantra [58] motivates us to develop an exploration interface that empowers the users to discover deficient routes in the network from the overview with a map (P1) to the details with novel flow matrices depicting passenger flows and transfers (P2). The alternative routes can be obtained via the manipulation interface (M1), which provides tailored visualizations of route performance to informed the users of the generation progress and the quality of route replacements in real-time (M2). The evaluation interface adopts a *conflict resolution* strategy that iteratively extracts the topological differences and route clusters from the generated routes to facilitate the progressive and reliable decision-making process in finding optimal routes based on topologies (L1) and performance criteria (L2).

6.1 Exploration Interface

A large-scale bus network typically comprises a complex hierarchy constituted by massive routes, stops, and trips. To facilitate the identification of deficient routes, we organize the interface by the network-, route-, and stop-level analyses. For the network-level analysis, a spatial aggregation view is designed to provide a spatial overview of the entire network and help the users filter routes with spatial constraints (P1, P2). For the route-level analysis, a route ranking view is implemented to depict the performance of the routes (P1), allowing the users to find inefficient routes based on performance criteria. For the stop-level analysis, a route matrix view is proposed to visualize the passenger flows and transfers among the stops in a selected route with matrices, establishing fine-grained inspection of route performance (P2).

6.1.1 Network-Level Analysis

An overview of the bus network is essential in locating the areas where the inefficient routes most likely exist. The spatial aggregation view (Fig. 1B), designed for the network-level analysis, comprises three linked layers, namely, the map, route, and aggregation layers, to help users analyze the network on a broad scale.

The map layer comprises a base map. The route layer draws all routes on the map in blue with opacity. However, the route layer cannot

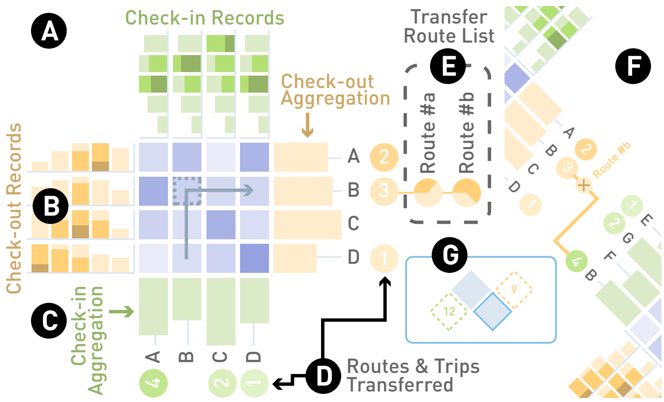


Fig. 4. The design of flow matrices. A) Flow matrices visualize the flows and transfers of passengers on the selected routes. B) The horizon charts show the temporal distribution of check-in/out records. C) The bar charts aggregate the total passenger flows. D) The numbers of transfer trips and routes are encoded with the opacity of the circles and the numbers on them. E) Clicking on the circles brings up a list of transfer routes. F) A new flow matrix for the transfer route will be aligned and linked with the original one. G) The overview of all flow matrices.

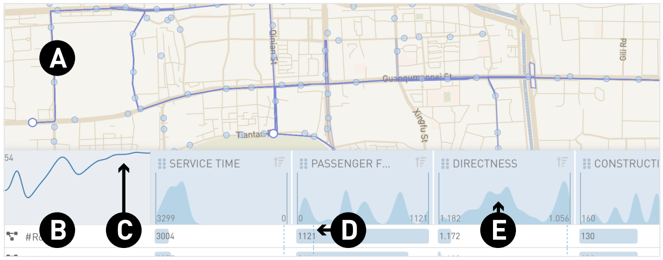


Fig. 5. The illustration of the manipulation interface. A) The feasible stops and the generated routes will be displayed on the map. B) The criteria of the routes are depicted in the table. C) An animate line chart shows the number of generated routes. D) The vertical dashed lines indicate the criterion values of the original route. E) The criterion distributions of the obtained routes are shown in the column headers.

depict the network topology because of the overlapping routes. Therefore, we designed the aggregation layer to visualize the topology with an *aggregation graph*. Each node in the aggregation graph corresponds to a group of bus stops aggregated spatially with hierarchical clustering [34] by balancing the number of stops in each group. These groups divide the city into *transportation zones*. The boundaries of these zones are computed with a Voronoi diagram [24] by unifying the polygons that enclose the bus stops inside the zones. In addition, the numbers of the routes between the zones are encoded with the link widths.

A *zone glyph* (Fig. 1D) is placed at the centroid of each transportation zone to summarize the key statistics of this zone. The radar chart at the glyph center encodes six averaged criteria, namely, route length (RL), number of stops (NS), passenger volume (PV), average load (AL), route directness (DR), and service cost (SC). Users can configure the visibility and order of the dimensions flexibly in the context menu. Two diverging circular distributions around the glyph visualize the amount of passenger flows by the geographical directions in which the passengers in this zone leave (green) or enter (orange). Double clicks magnify the glyphs, allowing users to obtain a clearer view of the radar charts inside. The design of this glyph is kept simple yet informative, such that the users can naturally obtain and compare the performance of different zones with a number of glyphs.

6.1.2 Route-Level Analysis

In the route-level analysis, we focus on assisting the users in finding inefficient routes based on performance criteria in complementary to the spatial information. Inspired by LineUp [28], a table-based ranking visualization is included in the route ranking view to facilitate the multi-criteria analysis of the routes. As illustrated in Fig. 1F, the columns represent six criteria similar to the zone glyphs, and each row is a route that can be ranked by selecting any column. The criteria can also be grouped and sorted with different weights. In addition, the criterion distributions are shown in the column headers, providing an overview and range filters for the routes in the table.

6.1.3 Stop-Level Analysis

The stop-level analysis enables the users to explore and evaluate the passenger flows and transfers among the stops in a selected route.

A *flow matrix* (Fig. 4A) is designed to visualize the passenger flows and transfers. The columns and rows of the matrix correspond to the bus stops, and the color intensity of each cell in the matrix encodes the number of passengers traveling between the stops. The color intensity is computed by normalizing the number of passengers into $[0,1]$ against a global passenger flow threshold, which can be changed by users. The horizon charts (Fig. 4B) visualize the number of passengers, aggregated by time, checking in or out at the corresponding stops. The visibility of the horizon charts can be toggled in the context menu to simplify the view. Three-band horizon charts are chosen over bar charts because the estimation accuracy of horizon charts is better than that of bar charts when chart height is limited [31]. In addition, the bars (Fig. 4C) encoding the number of passengers per stop are positioned to the bottom and right of the matrix. In case of long bus routes, users can right click on the matrix to select stops they wish to keep in the view.

Massive transfers may indicate that the routes are not well planned and discourage the use of bus transportation [73]. To visualize transfer information, we encode the number of passengers transferred to or from other routes with the opacity of the circles (Fig. 4D) next to the station names. The numbers on the circles indicate how many routes passengers have transferred to or from. The minimum opacity of the circles has been tuned to maintain the visibility of the numbers inside them. Clicking on a circle expands a list of the associated routes (Fig. 4E), each preceded by a small pie chart indicating the percentage of passenger flows transferred to or from the route. Selecting a route in the list reveals another flow matrix visualizing this route, which is aligned and linked to the original matrix (Fig. 4F). Inspired by MatrixWave [77], we rotate the matrices by 45° clockwise to accommodate them linearly for enhanced scalability. An overview of the matrices (Fig. 4G) is provided at the bottom-left of the view, where each matrix is represented with a square. The currently focused matrix is outlined in the overview, and the numbers of transferred routes are enclosed in the dashed squares.

6.2 Manipulation Interface

After a deficient route has been determined with the exploration interface, the users can obtain replacement routes from the manipulation interface. This interface allows the users to control the model by specifying model parameters, criterion filters, and anchored stops (M1). The model results will be streamed and visualized in real-time to help the users determine the quality of the generated routes (M2).

The toolbox at the left of the system (Fig. 1E) provides fine-grained model controls, including starting/pausing the optimization of the selected route, navigating to the previous/next result set, exiting the manipulation interface, toggling the visibility of the original route, and configuring the parameters for the generation process. After the generation starts, feasible stops will be detected and shown on the map as blue circles (Fig. 5A). The produced routes are drawn in blue with opacity. The users can anchor the stops with clicks or remove the stops with right clicks. New stops can also be added by clicking on the map.

The generated routes will be displayed in the route ranking view (Fig. 5B) in real-time, allowing the users to evaluate these routes with ranking and filtering. An animated line plot in the left top of the table (Fig. 5C) shows the number of generated routes. The vertical dashed lines in the table (Fig. 5D) indicate the criterion values of the original route. The criterion distributions in the column headers (Fig. 5E) not only allow the users to specify the ranges of criteria but also deliver a quality overview of the generated routes where the users can judiciously

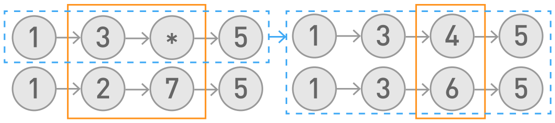


Fig. 6. An example of conflicts. The conflicted parts are marked with orange boxes. After aggregation, users have two conflicted choices on the left: `1-3-*-5` and `1-2-7-5`. The former choice leads to another two conflicted choices on the right, `1-3-4-5` and `1-3-6-5`.

Algorithm 1 The route clustering algorithm attempts to extract no more than β clusters from the given routes based on their similarities.

Input: A set of routes R, a set of criterion functions C and their weights w, and the desired number of choices β.
Output: A set of route clusters G.

```
1: procedure ClusterRoutes($R, C, w, \beta$)
2:     $G \leftarrow R$
3:     while $|G| > \beta$ do
4:         $P \leftarrow \arg\max_{(g_u, g_v) \in G^2} |g_u \cap g_v|$
5:         $G' \leftarrow \{g_u \cap g_v | (g_u, g_v) \in P \wedge \{g_i \in G | g_u \cap g_v \nsubseteq g_i\} \neq \varnothing\}$
6:         if $G' = \varnothing$ break
7:         $g_m \leftarrow \arg\min_{g \in G'} \text{StdDev}(\{\sum_{c_k \in C} w_{c_k} c_k(r_i) | r_i \in \{r_j \in R | g \subseteq r_j\}\})$
8:         $G \leftarrow \{g \in G | g_m \nsubseteq g\} \cup \{g_m\}$
9:     return $G$
```

determine the termination of the generation process.

6.3 Evaluation Interface

To help the users evaluate hundreds of alternative routes and identify the most optimal one, we propose an interactive *conflict resolution* strategy to facilitate efficient and reliable progressive decision-making among these routes. The strategy is twofold. First, the topological similarities and differences among the routes are determined with a route clustering method, and the differences between the routes when they go through different streets are grouped into *conflicts*. After the conflicts have been extracted, the users can interactively *resolve* these conflicts by iteratively choosing a cluster of routes that share a similar topology and eventually arrive at the most optimal route. To aid the users in such progressive decision-making processes, the routes and their topologies are depicted with *conflict markers* on the map (L1), and the criteria of the available choices are visualized in the ranking view to facilitate the analysis of route performance (L2).

6.3.1 Detecting Conflicts

The concept of conflicts assists the users in resolving the topological differences between the routes by comparing and selecting the candidate routes they prefer. However, excessive routes may lead to choice overloading [18] in such a complex decision-making scenario, prohibiting the users from identifying optimal routes. Therefore, we propose to group the routes into several clusters first and then detect the conflicts among the route clusters, such that the choices to resolve each conflict will be no more than these clusters. For example, three four-stop routes, `1-3-4-5`, `1-3-6-5`, and `1-2-7-5`, can be grouped into two clusters matching the route patterns `1-3-*-5` and `1-2-7-5`, respectively, when the number of choices is limited to two. Hence, a conflict occurs at the second and third stops as shown in Fig. 6. If the users choose to resolve this conflict with `3-*`, another conflict will be detected on the third stop; otherwise, the optimal route `1-2-7-5` is decided.

The route clustering algorithm is detailed in Alg. 1. Initially, we consider each route as a route cluster comprising only one route. First, we find the pairs of route clusters that share the most stops (cf. line 4). Then, for each pair of clusters, we compute the standard deviation of weighted criterion sums for the routes in a new cluster obtained by merging clusters whose topologies are similar to the common part of these two clusters (cf. line 7). Subsequently, the pair with the minimum standard deviation will be merged (cf. line 8) if the merge does not reduce the number of choices to one (cf. line 5). The algorithm repeats this procedure until such a merge is impossible (cf. line 6) or the number of clusters is under the given threshold β (cf. line 3).

After the routes have been clustered, the conflicts can be detected by aligning all clusters in G by stop and determining the consecutive stops that are not shared among the clusters.

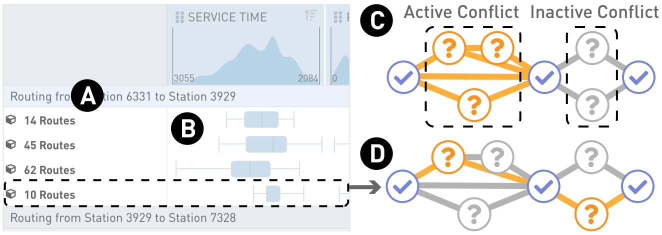


Fig. 7. The illustration of the evaluation interface. A) Each collapsible group in the ranking view represents a detected conflict, and each row in the groups is a route cluster. B) The criterion distributions of a route cluster are visualized with boxplots. C) The topology of the route clusters is depicted with a node-link diagram, where the state of each shared stop is indicated with a conflict marker. D) Each row in the table corresponds to a path from the origin stop to the destination stop.

6.3.2 Resolving Conflicts

Each detected conflict is shown in the ranking view as a collapsible group (Fig. 7A). Only one conflict is active for resolving at a time. The users can toggle between the conflicts by clicking on the groups. The rows in each group represent route clusters, where each of them corresponds to a path from the origin stop to the destination stop, as illustrated in Fig. 7D. The users can hover on a row to see the routes in the cluster on the map and click on a row to resolve the conflict with the cluster. When a route cluster has more than one route, its criterion distributions are visualized with boxplots (Fig. 7B); otherwise, the cluster is displayed similar to a normal route, showing criteria with bars to suggest that this cluster will be the final choice.

The topology of the route clusters is visualized on the map with a node-link diagram (Fig. 7C). A conflict marker is placed on every stop in the topology showing the conflict status associated with this stop: 1) *resolved* (blue check marks): this stop is shared by all remaining routes; 2) *active* (orange question marks): this stop is shared in a cluster and belongs to the conflict that is currently being examined in the ranking view; 3) *pending* (gray question marks): this stop is shared in a cluster that is waiting to be examined. The users can obtain the routes that a stop belongs to by hovering on that stop.

7 Evaluation

In this section, we present two usage scenarios and an expert interview to illustrate the effectiveness of the proposed system.

7.1 Usage Scenarios

The following two usage scenarios were conducted to analyze and improve the performance of the Beijing bus network in 2013. The dataset we used comprises 11,414 stops, 653 routes, and 668,346 sampled trip records spanning across two months. We follow Jim, a transportation system analyst, to see how our system can be used in identifying deficient routes and improving these routes interactively.

7.1.1 Deficient Route Identification

After Jim loaded the dataset into the system, he saw the map was divided into several transportation zones, whose performance was summarized with the zone glyphs. The radar charts in the glyphs (Fig. 1B) revealed that bus routes in the northern part of the city carried more passengers than those in the southern part (higher average load and passenger volume). He immediately noticed that one glyph (Fig. 8A) was particularly different from others due to its imbalanced passenger

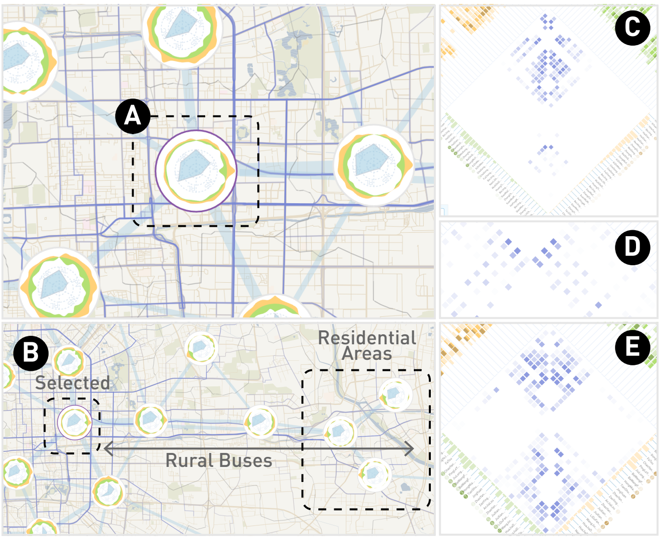


Fig. 8. A) Jim was interested in this glyph because of its heavily-biased passenger flows. B) Many rural buses were originated from the selected zone, connecting this zone, a business district, with residential areas. C, D, E) The matrix patterns that indicate that the second half of the route is not functional (C), indicate that the stops are placed too close to each other (D), and suggest that the route can be split (E).

flows biased heavily towards the east (outside) of the city. Hence, he selected this glyph to further analyze all routes that passed this zone. By judging from the distribution of the routes (Fig. 8B), Jim found that the imbalanced passenger flows were because a) many rural buses were originated from this zone and headed to the east, and b) this zone mainly served as a business district, and many of its employees commuted daily between this zone and the eastern residential districts.

To find potentially deficient routes, Jim applied several filters in the ranking view, including *passenger flows* $\geq$ 10,000 (eliminating the routes with poor data quality), *route length* $\leq$ 35 kilometers (eliminating long-distance rural routes), and *service cost* $\geq$ 2,000 (seeking the routes with high service costs). Nine routes that matched the criteria remained in the ranking view (Fig. 1F). Jim inspected them individually and found the following four interesting patterns:

Route #209. The ranking view revealed that this route had the lowest average load. The matrix view (Fig. 8C) further confirmed that the second half of this route was almost empty and non-functional due to low passenger flows. Hence, shortening such a route or removing a few intermediate stops might improve the performance of the bus network.

Route #675, #701, #715. The matrix view of these routes showed an interleaved pattern (Fig. 8D): the stops adjacent to those with passenger flows had few passenger flows. Jim determined that this was probably because the stops were placed too close to each other, and potential passengers were absorbed by the nearest stops. Based on this pattern, these routes could be further optimized by removing a few stops to reduce the service cost and passenger waiting time.

Route #677. From the matrix view (Fig. 8E), Jim determined that this route might be split into two because passenger flows in the first and second halves of the routes were independent. Such a pattern could also guide the scheduling of buses on this route to meet the demand in the rush hours, such as dispatching more buses to run the first half.

Route #683. In the matrix view, the hourly temporal distribution of check-out records (Fig. 1C) clearly distinguished the stops located in the business (passengers checking out around 9 a.m.) or residential (passengers checking out in the evening) areas, which indicated that this route mainly served as a commuting route. This pattern was also confirmed by the records aggregated based on weekdays. Furthermore, Jim found this route was particularly different from other ones because of the high transfer volume at the origin stop (Fig. 1G). By expanding the transfer route list and examining the pie charts before the routes, Jim found that most passengers transferred to Route #696. He then continued to obtain the matrix of Route #696 (Fig. 1H), where he found that many passengers boarding this route at the transfer stop were heading towards the city west (from the green stop to the orange stop in Fig. 1A). They chose to transfer to Route #696 because Route #683 did not extend into the far west region of the city, where the new residential areas were located. To improve the passengers' experience, Jim suggested that more stops in the west can be added to Route #683, such that the passengers would not have to make any transfers.

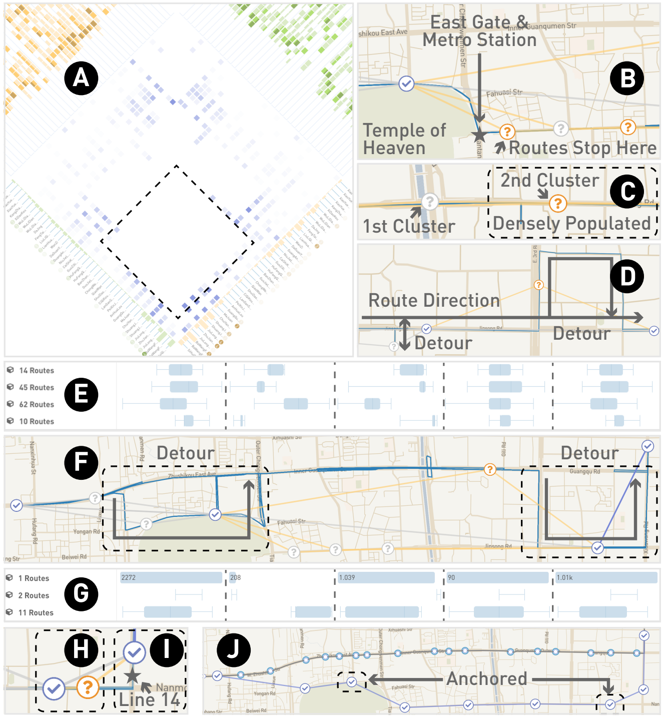


Fig. 9. A) The second half of Route #715 was optimized due to low passenger flows. B) The route cluster stops near the east gate of the Temple and a metro station. C) The second cluster is preferred because it stops in a densely populated area. D, F) The route clusters take multiple detours. E, G) The route clusters generated in step 1 (E) and 2 (G). The criteria from left to right: service time, passenger flow, directness, construction cost, and service cost. H) The stops are too close. I) This stop is located near a metro station. J) The final route (stops are marked with checkmarks) obtained from the decision-making process.

7.1.2 Interactive Route Replacement

In the ranking view (Fig. 1F), Jim ranked the routes with the passenger volume and service cost criteria combined and discovered that Route #715 performed worst due to its low passenger volumes. By checking the matrix view (Fig. 9A), Jim found that the second half of this route was not functional because this part of the route largely overlapped with a metro line. Hence, after observing the map, Jim decided to optimize this route by rerouting it towards the south, such that the route would pass through a tourist attraction (Temple of Heaven) and two hospitals.

Jim selected the part of the route to be optimized and anchored two stops on the map (Fig. 9J), one at the north gate of the Temple and another one near the hospitals. He then started the model and observed the route generation progress in the ranking view. The generation was terminated as the distributions of criteria stabilized, leaving 132 alternative routes in the ranking view. We show how Jim decided the most optimal route progressively with the following four steps.

Step 1. Jim obtained two conflicts and four route clusters from the ranking view (Fig. 9E). He observed from the boxplots that the third cluster had the largest passenger flows. This cluster mainly followed the original route and satisfied the stop anchoring constraint by taking two detours (Fig. 9F), resulting in particularly low route directness. Therefore, Jim inspected the first two clusters, which had the second-largest passenger flows, and found these clusters shared similar characteristics in terms of performance criteria. He then examined these two clusters on the map and identified that the routes in the first cluster would stop at the east gate of the Temple and a nearby metro station (Fig. 9B), gaining potential passenger flows. Hence, the first cluster was selected.

Step 2. After the selection, the ranking view (Fig. 9G) presented another two conflicts with three clusters, one of which comprised only a route. The third cluster had the most passenger flows and provided more diverse choices compared with the first and second because the routes in this cluster were connected with the metro line 14 via the stop illustrated in Fig. 9I. Therefore, Jim proceeded with this cluster.

Step 3. As the decision-making process continued, one conflict with two clusters was depicted in the ranking view. The difference between these two clusters shown on the map (Fig. 9C) with the conflict markers was that the routes in the first cluster stopped near a bridge and those in the second stopped at dense residential areas. Therefore, the second cluster was chosen due to its potential in attracting more passengers.

Step 4. Thereafter, Jim was presented with another three conflicts with three clusters, the first and second of which comprised only a route. For the third cluster which had the most passenger flows, the first and second conflicts indicated that the flows were obtained by taking multiple detours (Fig. 9D), which greatly impacted the bus riders' experience. For the second route, Jim noticed that, in the third conflict, two stops in this route were too close to each other (Fig. 9H), which might lead to the interleaved flow pattern. Hence, Jim concluded that the first route (the blue route in Fig. 9J) could be the most optimal one, given its balanced topology and performance criteria.

7.2 Expert Interview

To collect feedback from experts, we conducted structured interviews individually with four domain experts. First, we introduced the basic functionalities of the system. Then, we demonstrated the usage scenarios to show how the system can be used in detecting and improving deficient routes. Thereafter, we asked the experts to follow the think-aloud protocol and try to explore the bus network and improve a few routes with the system. Their comments during the process were recorded and analyzed. Finally, we requested them to give additional feedback on the usability and effectiveness of the system.

EA spoke highly of the transportation zones and zone glyphs. "*An overview of the bus network is clearly given,*" commented EA. EB shared similar opinions with EA, saying that transportation zones showed a clear structure of the bus network and also facilitated better route filtering. Moreover, EB and EC were both impressed by the matrix view. EB commented that "*A lot of insights can be obtained from such a view.*" Furthermore, EA, EB, and ED praised the conflict-resolving strategy in route decision-making, commenting that such a strategy was "*intuitive*", "*easy to use*", and "*engaging*".

While the experts confirmed the design of our system had met with their analytical requirements, they also gave feedback on how to improve the usability of the system. EA requested that the system shall allow the filters in the ranking view to be reset. EC suggested that an animated line chart showing the number of the generated routes could further help the users control the model. ED recommended adding support for additional geometry file types to enhance the compatibility with other software. The system had been improved accordingly.

To further verify the effectiveness of the proposed system, we have planned to continuously collaborate with the experts after the expert interview to deploy the system for production use and test it in real-world scenarios, which will allow us to fully evaluate the effectiveness of the system with practical use cases in the field.

8 Discussion

In this section, we discuss the implications, lessons learned, limitations, and future work of the proposed system.

Implications. This study is the first step towards the model-assisted visual exploration and planning of bus routes. In terms of *techniques*, BNVA comprises novel visualizations and decision-making strategies to facilitate the improvement of bus routes. In terms of *evaluation*, the usage scenarios indicated several abnormal route patterns that may provide insights for the bus network planners and guide the design of effective bus routes in the future. In terms of *applicability*, the progressive decision-making strategy proposed in this study can be applied to many urban planning scenarios, where the overwhelming number of choices will be reduced to help users make decisions judiciously.

Lessons learned. We present two design lessons we have learned while developing BNVA. First, hierarchical exploration is mandatory for complex data like bus networks. The complexity of bus networks not only lies in the horizontal (network size) and vertical (zones, routes, and stations) scales of the networks but also emerges from the association between the networks and other data sources, such as passenger flows. In this study, we employed a hierarchical exploration approach to dissect a network with multiple coordinated views from the overview to the details and provide visual hints for the users to drill down in the dataset and find interesting patterns. Second, progressive decision-making techniques can be useful in reducing the difficulties in making complex decisions. Excessive choices may overwhelm the cognitive capability of users. In the evaluation interface, we propose a progressive decision-making strategy that assists the users in comparing and evaluating the generated routes through choice aggregation. Users can eliminate a large number of choices efficiently because the choices that share similar characteristics are aggregated. We observed that the experts rapidly adapted to such a strategy and gave positive feedback.

Limitations and future work. Two limitations are observed in this study. First, the route generation model does not consider route shape requirements. Although circular routes can be planned by adding multiple anchored stops, direct support of specifying route shapes will facilitate design flexibility. This functionality can be achieved by integrating route shape optimization, which we will leave as a part of the future work. The second limitation involves the scope of our study. Planning an optimal and practical bus route not only requires optimized spatial structure but also a well-designed timetable and crew assignment. However, this study primarily focuses on the spatial optimization of the existing routes. In the future, we plan to implement other planning stages to establish a complete pipeline of bus network design.

There are some other potential future directions. BNVA is highly extensible in supporting customized quantifiable criteria. Additional data sources, such as traffic data and other transportation networks, can be incorporated to improve the performance of alternative route extraction and evaluation. Some machine learning approaches [80] may improve the performance of route generation. Furthermore, integrating choice recommendations in selecting the optimal routes will considerably improve the efficiency of decision-making. Such integration will be studied in the future to facilitate spatial decision-making.

9 Conclusion

This study presents BNVA, a novel visual analytics system that helps experts visually explore bus networks, identify deficient routes, obtain candidate routes, and determine the best alternative. The exploration, manipulation, and evaluation interfaces are designed to tackle three identified challenges, namely, the in-depth analysis of complex bus route networks, the interactive generation of improved route candidates, and the effective evaluation of alternative bus routes. The proposed system has been evaluated with usage scenarios and expert interviews based on real-world data. In the future, we will implement additional route generation criteria and expand the capability of BNVA by integrating scheduling and resource management functionalities.

Acknowledgments

The work was supported by NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization (U1609217), National Key R&D Program of China (2018YFB1004300), NSFC (61761136020), Zhejiang Provincial Natural Science Foundation (LR18F020001) and the 100 Talents Program of Zhejiang University.

References

[1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Human-Computer Interaction Series. Springer, 2011.

[2] G. Andrienko, N. Andrienko, P. Bak, D. A. Keim, and S. Wrobel. *Visual Analytics of Movement*. Springer, 2013.

[3] G. Andrienko, N. Andrienko, and U. Bartling. Visual analytics approach to user-controlled evacuation scheduling. *Information Visualization*, 7(1):89–103, 2008.

[4] 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 Trans. Intell. Transp. Syst.*, 18(8):2232–2249, 2017.

[5] 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 Trans. Vis. Comput. Graph.*, 23(9):2120–2136, 2017.

[6] G. Andrienko, N. Andrienko, C. Hurter, S. Rinzivillo, and S. Wrobel. From movement tracks through events to places: Extracting and characterizing significant places from mobility data. In *Proc. of IEEE VAST*, pages 161–170. IEEE, 2011.

[7] G. Andrienko, N. Andrienko, M. Mladenov, M. Mock, and C. Pölitz. Discovering bits of place histories from people's activity traces. In *Proc. of IEEE VAST*, pages 59–66. IEEE, 2010.

[8] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti. Interactive visual clustering of large collections of trajectories. In *Proc. of IEEE VAST*, pages 3–10. IEEE, 2009.

[9] N. Andrienko, G. Andrienko, L. Barrett, M. Dostie, and S. P. Henzi. Space transformation for understanding group movement. *IEEE TVCG*, 19(12):2169–2178, 2013.

[10] N. Andrienko, G. Andrienko, G. Fuchs, and P. Jankowski. Scalable and privacy-respectful interactive discovery of place semantics from human mobility traces. *Information Visualization*, 15(2):117–153, 2016.

[11] N. Andrienko, G. Andrienko, F. Patterson, and H. Stange. Visual analysis of place connectedness by public transport. *IEEE Transactions on ITS*, pages 1–13, 2019. Early Access.

[12] J. Bao, T. He, S. Ruan, Y. Li, and Y. Zheng. Planning bike lanes based on sharing-bikes' trajectories. In *Proc. of ACM SIGKDD*, pages 1377–1386. ACM, 2017.

[13] R. Beecham and J. Wood. Exploring gendered cycling behaviours within a large-scale behavioural data-set. *Transportation Planning and Technology*, 37(1):83–97, 2014.

[14] C. Browne, E. J. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. P. Liebana, S. Samothrakis, and S. Colton. A survey of Monte Carlo tree search methods. *IEEE TCIAIG*, 4(1):1–43, 2012.

[15] N. Cao, C. Shi, W. S. Lin, J. Lu, Y. Lin, and C. Lin. TargetVue: Visual analysis of anomalous user behaviors in online communication systems. *IEEE TVCG*, 22(1):280–289, 2016.

[16] P. Chakroborty and T. Wivedi. Optimal route network design for transit systems using genetic algorithms. *Engineering optimization*, 34(1):83–100, 2002.

[17] C. Chen, D. Zhang, N. Li, and Z. Zhou. B-planner: Planning bidirectional night bus routes using large-scale taxi GPS traces. *IEEE Transactions on ITS*, 15(4):1451–1465, 2014.

[18] A. Chernev, U. Böckenholt, and J. Goodman. Choice overload: A conceptual review and meta-analysis. *Journal of Consumer Psychology*, 25(2):333–358, 2015.

[19] Z. Deng, D. Weng, J. Chen, R. Liu, Z. Wang, J. Bao, Y. Zheng, and Y. Wu. AirVis: Visual analytics of air pollution propagation. *IEEE TVCG*, 26(1):800–810, 2020.

[20] ESRI Inc. ArcGIS desktop. `https://desktop.arcgis.com/en/`. Accessed: 2020-03-15.

[21] W. Fan and R. B. Machemehl. Using a simulated annealing algorithm to solve the transit route network design problem. *Journal of transportation engineering*, 132(2):122–132, 2006.

[22] A. Farooq, M. Xie, S. Stoilova, F. Ahmad, M. Guo, E. J. Williams, V. K. Gahlot, D. Yan, and A. Mahamat Issa. Transportation Planning through GIS and Multicriteria Analysis: Case Study of Beijing and XiongAn. *Journal of Advanced Transportation*, 2018:2696037, Oct. 2018.

[23] N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva. Visual exploration of big spatio-temporal urban data: A study of New York City taxi trips. *IEEE TVCG*, 19(12):2149–2158, 2013.

[24] S. Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2(1-4):153, 1987.

[25] Gapminder Foundation. Gapminder tools. `https://www.gapminder.org/tools/#$chart-type=bubbles`. Accessed: 2020-03-12.

[26] P. Gatalsky, N. Andrienko, and G. Andrienko. Interactive analysis of event data using space-time cube. In *Proc. of IV*, pages 145–152. IEEE, 2004.

[27] S. Goodwin, J. Dykes, S. Jones, I. Dillingham, G. Dove, A. Duffy, A. Kachkaev, A. Slingsby, and J. Wood. Creative user-centered visualization design for energy analysts and modelers. *IEEE TVCG*, 19(12):2516–2525, 2013.

[28] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. LineUp: Visual analysis of multi-attribute rankings. *IEEE TVCG*, 19(12):2277–2286, 2013.

[29] J. Guan, H. Yang, and S. C. Wirasinghe. Simultaneous optimization of transit line configuration and passenger line assignment. *Transportation Research Part B: Methodological*, 40(10):885–902, 2006.

[30] V. Guihaire and J.-K. Hao. Transit network design and scheduling: A global review. *Transportation Research Part A: Policy and Practice*, 42(10):1251–1273, 2008.

[31] J. Heer, N. Kong, and M. Agrawala. Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. In *Proc. of ACM CHI*, pages 1303–1312. ACM, 2009.

[32] R. Hughes. Visualization in transportation: Current practice and future directions. *Transportation research record*, 1899(1):167–174, 2004.

[33] International Business Machines. CPLEX optimizer. `https://www.ibm.com/analytics/cplex-optimizer`. Accessed: 2020-03-12.

[34] S. C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[35] D. A. Keim. Information visualization and visual data mining. *IEEE TVCG*, 8(1):1–8, 2002.

[36] L. Kocsis and C. Szepesvári. Bandit based Monte-Carlo planning. In *Proc. of ECML*, volume 4212, pages 282–293. Springer, 2006.

[37] C. Li, X. Dong, and X. Yuan. Metro-wordle: An interactive visualization for urban text distributions based on wordle. *Vis. Informatics*, 2(1):50–59, 2018.

[38] Y. Li, Y. Zheng, S. Ji, W. Wang, L. H. U, and Z. Gong. Location selection for ambulance stations: A data-driven approach. In *Proc. of ACM SIGSPATIAL*, pages 85:1–4. ACM, 2015.

[39] Y. Liang and J. Chiew. Rerouting buses using data science - Part I. `https://blog.data.gov.sg/rerouting-buses-using-data-science-part-i-4d6c9d4f1f`. Accessed: 2020-03-12.

[40] D. Liu, D. Weng, Y. Li, J. Bao, Y. Zheng, H. Qu, and Y. Wu. SmartAdP: Visual analytics of large-scale taxi trajectories for selecting billboard locations. *IEEE TVCG*, 23(1):1–10, 2017.

[41] H. Liu, S. Jin, Y. Yan, Y. Tao, and H. Lin. Visual analytics of taxi trajectory data via topical sub-trajectories. *Vis. Informatics*, 3(3):140–149, 2019.

[42] M. Liu, S. Liu, X. Zhu, Q. Liao, F. Wei, and S. Pan. An uncertainty-aware approach for exploratory microblog retrieval. *IEEE TVCG*, 22(1):250–259, 2016.

[43] S. Liu, D. Maljovec, B. Wang, P. Bremer, and V. Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE TVCG*, 23(3):1249–1268, 2017.

[44] C. MacKechnie. How do bus routes and schedules get planned? `https://www.liveabout.com/bus-routes-and-schedules-planning-2798726`. Accessed: 2020-03-11.

[45] C. Mandl. *Applied network optimization*. Operations research and industrial engineering. Academic Press, 1979.

[46] C. E. Mandl. Evaluation and optimization of urban public transportation networks. *European Journal of Operational Research*, 5(6):396–404, 1980.

[47] P. Mees, J. Stone, M. Imran, and G. Nielson. Public transport network planning: a guide to best practice in NZ cities. Technical Report 396, March 2010.

[48] M. Mistretta, J. A. Goodwill, R. Gregg, and C. DeAnnuntis. Best practices in transit service planning. Technical Report #BD549-38, March 2009.

[49] T. Munzner. A nested process model for visualization design and validation. *IEEE TVCG*, 15(6):921–928, 2009.

[50] T. Munzner. *Visualization Analysis and Design*. A.K. Peters visualization series. A K Peters, 2014.

[51] A. T. Murray. A coverage model for improving public transit system accessibility and expanding access. *Annals of Operations Research*, 123(1-4):143–156, 2003.

[52] B. Ni, Q. Shen, J. Xu, and H. Qu. Spatio-temporal flow maps for visualizing movement and contact patterns. *Vis. Informatics*, 1(1):57–64, 2017.

[53] Optibus Inc. Transportation software for the mass – transit planning and scheduling. `https://www.optibus.com`. Accessed: 2020-03-12.

[54] M. L. Pack. Visualization in transportation: Challenges and opportunities for everyone. *IEEE CG&A*, 30(4):90–96, 2010.

[55] S. B. Pattnaik, S. Mohan, and V. M. Tom. Urban bus transit route network design using genetic algorithm. *Journal of transportation engineering*, 124(4):368–375, 1998.

[56] F. Pinelli, R. Nair, F. Calabrese, M. Berlingerio, G. Di Lorenzo, and M. L. Sbodio. Data-driven transit network design from mobile phone trajectories. *IEEE Transactions on ITS*, 17(6):1724–1733, 2016.

[57] J. Sewall, D. Wilkie, and M. C. Lin. Interactive hybrid simulation of large-scale traffic. *ACM Trans. Graph.*, 30(6):135, 2011.

[58] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. of IEEE Symp. on Visual Languages*, pages 336–343. IEEE, 1996.

[59] G. Sun, Y. Liu, W. Wu, R. Liang, and H. Qu. Embedding temporal display into maps for occlusion-free visualization of spatio-temporal data. In *Proc. of IEEE PacificVis Symp.*, pages 185–192. IEEE, 2014.

[60] X. Sun, C. G. Wilmot, and T. Kasturi. Household travel, household characteristics, and land use: an empirical study from the 1994 portland activity-based travel survey. *Transportation Research Record*, 1617(1):10–17, 1998.

[61] The Public-Private Infrastructure Advisory Facility. Factors influencing bus system efficiency. `https://ppiaf.org/sites/ppiaf.org/files/documents/toolkits/UrbanBusToolkit/assets/1/1d/1d.html`. Accessed: 2020-03-11.

[62] Y. Wang, Y. Zheng, and Y. Xue. Travel time estimation of a path using sparse trajectories. In *Proc. of ACM SIGKDD*, pages 25–34. ACM, 2014.

[63] D. Weng, R. Chen, J. Zhang, J. Bao, Y. Zheng, and Y. Wu. Pareto-optimal transit route planning with multi-objective Monte-Carlo tree search. *IEEE Transactions on ITS*, pages 1–11, 2020. Early Access.

[64] D. Weng, H. Zhu, J. Bao, Y. Zheng, and Y. Wu. HomeFinder revisited: Finding ideal homes with reachability-centric multi-criteria decision making. In *Proc. of ACM CHI*, page 247. ACM, 2018.

[65] W. Wu, Y. Zheng, H. Qu, W. Chen, M. E. Gröller, and L. M. Ni. BoundarySeer: Visual analysis of 2D boundary changes. In *Proc. of IEEE VAST*, pages 143–152. IEEE, 2014.

[66] Y. Wu, D. Weng, Z. Deng, J. Bao, M. Xu, Z. Wang, Y. Zheng, Z. Ding, and W. Chen. Towards better detection and analysis of massive spatiotemporal co-occurrence patterns. *IEEE Transactions on ITS*, pages 1–16, 2020. Early Access.

[67] X. Xie, J. Wang, H. Liang, D. Deng, S. Cheng, H. Zhang, W. Chen, and Y. Wu. Passvizor: Toward better understanding of the dynamics of soccer passes. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):To appear, 2021.

[68] Y. Xiong and J. B. Schneider. Transportation network design using a cumulative genetic algorithm and neural network. *Transportation Research Record*, (1364), 1992.

[69] D. Yang, D. Zhang, and B. Qu. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM TIST*, 7(3):30:1–30:23, 2016.

[70] B. Yu, Z. Yang, C. Cheng, and C. Liu. Optimizing bus transit network with parallel ant colony algorithm. In *Proc. of Eastern Asia Society for Transportation Studies*, volume 5, pages 374–389, 2005.

[71] W. Zeng, C. Fu, S. M. Arisona, A. Erath, and H. Qu. Visualizing mobility of public transportation system. *IEEE TVCG*, 20(12):1833–1842, 2014.

[72] W. Zeng, C. Fu, S. M. Arisona, and H. Qu. Visualizing interchange patterns in massive movement data. *CGF*, 32(3):271–280, 2013.

[73] F. Zhao. Large-scale transit network optimization by minimizing user cost and transfers. *Journal of Public Transportation*, 9(2):6, 2006.

[74] F. Zhao and I. Ubaka. Transit network optimization – minimizing transfers and optimizing route directness. *Journal of Public Transportation*, 7(1):4, 2004.

[75] F. Zhao and X. Zeng. Optimization of transit route network, vehicle headways and timetables for large-scale transit networks. *European Journal of Operational Research*, 186(2):841–855, 2008.

[76] J. Zhao, P. Forer, and A. S. Harvey. Activities, ringmaps and geovisualization of large human movement fields. *Information Visualization*, 7(3-4):198–209, 2008.

[77] J. Zhao, Z. Liu, M. Dontcheva, A. Hertzmann, and A. Wilson. MatrixWave: Visual comparison of event sequence data. In *Proc. of ACM CHI*, pages 259–268. ACM, 2015.

[78] Y. Zheng, L. Capra, O. Wolfson, and H. Yang. Urban computing: Concepts, methodologies, and applications. *ACM TIST*, 5(3):38:1–38:55, 2014.

[79] Y. Zheng, W. Wu, Y. Chen, H. Qu, and L. M. Ni. Visual analytics in urban computing: An overview. *IEEE Transactions on Big Data*, 2(3):276–296, 2016.

[80] Z. Zhou. Abductive learning: towards bridging machine learning and logical reasoning. *Sci. China Inf. Sci.*, 62(7):76101:1–76101:3, 2019.