StaCover: Mobile Energy-sharing Cabinets Deployment with Public Bike System

Siwen Zheng*, Jianhui Zhang*†, Jiayu Gan*, Tianhao Zhang*
*College of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018 China
†Corresponding author email: jhzhang@ieee.org

***Abstract*—To enhance environmental sustainability and fulfill metropolitan public traffic demands, some metropolises are developing a kind of new Bike-Energy System (BES) including the existing Public Bike System (PBS) and Mobile Energy-sharing System (MES), which consists of some batteries-loaded cabinets. Cabinet location selection is expected to facilitate users to access battery in nearby cabinet. Meanwhile, the limited budget cannot afford establish a cabinet for each station. It thus brings up a problem: which bike stations should be selected to deploy the cabinets so as to minimize the deployment cost and satisfy that all users can rent and return battery conveniently to at least one nearby bike station on the way riding. This paper formulates the bike station selection problem as the *Set Cover Problem* and proposes a novel data-driven method—*StaCover* accordingly. *StaCover* presents the Density-Based Stations Clustering algorithm (DBSC) to select the candidate stations and then designs a Greedy Heuristic Selection algorithm (GHS) to determine the final stations to deploy the energy-sharing cabinets. Our experiments adopt diversified parameters to demonstrate the effectiveness of *StaCover* over the other methods.**

***Index Terms*—BES; Cluster; *StaCover*; Set cover**

I. Introduction

In recent years, we have witnessed the prevalence of Public Bike Systems (PBSs) from Chicago of USA to Hangzhou of China, which offer massive short-distance bike rental service with the ubiquitous bike stations in urban area [1] [2]. The attractive contribution of PBS is to provide an environment-friendly and convenient solution to the first-and-last mile transportation problem [3] [4]. To mitigate the pressure on the long-distance transport modes, Hangzhou Public Transport Company (HPTC) tries to explore the PBS deeply since 80% bike trips last no more than $3km$ [5]. Another work sums up that the taxi's average trip distance is 8.86 km [6]. If the bike trip distance can be extended to the middle distance such as 8 km, the air pollution caused by the vehicle emission can be decreased much. The main drawback for short bike trips is that most users have limited physical strength. To overwhelm it, a feasible way is to supply bike with extra power. HPTC thus is trying to combine the PBS with the Mobile Energy-sharing System (MES) to form a new PBS. We call it the Bike Energy System (BES). An example of the BES station is demonstrated in Figure 1.

In BES, each bike is embedded with a battery box and some intelligent sensors. If a fully-charged battery is inserted into the box, the bike can support user to extend his/her trip to middle distance. Since there usually are multiple bikes in each station of BES, each cabinet is suitable to contain multiple batteries.

(a) A bike station of PBS (b) A cabinet of MES

Fig. 1. An example of BES station

BES allows users to rent battery from the cabinets of MES and insert it into the battery-box under the basket of the bike. As the complete new transport mode, BES can provide a hybrid service with human riding and electric riding.

To do these, there are some detailed challenges to build BES. It will take great cost to build and maintain such one cabinet. To build one cabinet for every bike station is impossible because the system would be very huge and the budget is limited. Each of users should have at least one energy-sharing cabinet to rent a battery on the way riding and return the rental battery after arriving at the target location. Therefore, the challenging problem is how to select bike stations to deploy cabinets so as to minimize the deployment cost and to satisfy user demand. The detailed demands and realistic constraints are listed below:

- **Supporting middle-distance riding.** The goal of BES is to extend the ability of the PBS from supporting the short distance riding to the middle distance one. It can alleviate the pressure on other long-distance transport systems.
- **User demand.** Each user can have available station to rent and return battery in his/her acceptable distance range.
- **Limited budget.** Due to the massive amount of cabinets in the MES, the cost for their deployment must be very high and thus should be controlled.

Hangzhou has the largest PBS in China, with an average daily visitings of more than 30 million [7]. The stations selected for energy-sharing cabinets deployment is expected to increase the battery availability for each user under the high visiting frequency. By analyzing the metropolitan-scale datasets from HPTC, this paper extracts two attributes, station visiting frequency and resources balance deviation, for each bike station. With them, we formulate the station selection problem as the well-known *set cover problem* [8] [9]. The station selection problem aims to minimize the overall deployment cost while satisfying users demand.

This paper proposes the method *StaCover* to handle this problem. Firstly, *StaCover* uses the Density-based Station Clustering algorithm (DBSC) to select out the candidate bike stations based on the two station attributes observed from dataset analysis. It then incorporates a Greedy Heuristic Selection algorithm (GHS) to determine the stations which should be deployed with energy-sharing cabinets. *As far as we know, our work is the first to handle this problem.*

II. Bike Dataset Analysis and Problem Formulation

A. Dataset Analysis

The datasets provided by HPTC consist of three sub-datasets: the station dataset, the berth dataset and the lease record dataset of three months from April 1 to June 30, 2016. They contain the information of the PBS and users' riding records in this system in Hangzhou City. The first one records the IDs and addresses of 1773 bike stations. The second one records the timestamped numbers of available berths and bikes in each station. The third one records all trips' information including user IDs, leasing and returning time, leasing and returning station IDs.

From the above three datasets, we extract two attributes: (1) resources balance deviation, denoted by r, (2) bike visiting frequency, denoted by f. The station i's attributes can be denoted by $a_i(r_i, f_i)$, and calculated as follows.

Resources balance deviation. Each bike station has two kinds of resources: bikes and berths, which indicate the ability of the bike station for bike renting and returning respectively. The resources balance deviation describes the balance state between the two abilities at time t of each station, and is given by the ratio function in Equation (1):

$$r'^t_i = \begin{cases} \frac{b^t_i+1}{e^t_i+1} & \text{if } b^t_i \le e^t_i, i \in V \\ \\ \frac{e^t_i+1}{b^t_i+1} & \text{if } b^t_i > e^t_i, i \in V \end{cases} \tag{1}$$

where b^t_i and e^t_i are the numbers of bikes and berths respectively in station i at time t. V denotes the set of all bike stations. The ratio function guarantees the value of r'^t_i is no more than 1. It can avoid the 0 value of the denominator to add numerator and denominator with 1. If the value of r'^t_i equals 1, station i has the best balance state. The *standard deviation* is adopted to measure the resources balance state for each station during one period by Equation (2) [10] :

$$r_i = \sqrt{\frac{\sum_{t=1}^{T}(1 - r'^t_i)^2}{T}} \tag{2}$$

where T denotes the running time of the PBS for one period, such as 24 hours.

Public visiting frequency. If the bike station with higher public visiting frequency is selected to deploy energy-sharing cabinets, the cabinets are more likely to be used. This attribute thus is quite suitable for the bike station selection.

Figure 2 shows the Probability Mass Function (PMF) and Cumulative Distribution Function (CDF) of the two attributes [11]. By integrating the two attributes into consideration, the stations with both low resources balance deviation ($r < 1$) and high public visiting frequency ($f > 200$/day) take up a small portion within the square in Figure 3. The above observations motivate us to find an effective method to properly extract candidate stations to deploy energy-sharing cabinets by considering the two attributes and their distribution in different regions. The details will be presented in Section III-B.

B. Preliminary

In the PBS, some stations will be selected out to deploy energy-sharing cabinets while others are not. We call the former as e-station and the later as ne-station. Let D denote the set of all e-stations and thus D is a subset of V. Let c_i denote the cost to deploy a cabinet at station i.

Quality. By considering the goals of bike station selection problem described above, we introduce a metric, quality, to measure how good a bike station is selected to deploy an energy-sharing cabinet. The quality function $Q(i)$ is defined by Equation (3):

$$Q(i) = \frac{|M^d_i|}{c_i d} \tag{3}$$

in which M^d_i is the station set covered by station i. It can be defined as Equation (4):

$$M^d_i = \{j : l_{i,j} \le d, i, j \in V (i \ne j)\} \tag{4}$$

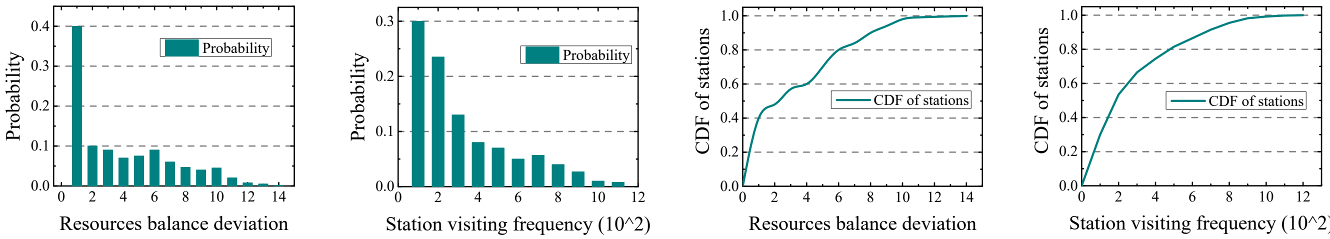
where cover radius, denoted by d (*e.g.*, 3 km), presents the maximum distance that users are able to ride for the battery. $|M^d_i|$ is a cardinality indicator. $l_{i,j}$ is the riding distance between station i and j. Note that i and j cannot be the same station. $\frac{|M^d_i|}{c_i}$ reflects station i's cover ability with unit cost. *StaCover* adopts $Q(i)$ to select the stations.

Coverage requirement. Because of the limited budget, only partial bike stations can be selected to build cabinets. Users may not find cabinet at each bike station, but can reach at least one nearby e-station within their acceptable riding distance d km. So we apply Equation (5) to ensure that all ne-stations in V are covered by at least one e-station:

$$| M^d_i \cap D | \ge 1, \quad \forall i \in V \setminus D \tag{5}$$

C. Problem Formulation

To select the set D of the bike stations to deploy energy-sharing cabinets, two factors should be taken into account. Firstly, the cost for deployment, *i.e.*, $\sum_{i \in D} c_i$, should be decreased. Secondly, each user must have at least one e-station to return the battery after arriving the target station in the maximal riding distance d km. So the coverage requirement



(a) PMF: resources balance deviation (b) PMF: station visiting frequency (c) CDF: resources balance deviation (d) CDF: station visiting frequency

Fig. 2. Distribution of the two stations attributes.

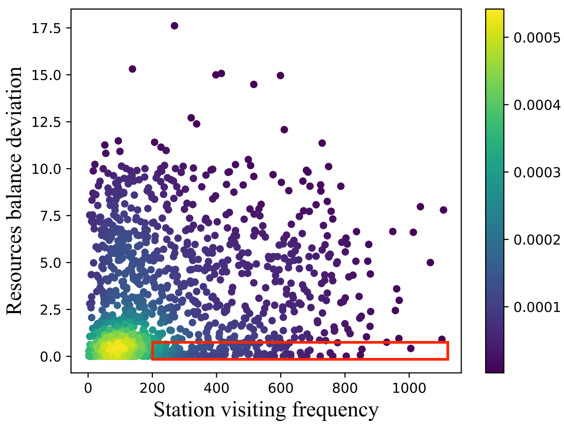


Fig. 3. Density scatter of the two station attributes.

in Section II-B should be satisfied. Our data-driven method *StaCover* aims to discover the subset $D \in V$, which follows these two factors. *StaCover* needs to clean up isolated stations in set V and generate a new set V'. Then it extracts a subset H from V' as the candidate set. The set H should satisfy the following constraint:

$$V' = \bigcup_{i \in H} M_i^d \tag{6}$$

where H is called a set cover of V'. Now, the problem becomes how to find the subset of $D \subset H$ $(H \subsetneq V')$ for the energy-sharing cabinets deployment. Formally, it can be represented as the *Set Cover Problem* [8] [9]:

$$\min \sum_{i \in H} c_i x_i \tag{7}$$

$$s.t. \sum_{i:j \in M_i^d} x_i \geq 1, \forall j \in V' \tag{8}$$

$$x_i = 0, 1, \quad \forall i \in H \tag{9}$$

Equation (7) indicates our objective to minimize the total deployment cost of the stations in D. Equation (8) guarantees that every station in the set V is covered by at least one e-station. In Equation (8), the binary integer x_i equals to 1 if the station i is selected for deploying energy-sharing cabinets, and 0 otherwise. We need solve this problem and get the set D that contains all stations with $x = 1$. Such a *Binary Integer Programming* (BIP) problem is proved to be NP-hard [12] [13].

III. StaCover Design

A. Method Overview

As shown in Figure 4, our method *StaCover* consists of three main stages as follows:

- **Pre-processing.** This stage performs the following two steps to prepare the data for further processing: (1) Dataset parsing, which cleans the isolated stations (do not have any neighbor station within d km) from set V to generate V', (2) Map matching, which projects the station onto corresponding coordinate in Mapbox [14].
- **Candidate station extraction.** (Section III-B). After data pre-processing, *StaCover* first clusters stations with the two attributes by DBSC and then use sorting approach to extract the stations with both high station visiting frequency and small resources balance deviation.
- **Deployment station determination.** (Section III-C). *StaCover* proposes the GHS to find an approximate solution for the previous problem formulated in Section II-C and outputs the e-stations set D as the result.

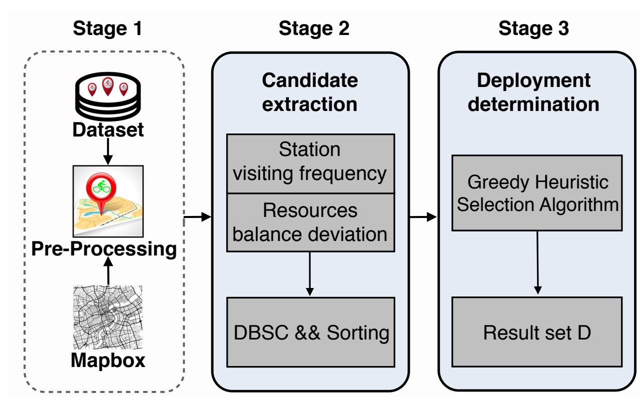


Fig. 4. An overview of *StaCover*

B. Candidate Station Extraction

The territory of Hangzhou consists of 6 functional regions (*e.g.*, commercial-area, residential-area). By observing the visiting frequency of PBS stations for a period of time, we find that the records of station visiting frequency in the same region tend to be similar and have a significant differences between different regions. For example, the station visiting frequency in downtown is generally much higher than that in the industrial region. However, the station selection for deploying energy-sharing cabinets still needs to consider the energy demands in the industrial area as well in order to support the BES operation covering the whole city. For this purpose, *StaCover* clusters stations with similar attributes to one group by DBSC and further extracts the stations with the better attribute values in each group. The extracted candidate are the ones with high suitability for energy-sharing cabinets deployment. Geographically, they are distributed in different functional regions. Specifically, our work first shows the normalization of station attributes, and then introduces how to cluster the stations by the DBSC. Finally, it illustrates the process of extracting candidate by referring to the stations attributes' sorting.

Station attributes normalization. We leverage the *0-1 normalization*, which is a linear transformation of the original attributes data, leaving the result in $[0, 1]$ interval [15]. All station attributes should be normalized before clustering.

Clustering bike stations. After the normalization, *StaCover* refers to Spatial Clustering and propose the DBSC to group stations in set V' into k clusters. The value of k is decided by the DBSC. Each bike station now is described with the two normalized attributes like point $a_1(0.5, 1)$. The station attributes set $A = \{a_i\}, i \in V'$ is defined to store stations' normalized attributes value. In the DBSC, there are three different types of points in the set A: *core point, border point* and *noise point*. There are two parameters, neighborhood radius γ and threshold α that we need to adjust. As a baseline, α is used to measure whether a point is a core point. Let $N_{a_i}^{\gamma}$ denote the set of points in the range γ centered at a_i. Every point a_i is associated with a density $\rho(a_i)$ which defined as:

$$\rho(a_i) = |N_{a_i}^{\gamma}|, a_i \in A \tag{10}$$

A core point has a large density in their neighborhood. It is defined as follow:

$$\rho(a_i) \geq \alpha, a_i \in A \tag{11}$$

All core points form the set A_c. If a point a_i is not a core point, but a neighborhood of the core point, it is called border point and described as:

$$A_c \cap N_{a_i}^{\gamma} \neq \varnothing, \quad a_i \in A \setminus A_c \tag{12}$$

We collect all border points into a set A_b. If a point a_i is not in both A_c and A_b, it is called noise point. Figure 5 (a) shows the three kinds of points when $\alpha = 2$. The circle covers the range with the radius γ centered at its corresponding station. It represents the maximum range in which each user is willing to find cabinets. There are three concepts in the DBSC. The first concept is *directly density-reachable*. If $a_1 \in A_c$ and $a_2 \in N_{a_1}^{\gamma}$, a_2 is directly density reachable from a_1. The second concept is *density-reachable*. If $a_1, a_2, ..., a_n \in A (n \geq 2)$, $a_{i+1} (i = 1, 2, ...n - 1)$ is directly density reachable from a_i, a_n is density-reachable from a_1. The last concept is *density-connected*. If $a_1, a_2, a_3 \in A$, a_2, a_3 are both density-reachable from a_1, a_2 and a_3 are density-connected with each other. The DBSC has three phases and is summarized in Algorithm 1.

Algorithm 1 Density-based Stations Clustering (DBSC)

Input:
Stations set V', station attributes set A, neighborhood radius γ and threshold α
Output:
Clusters $C_1, C_2, ...C_k$

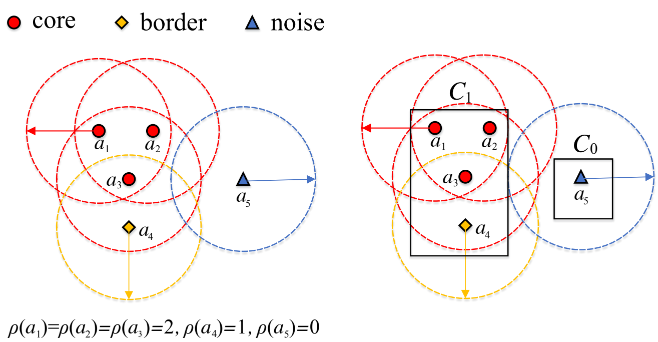
//Phase1: Initialization
Generate $N_{a_i}^{\gamma}, \forall a_i \in A$
Initialize $k \leftarrow 1$; $m_i \leftarrow 0, \forall i \in V'$
//Phase2: The process of clustering
while $A \neq \emptyset$ **do**
  Generate empty cluster set C_k
  choose a_i from A; $A \leftarrow A \setminus \{a_i\}$; Set $U \leftarrow N_{a_i}^{\gamma}$
  if $| U | < \alpha$ **then**
    a_i is a noise or a border: $m_i \leftarrow -1$
  else
    Station i is included to the k-th cluster: $m_i \leftarrow k$; $C_k = C_k \cup \{i\}$
    while $U \neq \emptyset$ **do**
      Choose a_j from U; $U \leftarrow U \setminus \{a_j\}$; $A \leftarrow A \setminus \{a_i\}$
      if $m_j = 0$ **or** -1 **then**
        Station j is included to the k-th cluster: $m_j \leftarrow k$; $C_k = C_k \cup \{j\}$
      if $| N_{a_j}^{\gamma} | \geq \alpha$ **then**
        $U \leftarrow U \cup N_{a_j}^{\gamma}$
    $k \leftarrow k + 1$
Generate set C_0 to store stations have not be classified

- **Initialization.** In this phase, the DBSC generates neighborhood set $N_{a_i}^{\gamma}$ for every point $a_i \in A$. Then it initializes the serial number k of clusters to 1 which will be updated in the process of phase 2. Next, the algorithm creates a cluster mark $m_i (i \in V)$ for each station and initializes them to 0. Cluster marks can be interpreted as follow:

$$m_i = \begin{cases} k (k > 0), i \in k\text{-th cluster} \\ -1, \ i \text{ is noise station} \end{cases} \tag{13}$$

- **Clustering.** In this phase, the algorithm runs iteratively. In each iteration, DBSC chooses a point a_i randomly and removes it from set A. If a_i is a core point, DBSC always to expand the density-connected points with a_i and marks these points belong to the k-th cluster. The k-th cluster is formed when the expansion is completed. Otherwise, a_i will be temporarily recorded as a noise point. In the subsequent iteration, a_i will be confirmed as a border point if a_i appears in the U set. All stations are classified into $k+1$ clusters: C_0 (noise cluster), $C_1, C_2, ..., C_k$.

Figure 5 gives an example of the DBSC. Neighborhood radius γ is expressed by the dotted circle. Threshold α is set to 2. There are five station attributes $a_1, a_2, ..., a_5$ in the example. Each time DBSC chooses a point a_i from A randomly and deletes it from A. Then it judges whether the a_i is a core



(a) Initialization (b) Clustering result

Fig. 5. An example of DBSC algorithm

point. If a_i is a core point, the neighborhood set $N^{\gamma}_{a_i}$ will be joined to the U. Otherwise a_i will be temporarily recorded as a noise point. DBSC next judges the station that chose from U until U becomes empty. The clustering result of example is presented in Figure 5(b).

Sorting and top-ranked stations extraction. Note that the higher station visiting frequency and the lower resources balance deviation a bike station has, more suitable it is for deploying energy-sharing cabinets. Then, the rank of station $i \in C_j$ is defined as:

$$R(i) = \frac{log(f_i)}{r_i} \tag{14}$$

f_i is generally much larger than r_i, so the logarithmic value of f_i is be used. To ensure the advantages of selected stations, stations with low ranks should be removed. So we sort the stations in each cluster in decreasing order of the rank. In each cluster, we remove the low-ranked stations and keep the top-ranked $\eta\%$ stations, and select them as the candidate for energy-sharing cabinets deployment, which are denoted as the set H. Note that parameter η should be adjust to ensure that set H covers V'.

C. Deployment Station Determination

After the candidate station extraction, *StaCover* needs to determine the stations to deploy energy-sharing cabinets. In this section, GHS is presented to solve this stations selection problem. The straightforward idea of GHS is to greedily select the station with the maximum value of quality $Q(i)$ of station i in each selection round, which is summarized in Algorithm 2. It only has two main phases. In the phase 1, it generates neighborhood set M^d_i for every station i in set H and initializes an empty set D to store the stations selected by the phase 2. Phase 2 selects the station with maximum value of the quality and includes it into the result set D iteratively. Until stations in set D completely cover all stations in set V, *i.e.*, $V = \bigcup M^d_i (i \in D)$, the algorithm outputs the result set D.

Algorithm 2 Greedy Heuristic Selection Algorithm (GHS)

Input:
Stations set V', candidate set H, deployment cost $c_i(\forall i \in H)$, and cover radius d
Output:
The e-station set D
//Phase1: Initialization
Generate sets $M^d_i, \forall i \in H$
Generate each station's quality: $Q(i), \forall i \in H$
Initialize result set: $D \leftarrow \emptyset$
//Phase2: The process of greedy selection
while $\bigcup M^d_i \neq V' (i \in D)$ **do**
  Sort all stations' quality: $Q(j), \forall j \in H$
  Find the station j with the max value of $Q(j), j \in H$
  Add station j to the set D: $D \leftarrow D \cup \{j\}$
  Remove station j from the set H: $H \leftarrow H \backslash \{j\}$
Output e-stations set D

IV. Experiment Evaluation

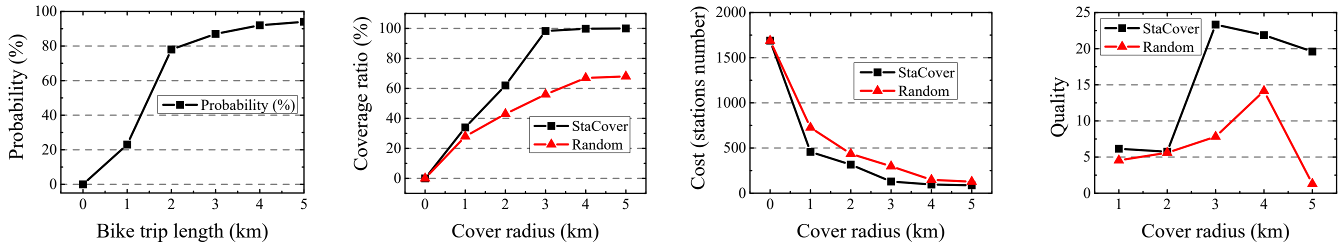
A. Experimental Settings and Results

The default parameter d is set to 3 km, and deployment costs are the same for every station ($c_i = c_j, \forall i, j \in H$). From total 1773 stations, *StaCover* selects 129 stations to deploy energy-sharing cabinets. Since there are no previous methods that handle the energy-sharing cabinets deployment in a PBS, we create a random placement method to compare with the *StaCover* (denoted by Random). In the following subsections, we study the effects of the *StaCover* in different cover radius and analyze the reason why we set cover radius d to 3 km by default.

B. Effectiveness Studies

We refer to the bike trip length distribution to determine the range of cover radius. Figure 6(a) summarizes the trip lengths distribution of the PBS users. It is clear that the majority of bike trips are relatively short, *i.e.*, more than 80% of the trips are shorter than $3km$ because people are limited by physical strength. Almost 95% of the bike trips are shorter than 5 km, so we evaluate the performance of two methods when the cover radius ranges between [0 km, 5 km] in integer. The following three factors reflect the performance of two methods:

Coverage ratio. Coverage ratio is the number of stations covered by e-stations accounts for the proportion of all stations. Figure 6(b) shows two methods' coverage ratio in different cover radius. For example, when the cover radius is 3 km, every e-station has the ability to cover stations within 3 km riding distance from itself. When the cover radius is set to 1 km, many stations have no neighbor within 1 km. So we need to remove these isolated stations. There is only about 35% of all stations covered by e-stations after remove. When cover radius is 3 km, coverage ratio in *StaCover* reaches 98.3%. However, when the cover radius is 4 km or 5 km, the increase in coverage ratio reaches the bottleneck. But the cover radius is not the higher the better. The higher the cover radius,



(a) Bike trip length distribution (b) Coverage ratio of two methods (c) Deployment cost of two methods (d) Quality of two methods

Fig. 6. Performance of two methods under different cover radius.

the longer people will ride to rent the battery. The *Random* method selects stations randomly with the same deployment cost as the *StaCover*. When the cover radius of two methods is the same, the coverage ratio of the *StaCover* is always higher than the *Random*.

Deployment cost. Figure 6(c) shows the total deployment cost of two methods in different cover radius. Our default deployment costs are same for every station ($c_i = c_j, \forall i, j \in H$). Here the deployment cost is represented by the number of selected stations. Obviously, the smaller the cover radius, the more stations we need to deploy the energy cabinets, and the more we will spend. In the view of curve changes, when the cover radius value change from $0\ km$ to $3\ km$, *StaCover*'s deployment cost declines more obviously and the number of selected stations decrease from 1688 to 129. However, when the cover radius is $4\ km$ or $5\ km$, the total cost no longer has a significant reduction. Compared with the *StaCover*, the *Random* method selects the stations with the same coverage ratio as *StaCover*. As Figure 6(c) shows, we can observe that the *Random* would cost more than *StaCover* under the same coverage ratio.

Quality. Figure 6(b) and Figure 6(c) respectively present the coverage ratio and total deployment cost of two methods under different cover radius. In order to observe the effect of two methods more intuitively, we use quality $Q(i)$ to integrate the coverage ratio and deployment cost. *StaCover* hopes the higher value of the coverage ratio, the lower value of the deployment cost and cover radius. Figure 6(d) presents the quality sum of all e-stations selected from two methods in different cover radius. It is obvious that under the same cover radius, *StaCover* always has a better quality. In addition, when the cover radius is $3\ km$, *StaCover* has the best quality. That is, *StaCover* get the best results when the cover radius is $3\ km$.

V. Conclusion

To mitigate the pressure on the long-distance transportation modes, HPTC tries to combine the PBS with cabinets of MES to form BES. It thus brings up a new problem: which bike stations should be selected to deploy the cabinets so as to minimize the deployment cost and satisfy that all users can rent and return battery conveniently from at least one nearby bike station on the riding trip. This paper proposes the method *StaCover* to handle this problem. Firstly, *StaCover* clusters stations by the DBSC, and then extracts candidate by sorting the stations attributes. At last, *StaCover* presents the GHS to solve the station selection problem. Some experiments are conducted to evaluate the method using three months of data which collected from HPTC. It also studies the effectiveness of our method by a set of different parameters comparison.

References

[1] Paul DeMaio. Bike-sharing: History, impacts, models of provision, and future. *Journal of public transportation*, 12(4):41–56, 2009.

[2] Yexin Li, Yu Zheng, Huichu Zhang, and Lei Chen. Traffic prediction in a bike-sharing system. In *Proceedings of the 23rd International Conference on Advances in Geographic Information Systems (SIGSPATIAL '15)*, Seattle, Washington, USA, November 3-6 2015. ACM.

[3] Zhang Jianhui, Lu Pengqian, Li Zhi, and Gan JiaYu. Distributed trip selection game for public bike system with crowdsourcing. In *Proceedings of 2018 IEEE International Conference on Computer Communications (INFOCOM'18)*, Accepted to appear. Honolulu, HI, USA, April 15-19 2018. IEEE.

[4] Zhi Li, Jianhui Zhang, Jiayu Gan, Pengqian Lu, and Fei Lin. Large-scale trip planning for bike-sharing systems. In *Proceedings of 14th IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS'17)*, Short paper. FL, USA, October 22-25 2017. IEEE.

[5] Yang Yadong. Pengpai paper. http://www.thepaper.cn/newsDetail_forward_1686466/. [Online; accessed 16-May-2017].

[6] Tong Xu, Hengshu Zhu, Xiangyu Zhao, Qi Liu, Hao Zhong, Enhong Chen, and Hui Xiong. Taxi driving behavior analysis in latent vehicle-to-vehicle networks: A social influence perspective. In *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining (KDD'16)*, pages 1285–1294, San Francisco, California, USA, August 13-17 2016. ACM.

[7] Susan Shaheen, Stacey Guzman, and Hua Zhang. Bikesharing in europe, the americas, and asia: past, present, and future. *Transportation Research Record: Journal of the Transportation Research Board*, 2143:159–167, 2010.

[8] Petr Slavık. A tight analysis of the greedy algorithm for set cover. *Journal of Algorithms*, 25(2):237–254, 1997.

[9] Wang Jiqiang. Model and algorithm for set cover problem. *Computer Engineering and Applications*, 49(17):15–17, 2013.

[10] J Martin Bland and Douglas G Altman. Statistics notes: measurement error. *British Medical Journal Publishing Group (BMJ)*, 313(7059):744, 1996.

[11] A-H Ahmed. Characterization of beta, binomial, and poisson distributions. *IEEE Transactions on reliability*, 40(3):290–295, 1991.

[12] Alberto Caprara, Matteo Fischetti, and Paolo Toth. A heuristic method for the set covering problem. *Operations research*, 47(5):730–743, 1999.

[13] Vasek. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of operations researchINFORMS*, 4(3):233–235, 1979.

[14] Mapbox. https://www.mapbox.com/mapbox-studio/. [Online; accessed 12-July-2017].

[15] C Saranya and G Manikandan. A study on normalization techniques for privacy preserving data mining. *International Journal of Engineering and Technology (IJET)*, 5(3):2701–2704, 2013.