SIGSPATIAL: U: An Intelligent and Interactive Route Planning Maker for Deploying New Transportation Services

Fandel Lin
Department of Electrical Engineering
National Cheng Kung University, Tainan, Taiwan
e24046666@mail.ncku.edu.tw

Hsun-Ping Hsieh
Department of Electrical Engineering
National Cheng Kung University, Tainan, Taiwan
hphsieh@mail.ncku.edu.tw

ABSTRACT

In this work, a novel decision supporting tool, called Route Planning Maker (RPM) is proposed to help governments or transportation companies to plan new route services in the city. The function of RPM is four-fold. First, RPM illustrates the local characteristics (e.g. geo-graphical information or spatial-temporal urban informatics) by visualizing multiple aspects of the city for users to easily understand the local characteristics of anywhere in the city, which is beneficial for proposing new routes. Second, RPM has a flexible user interface that allows users to arbitrarily sketch/adjust their idea by adding/removing routes and stations when deploying new routes. Besides, RPM can also show the existing routes which are correlated with the new route to let users check their transference or overlapping regions. Third, RPM provides an intelligent function to estimate passenger flows (PF) in certain time intervals and acquire relevant urban information so that the user can estimate the effectiveness of designed routes. According to our experimental results, RPM can obtain the passenger flow effectively and efficiently for given designated routes. However, in some cases users prefer that PRM can directly recommend a route which will have high potential PF. Therefore, the last function of RPM is that we proposed Bidirectional Prioritized Spanning Tree (BDPST) for route recommendation given users' constraints. By utilizing our proposed inference model and BDPST algorithm, RPM can recommend routes and stations with high potential PF in a certain extent on map along with some must-visit stations assigned by users. We did experiments on bus-ticket data of Tainan city and the results show that the inference model outperforms baseline and comparative methods from 17% to 75%. Moreover, we show that the proposed BDPST algorithm's performance is not too far away from the optimal PF and outperforms other comparative methods from 9% to 70% in large scale route recommendation.

CCS CONCEPTS

• **Information systems** → Web applications; Location based services;

• **Theory of computation** → Graph algorithms analysis;

KEYWORDS

Route Planner, Interactive Route Plan, Passenger Volume Estimation, Urban Planning, Bidirectional Spanning Tree

1 Problem and Motivation

Traffic deployment is high-correlated with the quality of life [13]. Governments or transportation companies dispose new transportation services such as bus or MRT routes to serve residents. For residents, new services bring convenience to users and reduce pollution. However, constructing unwanted and redundant routes or stations can lead to environmental damage and resource waste. Besides, according to our interview with civil servants in the bureau of transportation, they pointed out that the current procedure in planning new routes turns out to be lengthy due to many stakeholders involved in. Also, the overwhelming number of requests from public opinions makes it difficult to decide where to construct new routes and stations.

Therefore, we propose a novel tool, called Route Planning Maker (RPM), which not only has flexible interface to customize a new route efficiently and visually, but also assesses the effectiveness of a new route service in advance before deployment. RPM has four main functions. The first is visualization of urban characteristics, the second refers to the interactive route design, the third is passenger volume inference for designed route, and the last is high PF route recommendation based on given constraints. More specifically, RPM allows users to estimate the passenger volumes of their proposed routes, or directly recommends stations and a route with high potential PF based on desired must-visit stations in a certain extent. The system interface is shown in Figure 1.

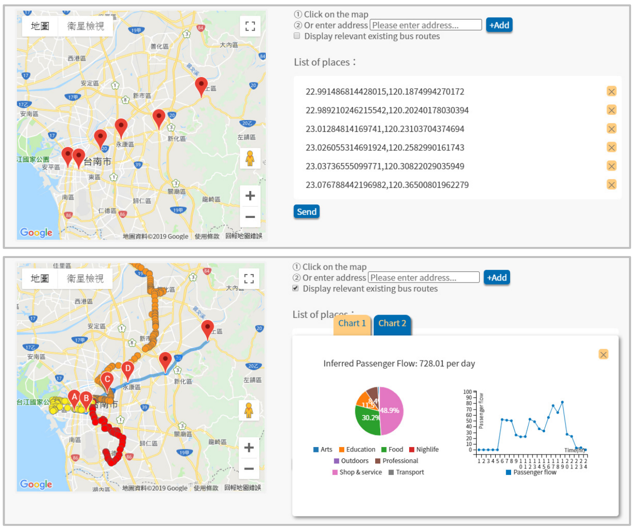

Figure 1: An overview of the proposed RPM system, including user query by clicking on map (up) and inference results (down).

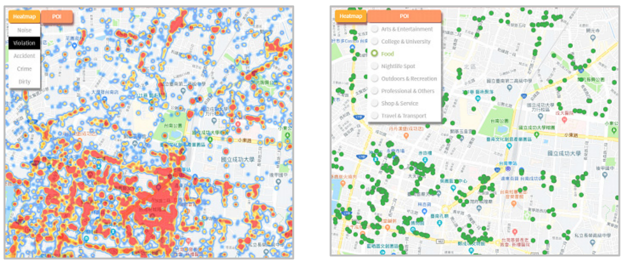

Figure 2: Heat map (left) and scattered point map (right).

Table 1: Summary of formulations for researches on designing new transportation routes.

	Feature extraction												System functionality				
	Station deployment	Interval deployment	Surrounding POI	Route transference	Route competition	Population structure	Human mobility	Network structure	Waiting time	Journey interval	Station distance	Trajectory length	Insert stations	Design route	Passenger Flow	Make free with map	Visualization
Mauttone & Urquhart (2009) [8]	■					■		■	■			■	■	■			
Quadrifoglio & Li (2009) [10]	■			■							■	■		■			
Szeto & Wu (2011) [15]	■	■		■	■			■	■				■	■			
Cancela et al. (2015) [2]	■			■		■		■	■			■		■			
Pternea et al. (2015) [9]	■			■				■	■					■			
Ours	■		■	■	■	■	■	■		■	■	■	■	■	■	■	■

2 Background and Related Work

Inferring the passenger volume of a new route is not trivial since multiple factors such as human mobility, impacts of existing routes and stations, road network structure, point-of-interests (POI) and local population structure should be considered. Therefore, we propose several strategies to support the third function. First, we believe features observed in our proposed route-affecting region (RAR), which indicates the influential range of a route, would influence the passenger flow (PF) of the route. Second, we exploit Deep Neural Network (DNN) to estimate PF by combining heterogeneous features. Third, to handle the effects on new routes along with existing ones, we propose to train the impacts by considering extended, nearby, and overlap segments. To our best knowledge, no existing work deals with such research problem. We claim that our proposed RPM system can be used in many kinds of urban transportation, such as subway and bus routes, and can be utilized in any cities, where ticket data is available for the government or transportation companies.

Related Works. Some works [2][8][9][10][15] in designing new transportation route focused on decreasing transportation time through route adjustment and shift. There are also some works [6][16] that optimized the route planning, in which distance, time, transference and passenger flow were considered. On the other hand, some works [3][7][11] studied the problem of predicting arrival time for public transportation based on regression analysis. Moreover, some researches [17][18] focused on the problem of predicting future passenger-flow. However, the above works focused on dealing with existing routes, which are not our target problem.

Therefore, by focusing on designing new transportation routes, Table 1 first presents a summary of formulations for previous works, listing the aspects each research approached. By analyzing the considered and extracted features, we generalize six kinds of relevant urban features in inferring the passenger flow for new route deployed in transportation network, which will be discussed in section 4.1.2. Most importantly, none of these works present an interface or a tool for users to design their own routes intuitively and interactively.

3 Overview

This paper presents a novel assisting system for deploying new public transportation system. The proposed system has three phases. In the first phase, relevant urban characteristics mined from open data are visualized on a map for users to preview the neighboring environments. The second phase allows users to design new routes and the passenger flow in several time intervals will then be inferred; meanwhile, existing routes that can be transferred will be displayed on map in the same time. In the third phase, users can query some constraints and our system can recommend stations and route.

System Prototype. The interface overview is shown in Figure 1 and 2. The system has four functions: (a) Visualization of relevant urban features on base map. (b) A base map for users to schedule their customized placement of routes and stations, and a control panel that allows users to add or delete stations or constrains. (c) The PF and relevant information of the route will be inferred and displayed. (d) The stations and route will be recommended and displayed.

A prototype of RPM has been developed and deployed as a web application. We use Google Maps for the whole city as the base map. The web application can be opened in a standard web browser without any additional software or hardware. When using this system, users can not only perceive certain regions for different kinds of urban characteristics, draw their own route on demand and let the system infer the PF of the designated route, but also set constraints for system to recommend stations and route.

3.1 Visualization of Urban Characteristics

To let users perceive certain regions when deploying new routes or stations, the system illustrates the urban characteristics by visualizing multiple aspects of the city. Visualized data includes POI information, which is crawled regularly from Google Maps in RPM system. The categories for POI visualization are arts, education, food, night life, outdoors, professional space, shop and service, and transportation spot. Other data including violation events, pollution reports, traffic accidents, crimes are from open data platform of Tainan City Government. The visualization is shown in Figure 2.

3.2 Route Design and Passenger Flow Inference

To design their own routes and stations, users can add stations by directly clicking on the map or by entering an address. Besides, certain stations can be deleted by clicking the corresponding buttons in the list. After submitting the designated route, the system will immediately infer the PF per day of the route with stations deployed; meanwhile, relevant information including PF in different time intervals, statistics of POI, population and road network structure of the nearby region are also shown in the panel as a complement for designing new route services. In addition, users can display existing routes which can be transferred to or correlated with the designated route on map. An overview of the inference phase is shown in Figure 1 and 3. In the section 4.1, we introduce the detail of PF inference.

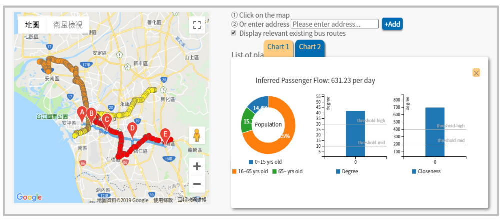

Figure 3: Inference results, statistics and relevant existing routes.

3.3 Route Recommendation with Constraints

Although RPM system can help users estimate PF for any designed route, sometime users prefer to obtain a recommended route which have high PF value in a certain urban space. Therefore, we design another function that can recommend stations and route with constraints given by users. To start with, users can set range to consider by directly clicking the corners of their desired extent on the map. Next, users can set some must-visit stations by clicking on map or entering addresses. Finally users can set the number of desired stations and submitting the request, the system will immediately display the recommended route and stations including must-visit ones and recommended ones. In the section 4.2, we introduce the detail of methodology.

4 Approach and Uniqueness

The methodology for this work is composed of two parts, the PF inference, and route recommendation, which utilizes the result of the former part. Meanwhile, the evaluation results for these proposed approaches are introduced in section 5.1 and 5.2 separately.

4.1 PF Inference

Problem Definition. Given a set of trajectories for the designated route with its stations labeled from users, our goal is to infer the passenger flow $PF(l_i t_j)$ for each route l_i in certain time intervals t_j. In other words, we devise RPM for users to plan their own routes and stations. Then, the system derives the passenger flow of the user-designated trajectory and stations in a certain time interval.

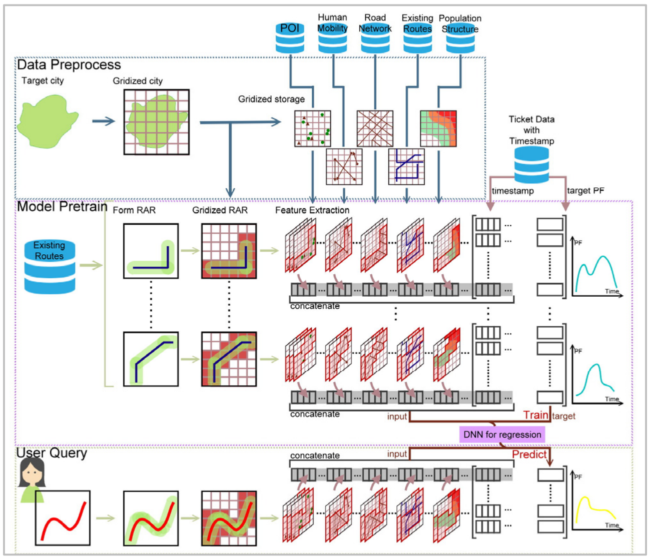

Figure 4: PF inference procedure in RPM.

The framework of PF inference in RPM is shown in Figure 4, which mainly consists of three components. In data preprocessing, we divide the city into disjointed grids (e.g., 0.1km × 0.1km) [12], and all features are fetched and stored in grids for further extraction. The second component is training models. The feature set for each existing route is extracted and integrated as the training data along with its corresponding ticket data, which is associated with timestamp and PF for each route. We treat various features as inputs and PF as the predictive label. We tried several machine learning methods as training models, and the DNN for regression gets the most promising result in our evaluation. In the third component, the pre-trained model is utilized for the query route given by user to infer PF value.

4.1.1 Route-Affecting Region

The demand for public transportation is not only based on the origin and destination, but also the nearby geographical environment and urban functions of nearby areas. Thus, we propose RAR for considering passenger-flow related features. A route can comprise multiple segments that contain successive points close to each other. Then we can draw a circle for each point, where we consider each point as the center of a circle, and then RAR formed by a set of circles. Based on Design Manual for Urban Sidewalks [14], the tolerance walking distance for pedestrian is 400 to 800 meters; therefore, the green area in Figure 5 is an example of RAR of the given route q_s to q_d, with a radius of 0.4km. Then we can extract the corresponding features correlated with passenger flow within RAR.

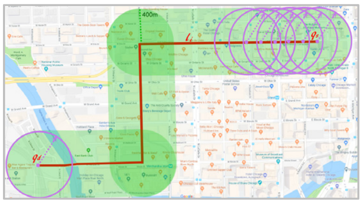

Figure 5: A RAR example of a user-designated route

4.1.2 Feature Extraction Based on RAR

To infer the PF value of the trajectory correctly, we consider six kinds of relevant urban features in RAR:

(1) POI-Related Features

Various POIs (specific point location such as transportation hubs or entertainment venues) and their density in RAR indicate the function of a region, which might have high correlation to the PF of a route. For example, a high PF might be associated with route to many shopping centers. We consider two aspects of POI features as follows:

POI Density. The density of POI indicates the popularity of a certain activity type in RAR. As the example mentioned above, a high density of certain types of POI such as shopping centers and schools can result in high PF value.

POI Entropy. The POI entropy in RAR shows the diversity of purpose for people to visit the nearby area of a route. The entropy for trajectory l_i is based on Information Theory [4]:

$$\text{Entropy}(l_i) = -\sum_{\gamma\in\Gamma}\left(\frac{N_\gamma(l_i,r)}{N(l_i,r)}\times\log\frac{N_\gamma(l_i,r)}{N(l_i,r)}\right) \qquad (1)$$

Where Γ indicates the set of POI, and γ refers to certain type of POI. Besides, N (l_i, r) displays the total number of POI in RAR of trajectory l_i based on radius r, Nγ (l_i, r) displays the number of type-γ POI in RAR of trajectory l_i relatively.

(2) Human Mobility

Human mobility is extracted from ticket data in three ways: transition density, incoming flow, and leaving flow. The transition density indicates the ratio of transitions occurred in the same RAR. The incoming flow shows the total records entering the RAR; on the contrary, the leaving flow displays the total records exiting the RAR.

(3) Road Network Structure

Road network structure, including degree and closeness centrality, is considered since it might be correlated with real traffic conditions. We extract network structures from OpenStreetMap (OSM), where degree centrality identifies the total number of reachable vertexes for all intersections in RAR, and closeness centrality shows the average distance between one intersection to another in RAR.

(4) Competition and Transference with Existing Routes

Two routes might cause a competitive relationship if their RAR is similar. However, intersected routes with considerable extended segments would encourage passengers to transfer between them. Therefore, we seek intersections between designated routes and existing routes, and then calculate the extended, nearby, and overlap grids each transferable existing route holds, as shown in Figure 6, then finally sum up each type of grids as features.

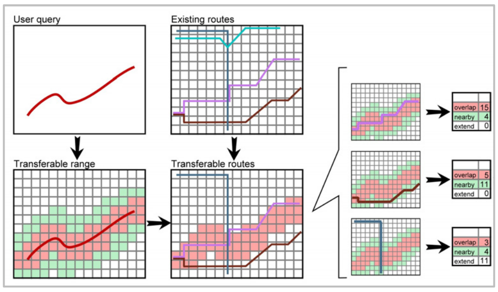

Figure 6: Procedure of dealing with existing routes.

(5) Population Structure

People in RAR of different ages and genders can have different intentions for taking public transportation. Consequently, we extract the population data of the target city, and then the population for each age group and gender are normalized as features to be considered. In this work, we crawled population-related data of Tainan City from Ministry of Interior's (MOI, Taiwan) open data platform.

(6) Time Information and Granularity

Seasons and holidays can influence the passenger flow of public transportation. We adopt one-hot encoding to record the time information for each ticket record.

4.1.3 Inference Model Construction

We adopt and modify multiple machine learning methods including SVR, Linear Regression, XGBoost, and DNN for Regression, to derive the PF for the designated route respectively. The input data includes all the features extracted based on the RAR of the user-designated route, including POI-related properties, human mobility, road network structure, correlations of existing routes, population structure, and time information. As it turns out, the output is the inferred PF value of the user-designated route.

4.2 High PF Route Recommendation

Problem Definition. Given a set of must-visit stations $S_M = \{S_{M0}, \ldots, S_{Mi}\}$ and extent along with constraints including the number of recommended stations r, our goal is to recommend a trajectory in the given extent along with a set of stations $S=S_M+S_R$ to maximizes the inferenced PF per unit length for the route (combination of trajectory and stations), where S_M refers to the set of must-visit stations, and S_R is the recommended stations $\{S_{R0}, \ldots, S_{Rr}\}$.

The optimal solution is quite difficult to obtain in large urban space since there are too many combinations of route segments and stations for forming a route. According to our experiments, some exhaustion-based methods are not feasible due to high execution time. Therefore, we proposed Bidirectional Prioritized Spanning Tree to help retrieve a not bad solution using reasonable time. The pseudo code for the proposed BDPST Algorithm for stations and route recommendation is shown in Figure 8, which consists of three parts. First, it calculates the PF of each must-visit grid utilizing the inference model proposed in section 4.2; then evaluates its spreading impact on other grids in the extent based on Gaussian function in two dimensions. Second, based on the negative Gaussian feedback of inference PF from must-visit and selected stations, scores for all grids are derived and we greedily select the grid with maximal PF as the recommended station.

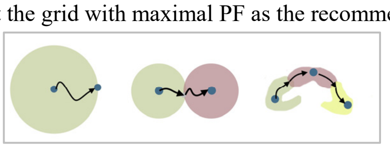

Figure 7: Schematic search space of Dijkstra's algorithm (left), bidirectional search (middle), and BDPST algorithm (right).

```
Algorithm BDPST
input: must-visit grids (stations) in map extent with road network, and
number of recommended stations
output: route with starter as stations such that it has a unit-length PF
approximating optimal solution
    foreach grid in extent do
        let pf(grid) be the PF of grid
        if grid is in must_visit_grids then
            foreach g in extent do
                let G(g, grid) be Gaussian function in 2 dimension
                gd(g, grid) ← G(g, grid)×pf(grid)
                tgd(g) ← tgd(g)+gd(g, grid)
    foreach j from 0 to number_of_recommendation do
        max(j) ← 0
        choice ← null
        foreach g in extent do
            if g is not in (must_visit_grids + recommended_grids) then
                if pf(g)−tgd(g)>max(j) then
                    max(j) ← pf(g)−tgd(g)
                    choice ← g
        add choice to recommended_grids
        foreach g in extent do
            let G(g, choice) be Gaussian function in 2 dimension
            gd(g, choice) ← G(g, choice)×pf(choice)
            tgd(g) ← tgd(g)+gd(g, choice)
    starter ← must_visit_grids + recommended_grids
    foreach grid in extent do
        sid(grid) ← -1
        if grid is in starter then
            root[starter_id] ← grid
            set[starter_id, 0].add(root[starter_id])
            sid(grid) ← starter_id
            count(starter_id) ← 0
            foreach g in extent do
                tgd[starter_id,0](g) ← tgd(g)−2×gd(g, grid)
    route ← null
    foreach i from 0 to infinite do
        foreach s in starter do
            foreach st in set[s, i] do
                if count(s)≥2 then
                    break
                nst ← st /*deep copy*/
                pt(nst, s) ← 0
                max(st) ← 0
                choice ← null
                foreach g in nearby_grid do
                    if reachable in road network and sid(g)≠s then
                        if tgd[s, i](g)>max(st) then
                            max(st) ← tgd[s, i](g)
                            choice ← g
                if choice≠null then
                    nst.visit(choice)
                    if sid(g)≠-1 and count(sid(g))<2 then
                        route.addseg(nst, obj(g))
                        count(s) ← count(s)+1
                        count(sid(g)) ← count(sid(g))+1
                        if success to form a route with all starters then
                            break
                        foreach g in extent do
                            tgd[s, i](g) ← tgd[s, i](g)−2×gd(g, root[sid(g)])
                        if count(sid(g))=2 /*same for count(s)=2*/ then
                            foreach st in starter do
                                foreach g in extent do
                                    tgd[st, i](g) ← tgd[st, i](g)−2×gd(g, root[sid(g)])
                    else
                        set[s, i+1].add(nst)
                        sid(g) ← s
                        obj(g) ← nst
                pt(st, s) ← pt(st, s)+1
                if pt(st, s)>1 then
                    set[s, i].delete(st)
            set[s, i+1].add(set[s, i])
            tgd[s, i+1] ← tgd[s, i]
        if success to form a route with all starters then
            break
    return route and starter
```

Figure 8: Pseudo code for the proposed BDPST algorithm.

The standard solution for one-to-all path problem is Dijkstra's algorithm [5], which updates values by superimposing connections iteratively. However, the features in our model including entropy and relationship with existing routes are not superimposable, which makes Dijkstra's algorithm or other route planning algorithm in transportation network inappropriate for our case [1]. Therefore, based on the idea of bidirectional search and depth-first spanning tree, we propose the Bidirectional Prioritized Spanning Tree. In the third component, BDPST minimizes depth of search by performing multi-bidirectional search and prunes breadth of search space on the basis of spreading impact of positive Gaussian feedback from other stations, which makes it act as a breadth-first-based target-prioritized spanning tree growing from multiple stations(starters) simultaneously.

We compare the schematic search space of Dijkstra's algorithm, bidirectional search, and our BDPST algorithm in Figure 7. As there is provably an intersection of both search spaces, bidirectional search visits roughly half as many grids as Dijkstra's algorithm. Moreover, BDPST visits fewer grids than bidirectional as the target-prioritized approach restricts the breadth based on the tendency to other targets.

5 Evaluation

Dataset. For both parts, we use bus-ticket data from Tainan City Government, which dataset contains 14,336,226 ticket records. The city bus system holds 104 routes and 6,575 stations; meanwhile, each ticket record lists route and timestamps, starting and ending station.

5.1 PF Inference

In this section, we hold two kinds of experimental scenarios. The first one queries only one route (trajectory-based), and the second one is a route with deployed locations (station-based). We developed four comparative methods: (a) *Support Vector Regression*, (b) *Linear Regression*, (c) *XGBoost*, and (d) *DNN for Regression*, along with two baseline methods: (e) *Median value* and (f) *Average value* using the median and average value of PF in all training routes respectively.

The evaluation is based on the leave-one-out method. In the trajectory-based scenario, for instance, we leave one route PF data out of the complete data, and then use the rest of data to train the model and infer the value of the left out route based on the features extracted from RAR of the route. Then we compare the inference value with the ground-truth, which is the total number of passengers that had taken the route. Thereafter, we leave the next PF and use the remaining data to train and infer again until each PF is inferred and compared with its ground-truth. Finally, we retrieve the normalized root-mean-square error. Accordingly, performance result is shown in Figure 9, where *DNN for Regression* gains the best normalized RMSE and outperforms other comparative methods for at least 75% in station-based scenarios and 17% in trajectory-based situations.

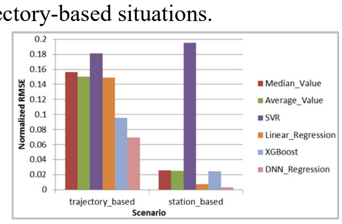

Figure 9: Normalized RMSE of PF results for different methods.

5.2 Route Recommendation

In this section, we developed five comparative methods: (a) *Bidirectional-Spanning-Tree(BDST)* uses multi-bidirectional concept based on BFS. (b) *Breadth-First-Spanning-Tree(BFST)* recommends stations on grid PF with negative Gaussian feedback and runs BFS station by station. (c) *Mixture-Depth-First-Spanning-Tree(MDFST)* recommends stations based on inferenced PF with negative Gaussian feedback and runs DFS station by station. (d) *Gaussian-Depth-First-Spanning-Tree(GDFST)* recommends stations based on negative Gaussian feedback and runs DFS station by station. (e) *Random-Depth-First-Spanning-Tree(RDFST)* runs DFS station by station. The baseline method: (f) Brute-Force (BF) systematically enumerates all possible combinations for solution and retrieves the optimal one.

The evaluation is based on the average PF per unit length and executing time. All methods run 1,000 randomly generated testing cases for extent range from 0.25 km^2 to 25 km^2 on same conditions. The results are shown in Figure 10, where PF per unit length for each method is divided by the value of BDPST algorithm into a unit PF ratio; Meanwhile, for methods that cannot finish in an average executing time of 10,000 seconds, its unit PF ratio would not be displayed. Accordingly, our proposed BDPST algorithm outperform other comparative methods from 9% to 70% in large scale (e.g. >9 km^2) case and not far away from optimal solutions in small space. Besides, Brute-Force can ensure to obtain optimal solutions but is only feasible for very small ranges (<1 km^2).

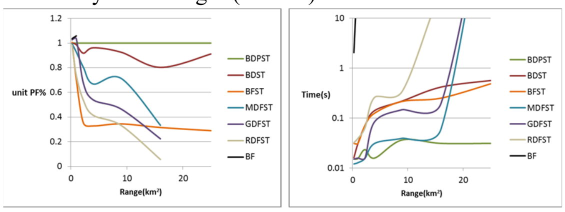

Figure 10: Unit PF% and executing time for different methods.

6 Conclusion

This work proposes an intelligent and interactive system called RPM to let users design novel routes and infer the passenger flows based on routing and ticket data. No existing work addresses the problem. Given heterogeneous features and faced with the competitive and transfer effects of existing routes, our proposed RAR and feature engineering methods are effective for handling dynamic and static data. The experiments on Tainan City bus-ticket data show that our proposed PF inference model and BDPST algorithm outperform baseline and comparative methods. Moreover, the proposed BDPST algorithm is feasible for real-time large scale route recommendation.

REFERENCES

[1] H. Bast et al. (2016). Route Planning in Transportation Networks. *Algorithm Engineering. Lecture Notes in Computer Science*, vol 9220. Springer.

[2] H. Cancela, A. Mauttone and M. E. Urquhart (2015). Mathematical programming formulations for transit network design. Transportation Research Part B: Methodological 77: 17-37.

[3] S. I.-J. Chien, Y. Ding, C. Wei (2002). Dynamic Bus Arrival Time Prediction with Artificial Neural Networks. Journal of Transportation Engineering 128(5): 429-438.

[4] T. M. Cover and J. A. Thomas (1991). Entropy, relative entropy and mutual information. *Elements of Information Theory*, ch. 2, sec. 1, pp. 12-13.

[5] E. W. Dijkstra (1959). A note on two problems in connexion with graphs. Numerische Mathematik, 269–271.

[6] V. Guihaire and J.-K. Hao (2010). Transit network timetabling and vehicle assignment for regulating authorities. Computers & Industrial Engineering 59(1).

[7] Y. Lin, X. Yang, N. Zou and L. Jia (2013). Real-Time Bus Arrival Time Prediction: Case Study for Jinan, China. Journal of Transportation Engineering, 1133-1140.

[8] A. Mauttone and M. E. Urquhart (2009). A route set construction algorithm for the transit network design problem. Computers & Operations Research 36: 2440-2449.

[9] M. Pternea, K. Kepaptsoglou and M. G. Karlaftis (2015). Sustainable urban transit network design. Transportation Research Part A: Policy and Practice 77: 276-291.

[10] L. Quadrifoglio and X. Li (2009). A methodology to derive the critical demand density for designing and operating feeder transit services. Transportation Research Part B: Methodological 43(10): 922-935.

[11] C. Shaowu, L. Baoyi and Z. Botao (2010). Bus arrival time prediction model based on APC data. 6th Advanced Forum on Transportation of China.

[12] L. A. Silman et al. (1974). Planning the route system for urban buses. Computers & Operations Research 1(2): 201-211.

[13] L. Steg and R. Gifford (2005). Sustainable transportation and quality of life. Journal of Transport Geography 13(1): 59-69

[14] H.-M. Su and C.-C. Kuan (2003). Planning and design guidelines. In *Design Manual for Urban Sidewalks*, ch. 4, sec,1, pp. 1-4.

[15] W. Y. Szeto and Y. Wu (2011). A simultaneous bus route design and frequency setting problem for Tin Shui Wai, Hong Kong. European Journal of Operational Research 209(2): 141-155.

[16] Y. Yan, Q. Meng, S. Wang and X. Guo (2012). Robust optimization model of schedule design for a fixed bus route. Transportation Research Part C: Emerging Technologies 25: 113-121.

[17] Y. Wei and M.-C. Chen (2012). Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks. Transportation Research Part C: Emerging Technologies 21: 148-162.

[18] Z. Wenfeng, S. Zhongke and L. Zhiyong (2008). Prediction of urban passenger transport based-on wavelet SVM with quantum-inspired evolutionary algorithm. IEEE IJCNN conference.