# A Novel Approach to Taxi-GPS-Trace-Aware Bus Network Planning

Liangyao Tang[1], Peng Chen[2], Ruilong Yang[1], Yunni Xia[1(✉)], Ning Jiang[3], Yin Li[4], and Hong Xie[1]

[1] School of Computers, Chongqing University, Chongqing 400030, China
[2] School of Computer and Software Engineering, Xihua University, Chengdu 610039, China
[3] Mashang Consumer Finance Co., Ltd. (MSCF), Chongqing 40022, China
[4] Institute of Software Application Technology, Guangzhou and Chinese Academy of Sciences, Guangzhou 511458, China

**Abstract.** Taxi GPS traces are rich with information regarding the human mobility pattern in metropolitans. In this paper, we aimed at fully exploiting the Taxi GPS traces and addressing the bus network planning problem. Specifically, the proposed framework comprises a method for determining candidate bus stations by utilizing passenger pick-up and drop-off records, a bio-inspired method for yielding bus routes and further for generating the final bus network. To prove the effectiveness of our framework, we conduct simulative studies as well based on a real-world taxi GPS data-set and show that our proposed framework considerably outperforms traditional ones.

**Keywords:** Taxi GPS traces · Bus routes planning · Bus network planning

## 1 Introduction

Buses are usually believed to be more energy-efficient, in terms of energy consumed per mile and per person, and less resource requiring than private cars when serving crowded city areas [11]. Nowadays, with the increasing demand for highly-efficient and green public transportation for metropolitan citizens, cost-effective and environment-friendly bus networks shows its great importance in serving versatile transportation demands [15,39].

---

Liangyao Tang and Peng Chen contribute equally to this work and thus are co-first authors of this paper.

This work is supported by China Scholarship Council Science and Technology Program of Sichuan Province under Grant 2020YFG0326, Talent Program of Xihua University under Grant Z202047, the Chongqing grand research and development project under Grant cstc2019jscx-fxyd0385.


H. Gao and X. Wang (Eds.): CollaborateCom 2021, LNICST 406, pp. 520–533, 2021.
https://doi.org/10.1007/978-3-030-92635-9_30

Traditional method for bus network planning and design are usually based on human surveys [2], which requires high cost and manpower but proved to be inaccurate. Moreover, survey-based planning is incapable of accommodating the fast changes and modifications of the metropolitan networks, especially when metropolitan roads are under continuing construction and optimization. Nowadays, with the wide application of wireless communication technology, taxis are usually equipped with GPS devices [16,20,40,42]. Taxi GPS traces are rich with time and spatial information that can be exploited. Such traces comprise time/positions of pick-up and drop-off events, which can be further used for analyzing the emergence patterns of "hot" areas and predicting real-time transportation needs.

In this paper, we propose a novel approach to planning bus routes and networks by exploiting taxi GPS traces. The propose method is capable of identifying candidate bus stops base on historical taxi passenger pick-up/drop-off distributions, appropriately manage "hot" areas for serving as-many-as-possible passengers and yielding bus networks with maximum passenger flows with the constraint of available candidate stops and serving time. We conduct extensive simulations as well to verify the effectiveness of the proposed approach.

## 2 Related Work

Existing works in the direction of exploiting automobile traces fall into multiple categories: social dynamics computing, traffic dynamics mining and operational dynamics computing [5,32,35]. The first category relies itself on analysis of collective behavior and movement of a city's population. Related works aim at analyzing the destination places of citizens [19,23], or the distribution patterns of "hot" spots [6,28], or the functions of "hot" spots [24,30]. The second category studies the flow of the population through the city's road network and aims at forecasting traffic flows and travelling durations for drivers [4,21,25,41]. The third category concerns taxi driver's behavior and aims at learning taxi drivers' expert knowledge [7,26,27,29,38].

Bus network planning can be seem as a extended topic of the trace mining ones [1,36,37,44]. It is known as a complex, nonlinear, non-convex, multi-objective NP-hard problem [3,22,34]. Related research objectives comprises planning and optimization of travelling route, travelling durations, travelling cost and throughput of road networks [17,18,43]. Among them, [9] aims to find an optimal bus route for a given origin-destination (OD) pair in a single direction. Similarly, [8] aims to find a bi-directional travelling route for a specified OD pair.

## 3 Main Steps

### 3.1 Candidate Bus Stop Identification

The first major stage of our proposed method is to decide candidate bus stops through analyzing the taxi PDRs (taxi passenger pick-up and drop-off spots and times). It comprises two steps: (1) Dividing the bus serving area into small equally-sized grid cells and marking "hot" ones for further processing (A hot grid cell refers to the grid with PDRs greater than zero); (2) Selecting candidate bus stops according to the Algorithm 1.

**Hot Grid Cells.** We first split the area under study into equally-sized grid cells, each of which covers a 25 m × 25 m square. Thus, the whole area is partitioned into 1500 × 1800 cells. Note that only reachable cells, excluding unreachable ones in terms of, e.g., lakes and mountains, are with PDR and are thus 'hot' ones. Figure 1 is a sample for hot grid cells.

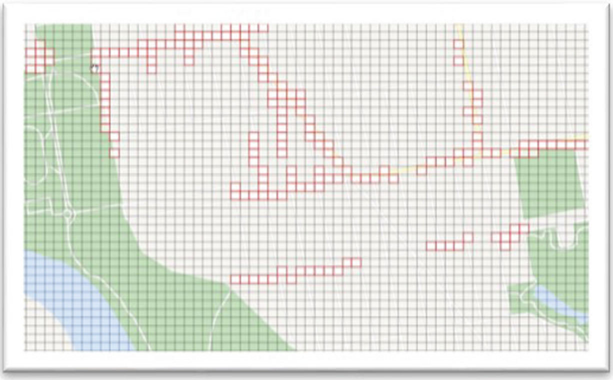


**Fig. 1.** A sample for hot grid cells.

**Selection of Candidate Bus Stops.** Traditional methods for deciding candidate stops are performed through clustering PDRs. However, such methods could lead to inappropriate setting of bus stops especially when some clusters are too large for a single stop to cover. To overcome related limitations, we consider an improved strategy, as shown in Algorithm 1 for deciding candidate bus stations:

(1) Taking each "hot" grid cell as the center and a certain distance as the radius to count the PDRs within the range. The above-mentioned distance refers to the service radius of a bus stop, and the service radius must be set properly because it affects the walking distance to the bus stop. According to the "Urban Road Traffic Planning and Design Code" [31], the service radius of the bus stop in this paper is set to 300 m.
(2) Sorting the "hot" grid cells in descending order according to *pdrs* value (It refers to the number of PDRs within 300 m of each "hot" grid.) and select the grid with the largest *pdrs* value as the first candidate stop. Obviously,

**Algorithm 1:** Candidate Stops Selection

```
Input  : The collection of "hot" grid: G
Output: The collection of candidate stops: S
i = 1 // Initialization
for each grid ∈ G do
    counting the number of PDRs within 300 m of this grid: pdrs
end
repeat
    G = sort(G) // sort G according to pdrs by descending order
    S_i = G_1      // select the grid with the largest pdrs as candidate stop
    Deleting the PDRs within 300 m of the G_1
    for each grid ∈ G do
        update the pdrs of grid
    end
    i = i + 1
until Get enough candidate sites;
```

the larger the *pdrs* value is, the more potential passengers there are around the grid.

(3) Deleting the PDRs within 300 m of the first candidate stop and update the *pdrs* of the remaining "hot" grid cells. This step is mainly to eliminate the impact of the selected candidate sites, because if there are candidate stops around a "hot" grid, the number of potential passengers around the grid will be reduced.

(4) Sorting the remaining "hot" grid cells in descending order according to *pdrs* value, and repeat the above process.

### 3.2 Bus Network Generation

Taking the bus stations generated above as inputs, the following stage aims at planning bus networks with 2 steps: (1) Estimating the user flow and trip time with any two candidate stops according to trace data. (2) Applying an ant colony algorithm for yielding bus routes and further generating bus networks.

**Passenger Flow and Travel Time Evaluation.** We employ two matrix, i.e., flow matrix (FM) and travelling time matrix (TM), for capturing travel demand and related time requirement. Every element in the matrix indicates the number of passengers or the required travelling time from one station to another.

In the FM matrix, we count the total passenger flow from the coverage of one stop to that of another. We further calculate the average taxi time between two candidate stops. In addition, in consideration of the speed gap between taxis and buses, we consider the bus travel time between two candidate stops to be the average taxi time multiplied by $\alpha$ ($\alpha = 1.5$). For the paths without taxi trip records, we consider using $dis(i, j)/v$ as an approximation. $dis(i, j)$ is the driving distance between station $i$ and station $j$, and $v$ is 25 km/h due to the fact that the average speed of taxis is about 25.4 km/h according to taxi GPS traces.

**Bus Route Generation.** Ant colony algorithm is a general-purpose heuristic algorithm which can be used to solve different combinatorial optimization problems [10,12,13,33]. As illustrated in Algorithm 2, we use ant colony algorithm with elitist strategy for selecting bus routes. This algorithm takes the following intermediate variables as inputs:

(1) Visibility. We define the visibility $\eta_{ij}$ as passenger flow between station $i$ and station $j$. $fm(i,j)$ refers to the passenger flow from station $i$ to station $j$

$$\eta_{ij} = fm(i,j) + fm(j,i) \tag{1}$$

(2) Pheromone update. We update the pheromone $\tau_{ij}$ from station $i$ to station $j$ according to the following formula. $\rho$ is pheromone evaporation coefficient, $\Delta\tau_{ij}$ the pheromone increment, $m$ the number of ants, $k$ the $k$-th ant and $Q$ the pheromone constant.

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \Delta\tau_{ij} \tag{2}$$

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{3}$$

$$\Delta\tau_{ij}^{k} = \begin{cases} Q\cdot Num, & if\ the\ k\text{-}th\ ant\ passes\ edge(i,j) \\ 0, & otherwise \end{cases} \tag{4}$$

(3) Passenger flow. $Num$ represents the passenger flow in both directions of the route. For route $R = [s_1, s_2, ...s_n]$, $Num$ is determined according to the following formula:

$$Num = \sum_{i;j(j>i)}^{n} (fm(s_i,s_j) + fm(s_j,s_i)) \tag{5}$$

(4) Transition probability. Ants randomly select the next stop according to a selection probability affected by visibility and pheromone. We use *allowed* to represent the set of optional next stations and define the transition probability from station $i$ to station $j$ for the $k$-th ant as:

$$p_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}\cdot[\eta_{ij}]^{\beta}}{\sum_{k\in allowed}[\tau_{ik}]^{\alpha}\cdot[\eta_{ik}]^{\beta}}\ ,\ if\ j\in allowed \tag{6}$$

(5) Time of current route. $T$ is the total time of the current route in both directions. If $T$ satisfies the time constraint, the corresponding ant continues to search for the next station, otherwise, it stops searching. For current route $R = [s_1, s_2, ...s_n]$, we calculate the $T$ according to the following formula. $t_0$ is the waiting time of the bus at each stop:

$$T = \sum_{i=1}^{n-1}(tm(s_i,s_{i+1}) + tm(s_{i+1},s_i)) + (n-2)\cdot t_0\cdot 2 \tag{7}$$

---

**Algorithm 2:** Bus Route Selection

---

**Input** : $FM$: Passenger flow matrix
$TM$: Travel time matrix
$iter_{max}$: Maximum number of iterations
$T_{max}$: Time constraints
**Output**: *Route*

```
Initialize the parameters of ant colony algorithm.
Initialize the position of the ants.
count = 0 // Initialize the number of iterations.
while count < iter_max do
    for each ant do
        repeat
            Choose the next stop j according to Eq. 6.
            Calculate the total time T according to Eq. 7.
        until T > T_max;
        Calculate the passenger flow of the route according to Eq. 5.
        Update the best route.
    end
    for each edge(i, j) do
        Update pheromone according to Eq. 2–Eq. 4.
    end
    count = count + 1
end
```

---

**Algorithm 3:** Bus Network Planning

---

**Input** : $FM$: Passenger flow matrix
$TM$: Travel time matrix
$Num_{min}$: Minimum passenger flow
**Output**: The collection of routes $\{R_1, R_2 ... R_n\}$

```
k = 1 // Initialization
repeat
    Run Bus Route Selection Algorithm, and we get R_k.
    //R_k is a route generated by Algorithm 2.
    Calculate Num (the passenger flow of the route R_k) according to Eq. 5.
    for i in R_k do
        for j in R_k do
            fm(i, j) = 0 //fm(i, j) is the passenger flow from stop i to stop j.
        end
    end
    k = k + 1
until Num < Num_min;
```

---

**Bus Network Generation.** Algorithm 3 shows the process of bus network generation. It first generates one single bus route using the Bus Route Selection algorithm and then updates passenger flow matrix (FM). For example, we firstly get a route $R_k$ by Algorithm 2. Then, we assume that the passenger flow between the stations of route $R_k$ becomes zero. Finally, we update FM and get a new route based on the new FM.

## 4 Simulations

In this part, we test our proposed approach with a third-party and real-world taxi GPS data-set. The data-set was generated from taxis in New York City for one week and comprises more than 3.4 million passenger delivering trips formatted as follows:

– Trip_Pickup_Datetime : passenger boarding time
– Trip_Dropoff_Datetime : passenger drop-off time
– Start_Lon : longitude of the place where passengers get on
– Start_Lat : latitude of the place where passengers get on
– End_Lat : longitude of the place where passengers get off
– End_Lat : latitude of the place where passengers get off
– Trip_Distance : total travel distance

| | Trip_Pickup_DateTime | Trip_Dropoff_DateTime | Start_Lon | Start_Lat | End_Lon | End_Lat | Trip_Distance |
|---|---|---|---|---|---|---|---|
| **2** | 2009-12-18 03:09:00 | 2009-12-18 03:34:00 | -73.955745 | 40.689503 | -73.937730 | 40.737463 | 14.451909 |
| **15** | 2009-12-19 22:05:00 | 2009-12-19 22:18:00 | -74.026760 | 40.656373 | -74.030567 | 40.658013 | 1.271382 |
| **17** | 2009-12-19 23:05:00 | 2009-12-19 23:38:00 | -74.005453 | 40.740142 | -73.967598 | 40.753637 | 5.761452 |
| **19** | 2009-12-16 01:31:00 | 2009-12-16 01:47:00 | -74.003327 | 40.751415 | -73.990005 | 40.690455 | 8.095000 |
| **24** | 2009-12-16 01:30:00 | 2009-12-16 01:51:00 | -73.981385 | 40.744072 | -73.966352 | 40.690532 | 12.665537 |
| **...** | ... | ... | ... | ... | ... | ... | ... |

**Fig. 2.** A sample for data set.

Based on the data-set, we further:

(1) Delete duplicate data and unreasonable data, e.g., the Trip_Distance with values of 0.
(2) Delete invalid data, for example, the place where passengers get on/off the taxi is not in the New York.
(3) Project the longitude and latitude coordinates into plane coordinates according to the UTM (Universal transverse Mercator) projection. It divides the earth's surface into 60 projection zones and New York is located in No.18 projection zone. Since New York is located in the northern hemisphere, the FalseEasting is set to 500 km and FalseNorthing is 0.

We generate 200 stations based on Algorithm 1 and Fig. 3 shows the number of PDRs within the 300 m of these stations. It can be seen from Fig. 3 that the stations ahead contain more PDRs. This indicates that there are more potential passengers at the stations ahead. Among them, the station No.0 contains 43825 PDRs, while the station No.199 only contains 1019 PDRs.

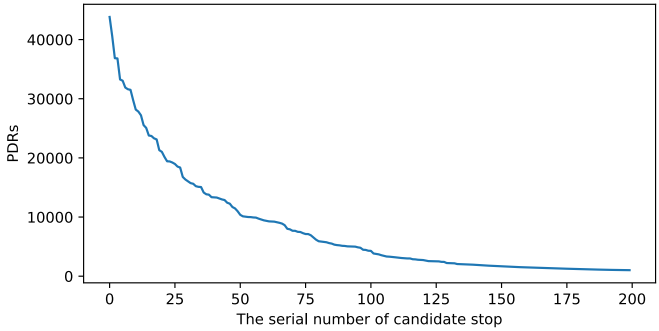



**Fig. 3.** The number of PDRs within the 300 m of each station.

As shown in the Fig. 4, the PDRs covered by the top 80 stations accounts for 74.29% of the total PDRs, and the PDRs covered by the top 100 stations accounts for 80.12% of the total PDRs. Therefore, we decide to select the top 100 stations as candidate stations.

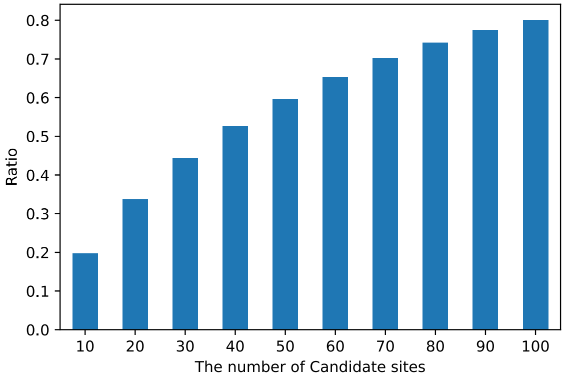
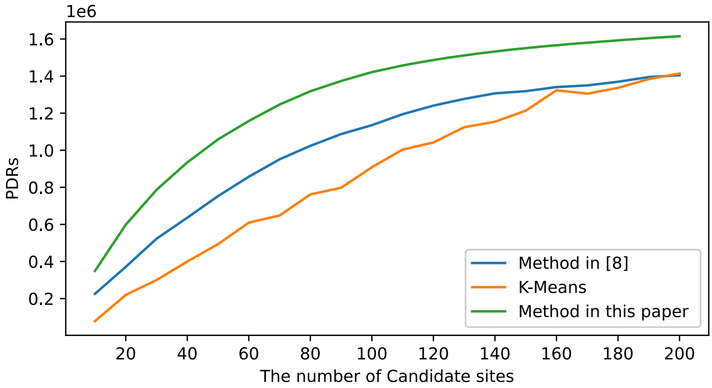


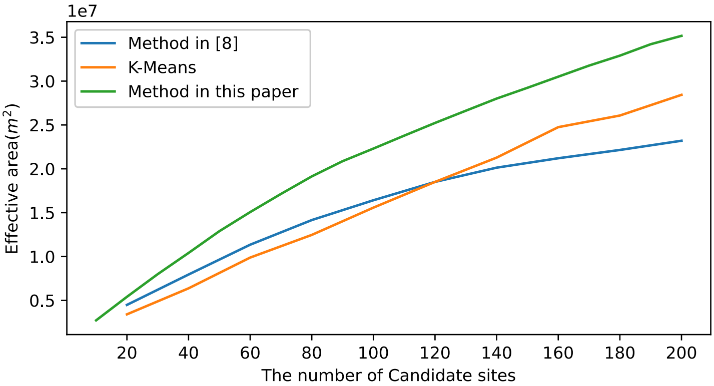

**Fig. 4.** Proportion of the PDRs covered by top $n$ stops.

In order to compare our method with the baseline methods [8,14], we propose two strategies to evaluate the rationality of candidate sites:

(1) We compare the number of PDRs covered by the same number of candidate sites. Obviously, with the same number of candidate stations, the more PDRs the candidate stations contain, the more potential passengers there are.
(2) We compare the effective areas covered by the same number of candidate sites. The effective area refers to the regions with taxi passengers pick-up/drop-off records. In this paper, we use the number of "hot" grids multiplied by the area of the grid to approximately represent the size of the effective area. Similarly, when the number of candidate stations is the same, the candidate stations with more effective areas are better.




(a) Comparison of PDRs




(b) Comparison of effective area

**Fig. 5.** Comparison of candidate stops selection methods.

Figure 5 shows the comparison results between our method and baseline methods [8,14], when the number of candidate site is set to 10, 20... 200 respectively. As illustrated in Fig. 5, in the case of the same number of candidate sites, the candidate sites determined by our method can cover more PDRs, which means more potential passengers. Similarly, the candidate sites determined by our method can contain more effective areas, indicating that the stations are more reasonable. Therefore, our method is superior to the baseline methods in many aspects.

In the stage of bus route selection, the methods in [8,9] generate an optimal route for a specified origin-destination (OD) pair. However, our method does not need to specify the origin-destination pair in advance, and can generate a global optimal route.

Parameter-setting of the Bus Route Selection algorithm is shown in Table 1 and the bus network generated in the experiment is shown in Table 2.

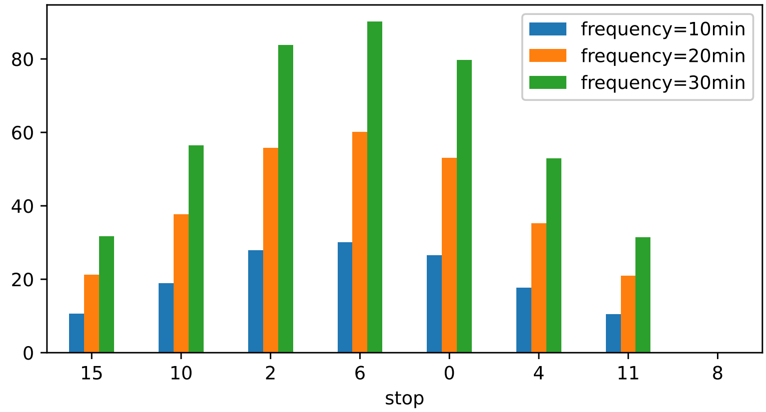
**Table 1.** Experimental parameters

| Parameter | Description | Value |
|---|---|---|
| stop_num | The number of candidate stops | 100 |
| ant_num | The number of ants | 500 |
| $\alpha$ | Pheromone weight value | 1 |
| $\beta$ | Visibility weight value | 1.5 |
| $\rho$ | Pheromone evaporation rate | 0.5 |
| $Q$ | Pheromone constant | 1 |
| $e$ | The number of elite ants | 10 |
| $t$ | Time constraints | 7200 s |

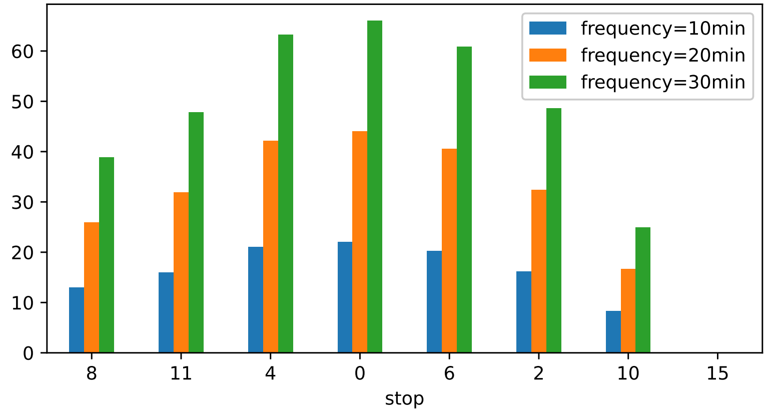
**Table 2.** Bus network

| | Path | Bus frequency | Bus type | Passenger flow |
|---|---|---|---|---|
| 1 | [15, 10, 2, 6, 0, 4, 11, 8] | 20 min | Medium-sized | 18652 |
| 2 | [1, 0, 18, 5, 7, 9, 11, 13] | 20 min | Medium-sized | 15271 |
| 3 | [12, 3, 17, 27, 7, 22, 16, 5] | 20 min | Medium-sized | 12702 |
| 4 | [23, 20, 4, 13, 24, 8, 28, 25, 21] | 20 min | Medium-sized | 12308 |
| 5 | [44, 25, 40, 8, 29, 4, 14, 0, 19] | 20 min | Small-sized | 9598 |
| 6 | [30, 6, 33, 1, 34, 26, 5, 14] | 20 min | Small-sized | 8762 |
| 7 | [35, 12, 9, 26, 0, 24, 11, 17] | 20 min | Small-sized | 7806 |

Next, according to the number of users on the bus at each stop in both directions, we determine the operation frequency and bus capacity for each route in Table 2. For example, for *path*: [15, 10, 2, 6, 0, 4, 11, 8], Fig. 6 illustrates the number of citizens on the bus at each stop, when bus operation frequency is respectively set to 10 min, 20 min, 30 min. Considering the waiting time of passengers, the cost of public transportation system and the capacity of buses, for this path, we set the bus operation frequency to 20 min and choose the medium-sized bus for transportation.




(a) Station 15 to station 8




(b) Station 8 to station 15

**Fig. 6.** The number of passengers on the bus at each stop.

Finally, we compare Bus Route Selection algorithm with the baseline methods [8,9]. As illustrated in Fig. 7, our method achieves higher passenger flow, i.e., the maximum number of passengers being served, than the baseline methods [8,9] at varying time constraints.

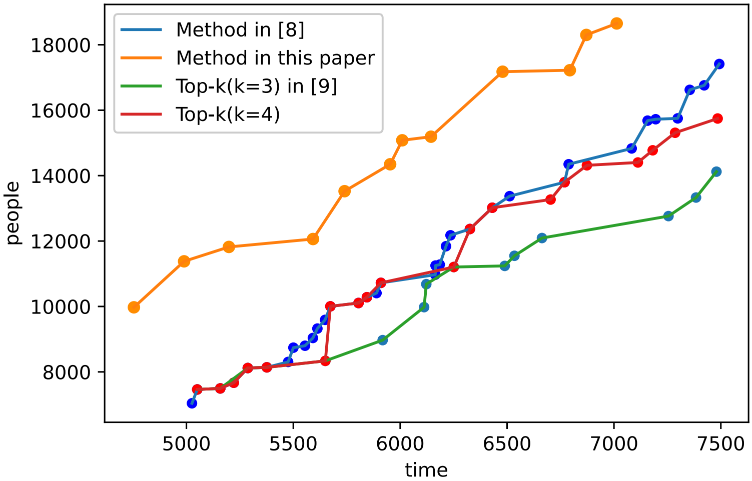



**Fig. 7.** Comparison of bus route selection methods.

## 5 Conclusion

In this paper, we propose a novel framework for generating bus routes and networks through exploiting the taxi GPS traces. The proposed framework comprises a method for determining candidate bus stations according to taxi passenger pick-up and drop-off records and a bio-inspired method for yielding bus routes and networks with adaptive strategies for deciding bus frequency and capacity. With a real-world data set of 3.4 million passenger delivery trips, we conduct simulations and show that our proposed framework clearly outperform baseline methods in terms of the number of people served.

## References

1. Anagnostopoulos, C., Anagnostopoulos, I., Loumos, V., Kayafas, E.: A license plate-recognition algorithm for intelligent transportation system applications. IEEE Trans. Intell. Transp. Syst. **7**, 377–392 (2006)
2. Aslam, J., Lim, S., Pan, X., Rus, D.: City-scale traffic estimation from a roving sensor network. In: SenSys 2012 (2012)
3. Baaj, M.H., Mahmassani, H.: Trust: a lisp program for the analysis of transit route configurations. Transp. Res. Rec. **1283**(1990), 125–135 (1990)
4. Balan, R., Nguyen, K., Jiang, L.: Real-time trip information service for a large taxi fleet. In: MobiSys 2011 (2011)
5. Castro, P.S., Zhang, D., Chen, C., Li, S., Pan, G.: From taxi GPS traces to social and community dynamics. ACM Comput. Surv. (CSUR) **46**, 1–34 (2013)
6. Chang, H.W., Tai, Y.C., Hsu, J.Y.J.: Context-aware taxi demand hotspots prediction. Int. J. Bus. Intell. Data Min. **5**, 3–18 (2010)
7. Chen, C., et al.: iBOAT: Isolation-based online anomalous trajectory detection. IEEE Trans. Intell. Transp. Syst. **14**, 806–818 (2013)

8. Chen, C., Zhang, D., Li, N., Zhou, Z.: B-planner: planning bidirectional night bus routes using large-scale taxi GPS traces. IEEE Trans. Intell. Transp. Syst. **15**, 1451–1465 (2014)
9. Chen, C., Zhang, D., Zhou, Z., Li, N., Atmaca, T., Li, S.: B-planner: night bus route planning using large-scale taxi gps traces. In: 2013 IEEE International Conference on Pervasive Computing and Communications (PerCom), pp. 225–233 (2013)
10. Deng, W., Xu, J., Zhao, H.: An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem. IEEE Access **7**, 20281–20292 (2019)
11. Dickens, M., Neff, J.: Apta 2011 Public Transportation Fact Book (2011)
12. Dorigo, M., Gambardella, L.: Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput. **1**, 53–66 (1997)
13. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst., Man, Cyber., Part B, Cyber. **26**(1), 29–41 (1996)
14. Hartigan, J., Wong, M.: A k-means clustering algorithm (1979)
15. He, Q., et al.: A game-theoretical approach for user allocation in edge computing environment. IEEE Trans. Parallel Distrib. Syst. **31**, 515–529 (2020)
16. He, Q., et al.: A game-theoretical approach for mitigatingedge ddos attack. IEEE Trans. Dependable Sec. Comput. (2021)
17. Jerby, S., Ceder, A.: Optimal routing design for shuttle bus service. Transp. Res. Rec. **1971**, 14–22 (2006)
18. Jiang, C., Bhat, C., Lam, W.K.: A bibliometric overview of transportation research part b: methodological in the past forty years (1979–2019). Transp. Res. Part B-Method. **138**, 268–291 (2020)
19. Li, B., et al.: Hunting or waiting? discovering passenger-finding strategies from a large-scale real-world taxi dataset. In: 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 63–68 (2011)
20. Li, B., He, Q., Chen, F., Jin, H., Xiang, Y., Yang, Y.: Auditing cache data integrity in the edge computing environment. IEEE Trans. Parallel Distrib. Syst. **32**(5), 1210–1223 (2020)
21. Li, X., et al.: Prediction of urban human mobility using large-scale taxi traces and its applications. Front. Comput. Sci. **6**, 111–121 (2011)
22. Liu, C.L., Pai, T., Chang, C.T., Hsieh, C.M.: Path-planning algorithms for public transportation systems. In: ITSC 2001. 2001 IEEE Intelligent Transportation Systems, Proceedings (Cat. No.01TH8585), pp. 1061–1066 (2001)
23. Liu, L., Andris, C., Ratti, C.: Uncovering cabdrivers' behavior patterns from their digital traces. Comput. Environ. Urban Syst. **34**, 541–548 (2010)
24. Liu, Y., Wang, F., Xiao, Y., Gao, S.: Urban land uses and traffic 'source-sink areas': evidence from GPS-enabled taxi data in shanghai. Landscape Urban Plan. **106**, 73–87 (2012)
25. Lu, H., Jin, L., Luo, X., Liao, B., Guo, D., Xiao, L.: RNN for solving perturbed time-varying underdetermined linear system with double bound limits on residual errors and state variables. IEEE Trans. Ind. Inf. **15**(11), 5931–5942 (2019)
26. Luo, X., Liu, H., Gou, G., Xia, Y., Zhu, Q.: A parallel matrix factorization based recommender by alternating stochastic gradient decent. Eng. Appl. Artif. Intell. **25**(7), 1403–1412 (2012)
27. Luo, X., Wang, D., Zhou, M., Yuan, H.: Latent factor-based recommenders relying on extended stochastic gradient descent algorithms. IEEE Trans. Syst., Man, Cyber. Syst. **51**(2), 916–926 (2019)

28. Luo, X., Xia, Y., Zhu, Q., Li, Y.: Boosting the k-nearest-neighborhood based incremental collaborative filtering. Knowl.-Based Syst. **53**, 90–99 (2013)
29. Luo, X., Zhou, M., Li, S., Wu, D., Liu, Z., Shang, M.: Algorithms of unconstrained non-negative latent factor analysis for recommender systems. IEEE Trans. Big Data **7**(1), 227–240 (2019)
30. Pan, G., Qi, G., Wu, Z., Zhang, D., Li, S.: Land-use classification using taxi GPS traces. IEEE Trans. Intell. Transp. Syst. **14**, 113–123 (2013)
31. Shanqing, Z.: Rationality and values of relevant indexes in "code for transport planning on urban road"(gb 50220–95). Transport Standardization (2011)
32. Shen, Y., Zhao, L., Fan, J.: Analysis and visualization for hot spot based route recommendation using short-dated taxi GPS traces. Information **6**, 134–151 (2015)
33. Stützle, T., Hoos, H.: Max-min ant system. Future Gener. Comput. Syst. **16**, 889–914 (2000)
34. Szeto, W.Y., Wu, Y.: A simultaneous bus route design and frequency setting problem for Tin Shui Wai, Hong Kong. Eur. J. Oper. Res. **209**, 141–155 (2011)
35. Tang, J., Liu, F., Wang, Y., Wang, H.: Uncovering urban human mobility from large scale taxi GPS data. Phys. A-stat. Mech. Appl. **438**, 140–153 (2015)
36. Wang, F.: Driving into the future with its. IEEE Intell. Syst. **21**, 94–95 (2006)
37. Wang, F.: Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications. IEEE Trans. Intell. Transp. Syst. **11**, 630–638 (2010)
38. Wu, D., Luo, X., Shang, M., He, Y., Wang, G., Zhou, M.: A deep latent factor model for high-dimensional and sparse matrices in recommender systems. IEEE Trans. Syst., Man, Cyber. Syst. **51**(7), 4285–4296 (2019)
39. Xia, X., Chen, F., He, Q., Grundy, J., Abdelrazek, M., Jin, H.: Cost-effective app data distribution in edge computing. IEEE Trans. Parallel Distrib. Syst. **32**, 31–44 (2021)
40. Xia, X., Chen, F., He, Q., Grundy, J., Abdelrazek, M., Jin, H.: Online collaborative data caching in edge computing. IEEE Trans. Parallel Distrib. Syst. **32**, 281–294 (2021)
41. Yuan, J., Zheng, Y., Xie, X., Sun, G.: T-drive: enhancing driving directions with taxi drivers' intelligence. IEEE Trans. Knowl. Data Eng. **25**, 220–232 (2013)
42. Yuan, L., et al.: Coopedge: a decentralized blockchain-based platform for cooperative edge computing. In: Proceedings of the Web Conference 2021 (2021)
43. Zhao, F., Zeng, X.: Optimization of transit route network, vehicle headways and timetables for large-scale transit networks. Eur. J. Oper. Res. **186**, 841–855 (2008)
44. Zhu, L., Yu, F., Wang, Y., Ning, B., Tang, T.: Big data analytics in intelligent transportation systems: a survey. IEEE Trans. Intell. Transp. Syst. **20**, 383–398 (2019)