Bicycle-Sharing Systems Expansion: Station Re-Deployment through Crowd Planning

Jiawei Zhang*, Xiao Pan†, Moyin Li*, Philip S. Yu*
*University of Illinois at Chicago, Chicago, IL, USA
†Shijiazhuang Tiedao University, China
jzhan9@uic.edu, smallpx@stdu.edu.cn, mli60@uic.edu, psyu@cs.uic.edu

ABSTRACT

Bicycle-sharing systems (BSSs) which provide short-term shared bike usage services for the public are becoming very popular in many large cities. The accelerating bike traveling demands from the public have driven several significant expansions of many BSSs to place additional bikes and stations in their extended service regions. Meanwhile, to capture individuals' traveling needs more precisely, in the expansion, many BSSs have set up online websites to receive station location suggestions from the public. In this paper, we will study the bike station re-deployment problem in the BSSs expansion. Besides the historical bike usage and construction cost information, the crowd suggestions are also incorporated in the problem. The station re-deployment problem is very challenging to solve, and it covers two sub-tasks simultaneously: (1) bike station locations identification, and (2) bike dock assignment (to the deployed stations). To address the problem, a novel bike station re-deployment framework, CROWDPLANNING, is introduced in this paper. In both station deployment and capacity assignment tasks, CROWDPLANNING fuses different categories of spatial information including the crowd suggestions, individuals' historical bike usage and the construction costs simultaneously. By formulating these two tasks as two optimization problems, the optimal expansion strategies can be identified by CROWDPLANNING for the BSSs. Extensive experiments are conducted on the real-world BSSs and crowd suggestion dataset to demonstrate the effectiveness of framework CROWDPLANNING.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications - Spatial Databases and GIS

Keywords

Crowd Planning; Station Deployment; Bicycle-Sharing Systems; Data Mining; Geographic Information Systems

1. INTRODUCTION

To facilitate the daily commutes for local residents and sightseeing trips for tourists in the urban areas, bicycle-sharing systems (BSSs) [15, 16] providing shared short-term bike rental services have been launched in many cities. Some representative examples of the existing BSSs in the US include the Chicago Divvy Bike[1], New York Citi Bike[2], and San Francisco Bay Area Bike[3], etc. Due to their relatively low rental prices and easily accessible bike stations, BSSs have been used as an important short-distance trip supplement for both private vehicles and regular public transportation.

BSSs [15] are becoming more and more popular nowadays, and people's accelerating traveling demands have driven the expansions of many BSSs in recent years. For instance, the Chicago Divvy Bike launched at the middle of 2013 had 75 bike stations and 750 bikes initially, but now Divvy operates $4,760$ bikes at 474 stations at the Chicago city, which has become the largest BSS in the US. A detailed analysis about the Chicago Divvy BSS is available in [15], based on which several interesting problems, like the trip route planning [16], has been studied already. Besides Divvy, many other BSSs have also planned their upcoming expansions. For instance, the Citi Bike plans to double the system in size by 2017 [1], and Bay Area Bike Share plans to expand to 7,000 bikes by 2017 [2]. In the expansion, additional bikes and stations will be added to the BSSs, where the re-deployment of the stations and the assignment of capacities (i.e., the bike docks) to the deployed stations are both important issues requiring careful and thorough investigations.

Meanwhile, crowdsourcing as an online, distributed problem-solving and production model has become very popular in recent years [6, 4]. Formally, crowdsourcing refers to the practice of obtaining needed services, ideas, or content by soliciting contributions from a large group of people and especially from the online communities. Crowdsourcing has been widely applied in addressing various practical problems. Depending on the specific application scenarios, existing crowdsourcing systems can generally be divided into several categories [8], which include crowdvoting, crowdsearching and crowdfunding.

Besides these existing applications, crowdsourcing can be applied in the concrete planning tasks as well, e.g., traffic planning and urban planning, in which companies/governments can ask for ideas and proposals from the public before carrying out the projects. Formally, the utilization of crowdsourcing in traffic planning and urban planning problems is defined as *crowd planning* in this paper. Crowd planning effectively combines the efforts of numerous self-identified volunteers or part-time workers, where each contributor, acting on their own initiative, adds a small contribution that combines with those of others to achieve better planning results.

[1]https://www.divvybikes.com
[2]https://www.citibikenyc.com
[3]http://www.bayareabikeshare.com

SIGSPATIAL'16, October 31-November 03, 2016, Burlingame, CA, USA

DOI: http://dx.doi.org/10.1145/2996913.2996926

Crowd planning systems have also been adopted in the real-world BSSs expansions already, and many BSSs have launched their websites to receive bike station placement suggestions from the public. For instance, Divvy creates a station suggestion webpage[4], where people can submit their bike station suggestions according to their bike usage needs (e.g., for shopping, work, school, and home, etc.) and can also post comments on other people's suggestions. By the middle of 2015, Divvy had received about $1,873$ new station suggestions and comments from the public, which were effectively considered in its recent expansion when placing new stations and adjusting existing stations in 2015.

Problem Studied: In this paper, we will study the bike station re-deployment problem in the BSSs expansion, besides the historical bike usage information and construction cost issues (to be introduced later), in which the crowd suggestion information is effectively incorporated. Formally, the problem is named as the "Crowd Suggestion based System Expansion" (CSSE) problem. Compared with conventional offline manual bike station planning, deploying stations by incorporating the online suggestions can accumulate a large number of diverse bike station suggestions from the public in a relatively short period of time, which has significant advantages in various aspects, e.g., the costs, speed, quality, flexibility, scalability, and diversity of the suggestions. Viewed in this perspective, CSSE will be an interesting and important research problem.

Besides its importance, the CSSE problem is also a new problem, and, to this context so far, no works have been done on station deployment through crowd planning for BSSs. We are the first to propose the CSSE problem, and we are also the first to introduce the "*crowd planning*" concept. The CSSE problem is very different from traditional station placement works about the *gas station* [3] and *hydrogen filling station* [12] and the recent *charging station* [9]. These existing works mainly focus on adding *new* public facility stations only based on *one single information source*, e.g., nearby geographic information or historical vehicle trajectory data. Distinct from these existing works, (1) instead of merely adding new stations [3, 12, 9], the objective of the CSSE problem is to *re-deploy the bike stations for BSSs*, where both *new* bike docks and stations can be *added*, while the *existing* bike docks and stations also need to be *removed* or *adjusted*; (2) instead of merely utilizing one single information source (like [3, 12, 9]), various categories of other information will also be incorporated in the CSSE problem, including the *crowd suggestions*, the *historical bike usages* and the *bike/station construction/adjustment costs*.

Despite its importance and novelty, the CSSE problem is quite challenging to address due to various reasons:

- *Crowd Suggestions Information*: The suggestions received from the public provide important information about the areas with heavy bike traveling demands, which can be the potential locations to place the new bike stations. How to utilize the crowd suggestion information in the station planning is still an open problem so far.

- *Historical Bike Usage Information*: Crowd suggestions are mainly about new stations, while the expediency of these existing stations can be revealed by the historical bike usages in the past. Incorporating the *historical bike usage* information is a must when adjusting the existing stations.

- *Station deployment cost*: Station re-deployment involves various types of actions, which include (1) adding a new station, (2) moving an existing station to a new place, (3) adding more docks to an existing station, as well as (4) removing redundant docks from an existing station. All of these actions will lead to certain construction costs inevitably, which should be minimized in the CSSE problem.

- *Planning Information Fusion*: Various categories of information are available, while fusing of these different types of information together to address the CSSE problem will be a great challenging problem.

To address these above challenges, a novel station planning framework CROWDPLANNING is introduced in this paper. CROWDPLANNING can effectively fuse different categories of spatial information together to address the CSSE problem. By iteratively partitioning the service region based on quadtree [14] into small-sized cells, CROWDPLANNING identifies the optimal regions (of pre-specified granularity) to deploy the bike stations. CROWDPLANNING optimizes the distance between the places to deploy the new bike stations and the suggested spots from the public. What's more, CROWDPLANNING also utilizes the historical usages information of each existing station, where frequently used stations tend to be preserved and get increased in capacities (i.e., adding more bike docks), while the remaining ones will either be removed or get decreased in capacities (i.e., removing the redundant bike docks). In addition, the costs introduced by adding/removing stations and bike docks are considered, which will be minimized in CROWDPLANNING. CROWDPLANNING transforms the tasks covered in the CSSE problem into two optimization problems, and can identify both the locations to place bike stations, as well as the capacities of these stations concurrently. (There are many other factors can cause the demand changes, e.g., the weather, temperature and seasonality reasons. However, since these factors will affect the whole service region equally, they are not helpful for distinguish the advantages of different potential station locations and will not be considered in this paper.)

The remaining part of this paper is organized as follows. In Section 2, we introduce the definitions of several important concepts and the formulation of the CSSE problem. After that, we talk about the CROWDPLANNING framework in detail in Section 3, which is evaluated in Section 4. Finally, the related works are available in Section 5 and we conclude the paper in Section 6.

2. PROBLEM FORMULATION

In this section, we will first define several important concepts, and then introduce the formulation of the CSSE problem.

2.1 Terminology Definitions

In the CSSE problem, we aim at deploying the bike stations within the service region, and the bike stations to be deployed can be formally represented as follows:

DEFINITION 1. *(Station): Let $\mathcal{S} = \{s_1, s_2, \cdots, s_N\}$ be the $N = |\mathcal{S}|$ stations available in the BSSs. The specific location of each station (e.g., $s_i \in \mathcal{S}$) can be represented as a geographical coordinate pair $(longitude(s_i), latitude(s_i))$. Meanwhile, at each station e.g., $s_i \in \mathcal{S}$, the number of available bikes is limited and can be represented as station capacity: $capacity(s_i) \in \mathbb{N}^+$. Therefore, station s_i in the BSSs can be represented as a tuple $s_i = (ID(s_i), (latitude(s_i), longitude(s_i)), capacity(s_i))$, where $ID(s_i)$ denotes the unique ID of the station.*

In the station re-deployment process, both the geographical locations and the capacities of these stations can be adjusted, but their IDs will be not be changed. In this paper, we will misuse s_i and

[4]http://suggest.divvybikes.com/page/about

$ID(s_i)$ interchangeably when referring to station s_i for simplicity. Formally, to differentiate the bike stations before and after the station re-deployment (i.e., system expansion), we name the bike stations before the expansion as the *current stations* (denoted as set $\mathcal{S}_H$), while those after the expansion are called the *future stations* (denoted as set $\mathcal{S}_F$). Some of the current stations which are not changed in the re-deployment will be preserved in the future station set (which can be denoted as $\mathcal{S}_H \cap \mathcal{S}_F$), while a set of new stations will also be added to the system, which can be represented as the *new station* set $\mathcal{S}_N = \{ID(s)|s \in \mathcal{S}_F\} \setminus \{ID(s)|s \in \mathcal{S}_H\}$.

In the online crowd suggestion sites, people can suggest new bike stations according to their own bike traveling needs, and the crowd station suggestion can be denoted as:

DEFINITION 2. *(Station Suggestions): A station suggestion can be denoted as a tuple* $h = ((latitude(h), longitude(h)), text(h))$, *where* $(latitude(h), longitude(h))$ *pair represents the suggested location for the new station, and* $text(h)$ *denotes the text information posted with station suggestion. The complete station suggestion can be represented as set* $\mathcal{H} = \{h_1, h_2, \cdots, h_{|\mathcal{H}|}\}$.

From the crowd suggestions, we can obtain both the locations and text information about the travel needs from the public. The crowd suggestion text information involves the suggestion motivations (e.g., for shopping, or for train-based commute) and can also effectively help reveal nearby geographical sites (e.g., shopping malls and metro stations) as well. Therefore, we do not need to take additional considerations about other nearby spot information again in the CSSE problem.

Meanwhile, in bicycle-sharing systems, when a bike is borrowed from (and returned to) a station, the system will keep a record about the bike ID, station ID and the specific time when the bikes are checked out/in, which will form a bike trip record.

DEFINITION 3. *(Historical Bike Trip): Formally, let set* $\mathcal{T} = \{t_1, t_2, \cdots, t_M\}$ *be the trips that people ride among the stations in* $\mathcal{S}$. *Each trip (e.g.,* $t_j \in \mathcal{T}$*) can be represented as a tuple* $t_j = (start\ station(t_j), start\ time(t_j), end\ station(t_j), end\ time(t_j))$. *The duration of trip* t_j *can be represented as* $duration(t_j) = end\ time(t_j) - start\ time(t_j)$.

For stations which are no longer needed, the service provider will revoke the station together with bikes and docks back to the warehouse, some of which will be used and placed at another location (it will be equivalent to moving an existing station to a new location). Meanwhile, from the BSSs service provider's perspective, adding new stations, changing the location and adjusting the capacities of current stations will all introduce certain construction costs. The costs introduced by adding/removing stations and bikes at different places can be different due to various reasons, e.g., traffic condition, the flow of people and the surroundings buildings. In this paper, we will neglect these factors for simplicity reasons and treat the costs at different stations to be the same.

DEFINITION 4. *(Station Re-deployment Costs): Formally, let* $cost_s^+(s_i)$ *and* $cost_s^-(s_i)$ *be the cost of adding/removing a station* s_i *in the system. Therefore, by adding a new station* $s_i \notin \mathcal{S}_N$ *to the system, the introduced cost will be* $cost_s^+(s_i)$*; by removing an existing station* $s_i \in \mathcal{S}_H$ *from the system, the cost will be* $cost_s^-(s_i)$*; while by moving a current station* $s_i \in \mathcal{S}_H$ *to a new place, the cost will be* $cost_s^+(s_i) + cost_s^-(s_i)$. *For simplicity, in this paper, we treat the cost of adding/removing different stations to be identical, which can all be simplified as* $cost_s^+$ *and* $cost_s^-$. *Similarly, the costs of adding/removing the bike docks at a current station to adjust its capacity can be simply represented as* $cost_d^+$ *and* $cost_d^-$ *respectively.*

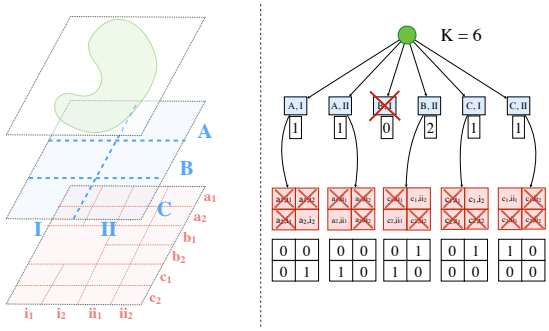


Figure 1: An example of the iterative map gridding.

In placing the bike stations and assigning the station capacities, the construction costs should be as low as possible.

2.2 Problem Definition

Before the bike station re-deployment, the BSSs service provider needs to perform very thorough investigation and determine the numbers of stations and bikes to operate after the expansion. Formally, let $K \in \mathbb{N}^+$ and $C \in \mathbb{N}^+$ be the numbers of stations and bikes to operate after the expansion respectively (in the CSSE problem, the K and C are already obtained from the investigation). According to the above concepts, we can define the CSSE problem formally as follows:

DEFINITION 5. *(The* CSSE *Problem): Given the current stations* $\mathcal{S}_H$, *historical trip records* $\mathcal{T}$, *crowd suggestions from the public* $\mathcal{H}$, *the* CSSE *problem aims at expending the bicycle-sharing system to obtain the future bike stations* $\mathcal{S}_F$ *after the re-deployment, which can maximize the convenience for the public to use the bikes and minimize the construction costs at the same time. The optimal future bike stations* $\mathcal{S}_F^*$ *(covering both the optimal locations/regions and capacities of the future stations in* $\mathcal{S}_F$*) can be obtained by addressing the following objective function:*

$$\mathcal{S}_F^* = \arg\max_{\mathcal{S}_F} convenience(\mathcal{S}_F) - \beta \cdot cost(\mathcal{S}_F)$$

$$s.t.\ |\mathcal{S}_F| = K, \sum_{s \in \mathcal{S}_F} capacity(s) = C,$$

where β *denotes the weight of the cost measure, while the concrete representation of* $convenience(\cdot)$ *and* $cost(\cdot)$ *will be introduced in Section 3 in detail.*

3. PROPOSED METHOD

In this section, we will introduce the CROWDPLANNING framework in detail, which consists of 2 steps: (1) iterative map gridding based station deployment, and (2) bike capacity assignment to the deployed stations. We will introduce these two steps in Section 3.1 and Section 3.2 respectively.

3.1 Iterative Map Girdding based Station Deployment

Identifying the specific location coordinates of bike stations directly on the map can be quite difficult, and instead the objective of CSSE will be to identify the regions (of a reasonable granularity) to place the bike stations. In this section, we propose to partition the service region on the map into cells based on quadtree [14] and deploy the stations in these cells iteratively until the cells can meet the pre-specified granularity requirement.

3.1.1 Iterative Map Gridding

Deployment of the bike stations in the city can be affected by various factors. Instead of identifying the exact location coordinates for the bike stations directly, we propose to place the bike

stations in the region of a larger granularity, which are called the cells in the paper.

DEFINITION 6. *(Map Cell): Formally, a map cell refers to a specific square region on the map, and the map can be partitioned into a set of map cells by specifying their Cartesian coordinates respectively. The process of partitioning a map region into cells of identical size is formally called "map gridding".*

Given the square map cell size w, at the initial stage, CROWDPLANNING will partition the target service region into equal-sized cells. This step is quite flexible. By adjusting the parameter w, CROWDPLANNING can study the station deployment on cells of different granularities (which can also be an exact point at the extreme case when $w \to 0$). Let the target service area on the map be a rectangle region of dimensions $L \times W$. Framework CROWDPLANNING will partition the region into $\lceil \frac{L}{w} \rceil \times \lceil \frac{W}{w} \rceil$ cells, and the cell number will grow quadratically as w decreases.

CROWDPLANNING transforms the station deployment task as an optimization problem, where one unique variable will be defined to each partitioned cell. Therefore, a small w will introduce too many cells and inference variables in the CROWDPLANNING framework, which will pose great challenges in solving the optimization objective function. In this paper, we propose to partition the service region iteratively from large-sized cells, and keep partitioning the cells into smaller ones until meeting the required granularity. In the iterative partitioning process, cells without any deployed stations will be pruned and not involved in the next round of partitioning, which will help shrink the search space greatly in each round.

For instance, as shown in the left plot of Figure 1, the BSSs service provider has pre-specified a service region, which is the area of the green color. By specifying the initial parameter w_0, CROWDPLANNING can obtain the initial cells of the service region, which can be represented as set $\mathcal{G} = \{g_1, g_2, \cdots, g_l\}$ (where $l = \left\lceil \frac{L}{w_0} \right\rceil \times \left\lceil \frac{W}{w_0} \right\rceil$). In the Figure 1, the initial cells are the square areas with coordinates (A, I), (A, II), (B, I), (B, II), (C, I) and (C, II) of the blue color, which also correspond to the blue nodes at the second level of the right tree-structured plot. The K bike stations will first be deployed in these 6 partitioned cells in $\mathcal{G}$, and the detailed station deployment methods will be introduced in the following subsections. Let's assume, the cells (A, I), (A, II), (C, I) and (C, II) are all assigned with 1 bike stations and cell (B, II) is assigned with 2 bike stations, while cell (B, I) has no stations. As introduced before, we will prune the cell (B, I) from the candidate cells, which will not be involved in the next round any more.

Meanwhile, for the cells $g_j \in \mathcal{G}$ being assigned with $M \geq 1$ bike stations but the size of cell g_j is greater than the pre-required granularity $\bar{w}$, we propose to further partition them into smaller cells with the quadtree technique [14] and further assign the M bike stations in the smaller cells with the same method. For instance, in Figure 1, (A, I), (A, II), (B, II), (C, I) and (C, II) are further quartered into 4 equal-sized smaller cells, and the assigned stations in the previous round are further placed in the smaller cells. According to the results shown below the red quad-cells, the small-sized cells (a_2, i_2), (a_2, ii_1), (b_1, ii_2), (b_2, ii_1), (c_1, i_2) and (c_1, ii_1) are assigned with 1 bike station, while the rest having no bike stations will be pruned from the candidate list. Such a process continues until the cell size is no greater than $\bar{w}$.

In the next subsections, we will introduce the method about how to deploy stations into cells with various spatial information.

3.1.2 Crowd Suggestion based Station Deployment

The station locations suggested by the crowd are generally the places that people want to go from/to. In the station suggestion website, the station suggestion text information can provide information about various nearby sites (e.g., shopping malls and metro stations). Therefore, we don't need to obtain and consider the offline GIS data from other sources over again as the traditional transportation planning models do.

As described in Section 2, the set of suggestions received from the crowd can be represented as $\mathcal{H} = \{h_1, h_2, \cdots, h_{|\mathcal{H}|}\}$. After the map gridding process, each suggested station will belong to exactly one map cell, which can be represented with a mapping $f : \mathcal{H} \to \mathcal{G}$, and the cell that suggestion h_i belongs to can be denoted as $f(h_i) \in \mathcal{G}$. Via mapping $f(\cdot)$, we can also represent the set of suggestions that are mapped to cell g_i via $f(\cdot)$ as $\Gamma_H(g_i) = \{h_j | h_j \in \mathcal{H}, f(h_j) = g_i\}$. In addition, for each cell, e.g., $g_i \in \mathcal{G}$, we introduce an unique binary station indicator variable $y(g_i) \in \{0, 1\}$ to represent the re-deployment result, where $y(g_i) = 1$ if a bike station is placed in the cell, and $y(g_i) = 0$ otherwise. For all the cells, the whole set of binary variables can be represented as $\mathcal{Y} = \{y(g_1), y(g_2), \cdots, y(g_l)\}$, where $l = |\mathcal{G}|$.

Based on the suggestions from the crowd, the re-deployed stations should be as close to the crowd's suggested station locations as possible, so as to maximize their bike usage convenience. In other words, based on the suggestions from the crowd, the binary indicator variables of cells containing more crowd station suggestions are more likely to be 1, while the variables of the remaining cells tend to be 0 on the other hand. Based on such an intuition, for any two given cells $g_i, g_j \in \mathcal{G}$, if g_i contains more station suggestions than g_j (i.e., $|\Gamma_H(g_i)| \geq |\Gamma_H(g_j)|$), then a bike station is more likely to be placed in g_i than in g_j, i.e., $y(g_i) \geq y(g_j)$. Therefore, we can define the convenience$(\cdot)$ measure based on the crowd suggestion information in the problem formulation as the sum of the binary variables of all the cells, together with a set of constraints:

$$\text{convenience}_c(\mathcal{S}_F) = \sum_{g_i \in \mathcal{G}} y(g_i)$$

$$s.t.\ \ y(g_i) \geq y(g_j), \text{if } |\Gamma_H(g_i)| \geq |\Gamma_H(g_j)|, \forall g_i, g_j \in \mathcal{G}.$$

3.1.3 Historical Usage based Station Deployment

Crowd suggestions effectively indicate the crowd's expectations for the new stations. However, when it comes to the current stations, expediency of those stations in the past will be more important, which can be revealed by the historical bike usages.

After a long-time of usages, customers will be familiar with the locations of stations that they frequently go to. To avoid moving too many existing bike stations to new places and creating troubles for customers to change their usage behaviors, framework CROWDPLANNING will not adjust the locations of existing stations that are heavily used by the customers. Based on the historical trip records, for each cell $g_i \in \mathcal{G}$, we can count the trips either starting from or ending at the station in g_i and represent it as set $\Gamma_T(g_i) = \{t_j | t_j \in \mathcal{T}, \text{start station}(t_j) \in g_i \vee \text{end station}(t_j) \in g_i\}$. If cell g_i contains a current bike station, the number of usages of the station can be represented as $|\Gamma_T(g_i)| > 0$; if cell g_i doesn't contain any current stations, we will have $|\Gamma_T(g_i)| = 0$.

Current stations which are heavily used would be more likely to be unchanged and preserved in the re-deployment. To achieve such an objective, without loss of generality, we can further represent the station preserving probability with the following logistic function

$$P(g_i) = \frac{e^{k|\Gamma_T(g_i)|}}{1 + e^{k|\Gamma_T(g_i)|}},$$

where parameter k is a constant number and it is assigned with value 1 in this paper.

The function $P(g_i)$ is a continuous function between 0 and 1, which grows monotonically as more trips starting or ending at cell g_i. In other words, for stations with a larger number of usages, their preserving probability will be higher. Therefore, we can represent the convenience$(\cdot)$ measure based on the historical usage information as follows:

$$\text{convenience}_u(\mathcal{S}_F) = \sum_{g_i \in \mathcal{G}} y(g_i) \cdot P(g_i),$$

where the more usages of stations in a cell g_i, the larger its station preserving probability $P(g_i)$ will be, and the more likely its corresponding binary variable $y(g_i)$ will be assigned with value 1 when maximizing the convenience measure.

3.1.4 Minimum Cost based Station Deployment

Besides the concerns about the convenience for customers, another important factor needs to be considered in bike station redeployment is the station deployment costs. The cost can be introduced by either removing existing stations or constructing new stations, where changing the location of an existing stations can be achieved by removing the station first and construct a new station at a new place. As introduced in Section 2, the costs of removing/constructing different stations can be represented as cost_s^- and cost_s^+ respectively.

For cell g_i in set $\mathcal{G}$, we introduce a new notation $\bar{y}(g_i) \in \{0, 1\}$ to denote whether there is a current bike station in g_i or not before the re-deployment, which is a known value and can be obtained by scanning the current station data. By combining notation $\bar{y}(g_i)$ with the target variable $y(g_i)$, we can know what happens to cell g_i during the system expansion:

$$\begin{cases} \text{no current nor future station at } g_i, & \text{if } \bar{y}(g_i) = 0, y(g_i) = 0; \\ \text{a new station is constructed at } g_i, & \text{if } \bar{y}(g_i) = 0, y(g_i) = 1; \\ \text{a current station is removed from } g_i, & \text{if } \bar{y}(g_i) = 1, y(g_i) = 0; \\ \text{a current station is preserved at } g_i, & \text{if } \bar{y}(g_i) = 1, y(g_i) = 1. \end{cases}$$

Meanwhile, the costs introduced by cell g_i in bike station redeployment can be represented as

$$\text{cost}(g_i) = \begin{cases} \text{cost}_s^+, & \text{if } \bar{y}(g_i) = 0, y(g_i) = 1; \\ \text{cost}_s^-, & \text{if } \bar{y}(g_i) = 1, y(g_i) = 0; \\ 0, & \text{otherwise.} \end{cases}$$

And the overall costs introduced by deploying stations in $\mathcal{G}$ can be represented as be

$$\text{cost}(\mathcal{S}_F) = \text{cost}(\mathcal{G}) = \sum_{g_i \in \mathcal{G}} \text{cost}(g_i)$$
$$= \sum_{g_i \in \mathcal{G}} \max\{y(g_i) - \bar{y}(g_i), 0\} \cdot \text{cost}_s^+ + \max\{\bar{y}(g_i) - y(g_i), 0\} \cdot \text{cost}_s^-.$$

3.1.5 Distribution Density based Station Deployment

In addition to the convenience and cost issues, the overall distribution of the stations within the city needs to be considered carefully. Concentrating too many stations densely within a small area will greatly decrease usage frequency of each bike, while separating station too sparsely across the whole city will make the stations unreachable within the free-ride time. To avoid these two extreme cases, in this paper, we propose to constrain the bike station distribution density in the CROWDPLANNING framework. Considering that the stations are placed in the cells, the density of bike distributions is actually calculated based on a group of nearby cells on the map.

The partitioned cell are actually square regions partitioned based on the map, and each cell (e.g., $g_i \in \mathcal{G}$) can be connected to 8 other cells on the map by sharing either common boundaries or common vertices. In this paper, we introduce the concept of "*grid-based walk*", which can go to other nearby cells from the origin cell step by step and each step passes through either a common boundary or a vertex. For instance, as shown in Figure 2, from the origin cell g_i, by traveling at most one single step, the "*grid-based random walk*" can reach 8 other different cells, which can be represented as set $\Gamma_D(g_i, 1)$. Similarly, we can represent the set of cells that the "*grid-based random walk*" can reach with at most n steps as $\Gamma_D(g_i, n)$, $n \in \mathbb{N}$ and $\Gamma_D(g_i, n-1) \subset \Gamma_D(g_i, n)$ holds. With the "*grid-based walk*", we can represent the density of bike station distribution within a square region involving $(2n+1)^2$ $(n \geq 1)$ cells with g_i as the center as

$$\text{density}(g_i, n) = \frac{\sum_{g_j \in \Gamma_D(g_i, n) \cup \{g_i\}} y(g_j)}{(2n+1)^2}.$$

Furthermore, we can introduce the bike distribution density constraints to be

$$\underline{D} \leq \text{density}(g_i, n) \leq \overline{D}, \forall g_i \in \mathcal{G}.$$

Here, the parameters $\underline{D}$ and $\overline{D}$ represent the lower-bound and upper-bound of the station distribution density respectively, whose sensitivity analysis is available in the experiment section.

3.1.6 Joint Objective Function of Station Deployment

By taking considerations of both the suggestions from the crowd and current station usages, we can obtain the concrete representation of the $convenience(\cdot)$ used in defining the CSSE problem as

$$\begin{aligned} \text{convenience}(\mathcal{S}_F) &= \text{convenience}_c(\mathcal{S}_F) + \alpha \cdot \text{convenience}_u(\mathcal{S}_F) \\ &= \sum_{g_i \in \mathcal{G}} y(g_i) + \alpha \cdot \sum_{g_i \in \mathcal{G}} y(g_i) \cdot P(g_i) \\ &s.t.\ \ y(g_i) \geq y(g_j), \text{if } |\Gamma_H(g_i)| \geq |\Gamma_H(g_j)|, \forall g_i, g_j \in \mathcal{G}, \end{aligned}$$

where α denotes the weight for the convenience term based on the historical usage information.

The convenience measure together with the the cost measure can be used to define the final joint objective function, and the optimal bike station re-deployment results $\mathcal{Y}^*$ can be represented with the following joint optimization function:

$$\begin{aligned} \mathcal{Y}^* &= \arg\max_{\mathcal{Y}} \text{convenience}(\mathcal{S}_F) - \beta \cdot \text{cost}(\mathcal{S}_F) \\ &= \arg\max_{\mathcal{Y}} \sum_{g_i \in \mathcal{G}} \Big(y(g_i) + \alpha \cdot y(g_i) \cdot P(g_i) - \beta \cdot \\ &\quad (\max\{y(g_i) - \bar{y}(g_i), 0\} \cdot \text{cost}_s^+ + \max\{\bar{y}(g_i) - y(g_i), 0\} \cdot \text{cost}_s^-) \Big) \\ s.t.\ \ &\underline{D} \leq \text{density}(g_i, n) \leq \overline{D}, \forall g_i \in \mathcal{G}, \\ &y(g_i) \geq y(g_j), \text{if } |\Gamma_H(g_i)| \geq |\Gamma_H(g_j)|, \forall g_i, g_j \in \mathcal{G}, \\ &\sum_{g_k \in \mathcal{G}} y(g_k) = K; y(g_k) \in \{0, 1\}, \forall g_k \in \mathcal{G}. \end{aligned}$$

The objective equation is an integer programming problem, which is shown to be an NP-hard problem and the k-median problem can be mapped to it in polynomial time. Therefore, no close-form solution exists for the above objective equation. To address the problem, in this paper, a two-step method is applied in framework CROWDPLANNING:

Step 1 Constraint Relaxation: We propose to relax the strict binary constraints on variables $y(g_k) \in \{0, 1\}, \forall g_k \in \mathcal{G}$ and allow them to take any real values within range $[0, 1]$, i.e., $y(g_k) \in$

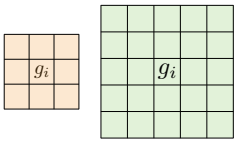


Figure 2: An example of grid-based walk.

$[0, 1], \forall g_k \in \mathcal{G}$. The relaxed objective function can be solved with existing linear programming methods easily. We use the open-source linear programming toolkit, e.g., PuLP[5] and SciPy[6], and detailed derivation steps will not be introduced here due to the limited space. The objective function of the linear programming problem itself is convex, and the optimal solution can be identified.

Step 2 Post-Processing: Due to the constraint relaxation, the variables $\mathcal{Y}$ obtained in the previous step can take any real values in range $[0, 1]$. To preserve the binary constraints, we propose to round the obtained parameters by sorting the parameters decreasingly. The top-K parameters with the largest values are selected from the results and assigned with value 1, while the remaining are assigned with 0 instead.

3.2 Capacity Assignment at Stations

For all the cells selected to place the bike stations, in this section, we will study how to distribute the bikes to stations in these cells, which is called the *capacity assignment* sub-problem. Similar to the station deployment problem, the *capacity assignment* problem can also be addressed based on information about (1) the suggestions from the crowd, (2) current station usages, and (3) the construction costs for adjusting the station capacities.

Crowd Suggestions based Capacity Assignment

Based on the re-deployment results, among all the cells in set $\mathcal{G}$, we can extract the cells with bike stations being deployed, which can be represented as $\tilde{\mathcal{G}} = \{g_i | g_i \in \mathcal{G} \wedge y(g_i) = 1\}$. The capacity assigned to these stations can be denoted as variable $c(g_i) \in \mathbb{N}^+, \forall g_i \in \tilde{\mathcal{G}}$. Meanwhile, based on the suggestions from the crowd, we will be mainly concerned about the suggested stations which are in cells in $\tilde{\mathcal{G}}$, and these suggested stations can be represented as $\tilde{\mathcal{S}} = \{s_i | s_i \in \mathcal{S} \wedge f(s_i) \in \tilde{\mathcal{G}}\}$. Multiple suggestions can exist in each cell, and the suggestions received in cell g_i can be represented as $\Gamma_H(g_i) = \{s_i | s_i \in \tilde{\mathcal{S}} \wedge f(s_i) = g_i\}$.

Generally, the more suggestions received from the crowd in certain cells, the more bike usage needs people will have, and the more bikes should be placed at the station in the cell. Based on such an intuition, we can introduce the following crowd suggestion based capacity constraint:

$$c(g_i) \geq c(g_j), \text{ if } |\Gamma_H(g_i)| \geq |\Gamma_H(g_j)|, \forall g_i, g_j \in \tilde{\mathcal{G}},$$

Historical Usage based Capacity Assignment

Besides the suggestion from the crowd, from historical trip records, we can also get information about the travel needs of the public for stations in each cell. As introduced in Section 3.1, for cell $g_i \in \mathcal{G}$, we can represent the trips starting from and ending at the station in g_i as set $\Gamma_T(g_i)$. Generally, the more usages of the station in a cell, the more travel needs people have, and the more bikes should be assigned to the station. Therefore, for all the cells contain the bike stations after re-deployment (i.e., $\tilde{\mathcal{G}}$), we introduce another set of capacity constraint based on the historical bike usage records:

$$c(g_i) \geq c(g_j), \text{ if } |\Gamma_T(g_i)| \geq |\Gamma_T(g_j)|, \forall g_i, g_j \in \tilde{\mathcal{G}}.$$

Adjustment Costs based Capacity Assignment

[5]https://pypi.python.org/pypi/PuLP

[6]http://www.scipy.org/topical-software.html

Similar to placing stations, station capacity assignment will lead to certain construction costs as well. Considering that the costs of assigning bike capacities to brand new stations (as well as moving current stations to new places) have already been counted in the station construction, here, we will be mainly concerned about the costs introduced by adjusting the capacities of the current stations, whose locations are not changed. As introduced in Section 2, the costs introduced by adding/removing bike docks can be represented as cost_d^+ and cost_d^- respectively. Meanwhile, the capacities of current stations in cells $\tilde{\mathcal{G}}$ before and after the system expansion can be represented as $\{\overline{c}(g_i) | g_i \in \tilde{\mathcal{G}}\}$ and $\{c(g_i) | g_i \in \tilde{\mathcal{G}}\}$ respectively, depending on which, we can infer the the capacity changes of the cells in the system expansion:

$$\begin{cases} \text{more bikes added to } g_i, & \text{if } \mathrm{c}(g_i) > \overline{\mathrm{c}}(g_i); \\ \text{station capacity doesn't change at } g_i, & \text{if } \mathrm{c}(g_i) = \overline{\mathrm{c}}(g_i); \\ \text{some bikes are removed from } g_i, & \text{if } \mathrm{c}(g_i) < \overline{\mathrm{c}}(g_i). \end{cases}$$

The costs introduced due to the capacity re-assignment in cell $g_i \in \tilde{\mathcal{G}}$ can be represented as

$$\text{cost}(g_i) = \begin{cases} \text{cost}_d^+ \cdot (\mathrm{c}(g_i) - \overline{\mathrm{c}}(g_i)), & \text{if } \bar{y}(g_i) = y(g_i) = 1, \mathrm{c}(g_i) \geq \overline{\mathrm{c}}(g_i); \\ \text{cost}_d^- \cdot (\overline{\mathrm{c}}(g_i) - \mathrm{c}(g_i)), & \text{if } \bar{y}(g_i) = y(g_i) = 1, \overline{\mathrm{c}}(g_i) \geq \mathrm{c}(g_i); \\ 0, & \text{otherwise.} \end{cases}$$

And the costs introduced by all the cells in $\tilde{\mathcal{G}}$ will be

$$\begin{aligned} \text{cost}(\tilde{\mathcal{G}}) &= \sum_{g_i \in \tilde{\mathcal{G}}} \text{cost}(g_i) \\ &= \sum_{g_i \in \tilde{\mathcal{G}}} \bar{y}(g_i) \cdot y(g_i) \cdot (\text{cost}_d^+ \cdot \max\{\mathrm{c}(g_i) - \overline{\mathrm{c}}(g_i), 0\} + \text{cost}_d^- \cdot \max\{\overline{\mathrm{c}}(g_i) - \mathrm{c}(g_i), 0\}). \end{aligned}$$

Joint Optimization Objective Function

By integrating all the information together, we can obtain the optimal capacity assignment which can minimize the construction costs with the following optimization objective function:

$$\begin{aligned} \min_{\{c_{g_i}\}_{g_i \in \tilde{\mathcal{G}}}} & \sum_{g_i \in \tilde{\mathcal{G}}} \bar{y}(g_i) \cdot y(g_i) \cdot \big(\text{cost}_d^+ \cdot \max\{\mathrm{c}(g_i) - \overline{\mathrm{c}}(g_i), 0\} \\ & + \text{cost}_d^- \cdot \max\{\overline{\mathrm{c}}(g_i) - \mathrm{c}(g_i), 0\}\big) \\ s.t.\ & \mathrm{c}(g_i) \geq \mathrm{c}(g_j), \text{ if } |\Gamma_H(g_i)| \geq |\Gamma_H(g_j)|, \forall g_i, g_j \in \tilde{\mathcal{G}}, \\ & \mathrm{c}(g_i) \geq \mathrm{c}(g_j), \text{ if } |\Gamma_T(g_i)| \geq |\Gamma_T(g_j)|, \forall g_i, g_j \in \tilde{\mathcal{G}}, \\ & \sum_{g_i \in \tilde{\mathcal{G}}} \mathrm{c}(g_i) = C; \mathrm{c}(g_i) \in \mathbb{N}^+, \forall g_i \in \tilde{\mathcal{G}}. \end{aligned}$$

Similarly, the objective function is also a constrained integer programming problem, which can be solved with open-source python toolkits, e.g., PuLP and SciPy. We will not describe the solutions here due to the limited space. By addressing the equation, we can obtain the number of bikes assigned to the stations of each cell in $\tilde{\mathcal{G}}$, which together with cells $\tilde{\mathcal{G}}$ are used as the final output of framework CROWDPLANNING.

4. EXPERIMENTS

To test the effectiveness of the proposed framework CROWDPLANNING in addressing the CSSE problem, extensive experiments have been done on a real-world BSS dataset, Divvy, in the paper. In this section, we will first introduce the datasets used in the experiments, and then talk about the experiment settings in detail. Experiment results and detailed analysis will be provided at the end of the section.

4.1 Dataset Descriptions

Divvy is a BSS launched in the Chicago city, which started to operate on June 28, 2013. Now, Divvy has about 4,760 bicycles at 474 stations within the Chicago city area, and plans to further expand to nearby areas soon. Divvy releases new datasets every two quarters and 5 separate datasets covering the past years are available for download already at its official webpage[7]. These datasets contain both the bike station distribution information, as well as the complete bike trip records. The statistical information about these 5 datasets is available in Table 1 (the station number in these datasets denote the number of available stations at the end of each time range.

Table 1: The Divvy Datasets

datasets	trip	station	bike
2013 Q3-Q4	759,788	300	5,040
2014 Q1-Q2	905,699	300	5,209
2014 Q3-Q4	1,548,935	300	5,040
2015 Q1-Q2	1,096,239	474	8,274
2015 Q3-Q4	2,087,204	474	8,274

Besides the Divvy bike station and trip information, we also crawled all the crowd suggestion data from the site[8] on November 11, 2015. As shown in Table 2, 1,098 new station suggestions together with 775 comments are received from the public. For each suggestion, we can get both the specific location coordinates of the suggested station and the suggestion text information.

Table 2: Crowd Suggestion Dataset

datasets	suggestions	comments
Crowd Suggestion	1,098	775

4.2 Experiment Setting

4.2.1 Experiment Setup

In the experiments, we use the station placement at the end of year 2014 (i.e., stations from dataset "2014 Q3-Q4") as the current stations in the bicycle-sharing system. Meanwhile, considering that Divvy had the expansion in the first half year of 2015, the bike stations at the end of 2015 (i.e., stations from dataset "2015 Q3-Q4") are used as the future bike stations after the expansion, i.e., the ground-truth. The ground-truth will be used for evaluation only and are not involved in the model building. Trips taken by the end of 2014 are used as the historical trips among all the current bike stations, and the posts and comments written by the public online by 2015 Q2 are extracted and used as the effective crowd suggestions, while the remaining trips and posts are not used.

Initially, we obtain the future service region from the future stations (which should be pre-determined in the real scenarios). The crowd suggestions are pruned by removing the suggested stations outside the service region. The service region is divided into cells iteratively, where the distance between the closest bike stations is used as the required granularity of the cells (i.e., parameter $\bar{w}$ used before). In the station deployment step, for each cell in the service region, one unique variable is defined to denote whether there is a station placed in it or not. In addition, based on the current station location, historical usage and crowd suggestion information, we further check the existence of current stations and count up the trips/suggestions taken/suggested in each cell respectively, which

[7]https://www.divvybikes.com/data

[8]http://suggest.divvybikes.com/page/about

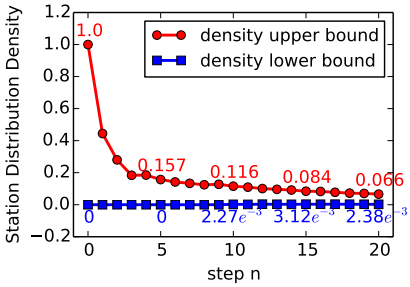


Figure 3: Bike station distribution density.

together with the station construction costs cost_s^+ and cost_s^- will be used to define the station deployment objective function. Considering that more bike stations are added to the system than those removed in the expansion, the cost of adding and removing stations are set as $\text{cost}_s^+ = 10.0$ and $\text{cost}_s^- = 20.0$ in the experiment respectively to model such an operation bias. From the solution, the cells corresponding the variables being assigned with value 1 will be selected to place the bike stations.

Based on the station deployment result, we can obtain the cells being deployed with a station. In the capacity assignment step, for each cell selected in the previous step, one unique capacity variable is introduced to represent the number of bikes being assigned to the station. In addition, we will also count the number of historical trips and station suggestions taken/posted in each cell, which together with the bike dock construction/remove costs $\text{cost}_d^+(1.0)$ and $\text{cost}_d^-(5.0)$ will be used to define the bike capacity assignment objective function. From the solution, the values of these capacity variables will be outputted as the results, denoting the number of bikes assigned to the stations in these cells.

4.2.2 Comparison Methods

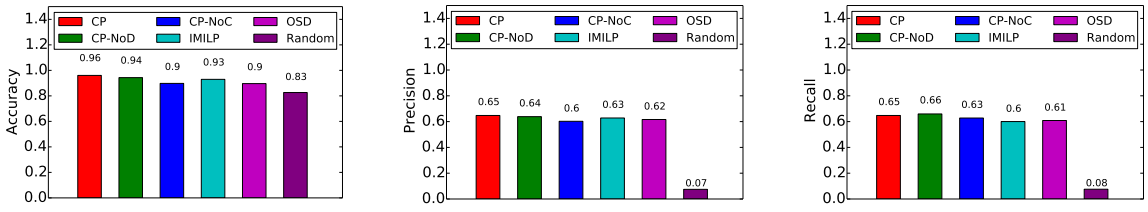
The CSSE problem is a new research problem, and no crowd-planning based planning methods exist that can address it directly so far. To show the advantages of the CROWDPLANNING framework in addressing the CSSE problem, we compare CROWDPLANNING with several traditional planning methods. The comparison methods used in the experiments are summarized as follows:

- CROWDPLANNING: The framework CROWDPLANNING is the method introduced in this paper. CROWDPLANNING infers both the locations and capacities of the stations after the system expansion based on various categories of information, which include (1) the crowd suggestion from the public, (2) the historical bike usages, and (3) system expansion costs.

- CP-NODENS: Density constraint introduced in the paper can effectively constrain the overall distribution of the bike stations. To demonstrate its effectiveness, a variant of framework CROWDPLANNING is used as a baseline method to compare with CROWDPLANNING. The capacity assignment step in CP-NODENS is identical to that of CROWDPLANNING.

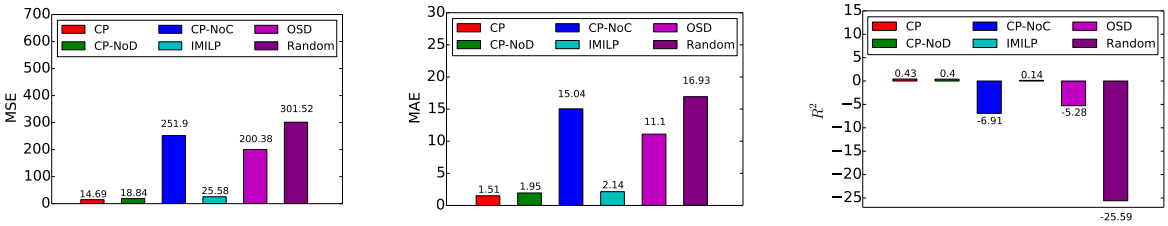
- CP-NOCOST: Construction costs is an important factor affecting the bicycle-sharing system expansion. To support such a claim, we compare CROWDPLANNING with another variant method CP-NOCOST. Method CP-NOCOST assumes the system expansion will introduce no costs at all, and station and bike dock adding/removing costs are all fixed as 0 in CP-NOCOST.

- IMILP: In the experiments, we also extend an existing station placement algorithm, IMILP (Iterative Mixed-Integer



(a) Accuracy (b) Precision (c) Recall

Figure 4: Station deployment result evaluated by Accuracy, Precision and Recall.



(a) MSE (b) MAE (c) R^2

Figure 5: Local station capacity assignment result evaluated by MSE, MAE and R^2.

Linear Program) [7] based station placement, to find the optimal station places which are closest to the suggested places with consideration of the costs. Since no capacity assignment problem is studied in [7], therefore the capacity assignment algorithm in IMILP is identical to that in CROWDPLANNING but merely with the suggestion information only.

- OSD: Based on the historical usage information, we propose to extend a state-of-art station placement algorithm OSD (Optimal Station Deployment) [9] to identify the spots to place the bike stations with consideration of the costs. The capacity assignment algorithm in OSD is identical to that in CROWDPLANNING, but it is based on the historical usage information only.

- RANDOM: The method RANDOM assigns the stations in the cells within the service region randomly. Meanwhile, the capacities of the cells being assigned with stations are random assigned as well.

4.2.3 Evaluation Metrics

To evaluate the performance of different methods, various evaluation metrics are applied in this paper for the station deployment and capacity assignment problems.

In the station deployment, we will obtain the prediction values of the binary variables corresponding each cell in the service region, where the value 1 denotes the station is placed in the cell. Meanwhile, from the ground truth, we can obtain the true values of these variables. Most of the evaluation metrics for traditional binary classification problems can be applied in the station deployment problem, and we will use the F1, Precision, Recall and Accuracy metrics in this paper.

On the other hand, for the cells selected to place the bike stations, we can further infer the capacities assigned to the stations. In addition, from the ground truth, we can obtain the real capacities of these stations as well. Most evaluation metrics for regular regression problems can be used here for the bike capacity assignment problem, and we propose to use the MAE (mean absolute error), MSE (mean square error) and R^2 in this paper.

4.3 Parameter Selection

Bike station distribution density can affect the overall placement of the bike stations a lot. Due to the bike free-ride time is limited in bicycle-sharing systems, the stations cannot be placed too far from each other. Meanwhile, the stations cannot concentrated within a limited area neither. To control the overall distribution of bike stations within the service region, we introduce the distribution density concept in this paper, A group of parameters are introduced to control the overall station distributions densities, i.e., n, $\overline{D}$ and $\underline{D}$ respectively, where n denotes the steps of walking from the centric cells, $\overline{D}$ and $\underline{D}$ represent the upper and lower bounds of the station distribution densities. Before deploying the bike stations, we need to pre-determine these parameters to ensure the CROWDPLANNING framework can work effectively.

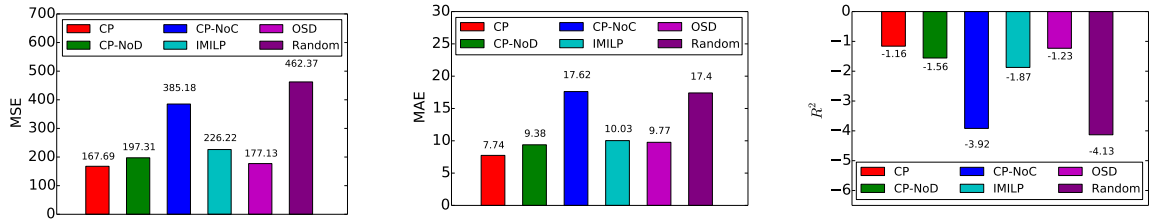
Therefore, we first study the parameters based on the historical bike station distribution. As show in Figure 3, we change the step size n with values in $\{0, 1, 2, \cdots, 19\}$. The bike station distribution upper bound decreases as n increases steadily, while the lower bound increases first and then decreases. The specific values of the upper and lower density bounds achieved at $n \in \{0, 5, 10, 15, 20\}$ are marked on the plot. For instance, the density upper bound achieved at $n = 0$ (i.e., single cell) is 1.0 (if the cell contains a bike station), and the lower bound is 0 (if the cell doesn't contain a bike station). Meanwhile, when n increases to 10 (i.e., the nearby 21^2 cells), the density upper bound decreases to 0.116, while the lower bound increases to 2.27×10^{-3}. In the following experiment, we set $n = 10$ and use 0.116 and 2.27×10^{-3} as the value of parameters $\overline{D}$ and $\underline{D}$ respectively.

4.4 Experiment Result

The experiment results are shown in Figures 4-6, where Figure 4 is about the station deployment results of different comparison methods, while Figure 5 and Figure 6 show the capacity assignment results.

According to Figure 4, the Accuracy score obtained by CROWDPLANNING is still slightly higher than the other methods, which can denote the advantages of CROWDPLANNING compared with these baseline methods. In addition, we can observe that the Accuracy scores obtained by all the comparison methods are all very high (about 0.9). The potential explanation can be: in the service region on the map, by partitioning the region into cells, the majority of the cells are actually have no bike stations. Therefore, inference of the cells to put the stations is actually a class imbalance problem. In such a problem setting, Accuracy cannot evaluate the comparison methods' performance well.

Therefore, the Precision and Recall metrics are also applied in this paper. Meanwhile, in the station deployment problem, the



(a) MSE (b) MAE (c) R^2

Figure 6: Global station capacity assignment result evaluated by MSE, MAE and R^2.

number cells predicted to place the stations and the number of those containing real stations are both K. Therefore, according to the definitions of Recall and Precision, the numerator and the denominator of these two measures are actually equal. Therefore, the corresponding Recall and Precision score (as well as the F1 score, which is not shown) of these different methods are identical. For instance, the Precision and Recall scores achieved by CROWDPLANNING are both 0.69 as shown in the plot, which is much higher than the other baseline methods.

Based on the station predicted to deploy the bike stations, we further assign the bikes to these stations. Merely by treating the cells predicted to place the stations as the domain, we evaluate the bike assignment performance of different comparison methods in Figure 5.

According to the result, CROWDPLANNING outperforms the other comparison methods with great advantages. For instance, the MSE score achieved by CROWDPLANNING is 14.69, which is 22% lower than that achieved by CP-NODENS, 42.57% lower than the MSE achieved by IMILP, and takes only about $\frac{1}{13}$ of the MSE obtained by OSD, $\frac{1}{17}$ of MSE achieved by CP-NOCOST and $\frac{1}{20}$ of RANDOM. Similar results can be observed for the MAE and R^2 metrics. By comparing CROWDPLANNING with CP-NOCOST, the observation that CROWDPLANNING can outperform CP-NOCOST with significant advantages demonstrates the importance of incorporating the cost factors in the system expansion. Meanwhile, compared with the OSD method without utilizing the crowd suggestions, CROWDPLANNING can achieve better bike assignment result, which shows the effectiveness of the crowd suggestions in addressing the CSSE problem. Furthermore, compared with the other comparison methods, better performance of CROWDPLANNING shows the importance of the density constraint and historical usages respectively.

Considering that the predicted cells to place the bike stations are not all correct, and lots of the cells containing real bike stations are pruned in the deployment process and are not actually involved in the capacity assignment result. Therefore, to evaluate the bike assignment of the different comparison methods from a global perspective, we treat all the cells in the service region as the domain, and the performance achieved by different comparison methods are shown in Figure 6. Compared with Figure 5, generally the performance of these comparison methods in Figure 6 are slightly worse, as it incorporates both the mistakes made in the deployment and the assignment at the same time. However, according to the results, CROWDPLANNING can outperform the other comparison methods with great advantages, which have demonstrated the motivations of introducing the framework CROWDPLANNING in the method section.

4.5 Result Analysis

Besides evaluating the comparison methods' performances with these metrics, we also analyze the deployment and capacity assignment results and compare with the historical usages and crowd suggestions in Figures 7.

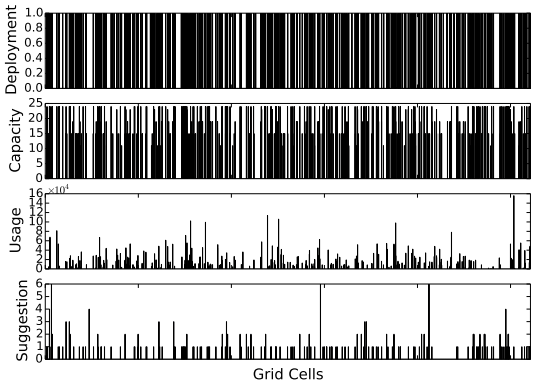


Figure 7: Station Deployment and Capacity Assignment Results Analysis.

In Figure 7, we show the station deployment results and historical bike usages and crowd suggestions for each cell in the service region. In the deployment bar chart, for cells assigned with stations, the bar value is 1, while for the remaining cells, the number will be 0. By enlarging the plot, we can observe that for the cells having very heavy historical bike usages and many suggestions, CROWDPLANNING will place a bike station in the cell. Meanwhile for those having neither usage counts nor suggestions from the public, CROWDPLANNING will not place the station in the cell generally. Similarly, by comparing the number of bikes assigned to each cell with the usage and suggestions, we can observe that generally the number of bike docks assigned to each cell is positively correlated with the historical usages and crowd suggestions in the corresponding cell.

In addition, as shown in Table 1, by comparing the trips taken in 2015 during as well as after the system expansion, we observe that the system expansion can effective increase the bike usages form the public. The total number of trips taken during 2015 (from Q1 to Q4) is $3,183,443$, which is even comparable to the total trips taken within the past one year and a half (from Q3 2013 to Q4 2014). After the expansion, with more bikes and stations available in the extended service region, the number of bike usage has increased to $2,087,204$, which is the highest on record. Considering the high Accuracy, Precision and Recall achieved by CROWDPLANNING compared with the ground truth, the usage record also demonstrates the effective of the expansion strategy indirectly.

5. RELATED WORK

To the best of our knowledge, we are the first to propose to redeploy bike stations in BSS expansion based on the crowd planning. The problem studied in this paper is different from existing works on (1) gas station and hydrogen filling station placement, (2) bicycle-sharing system studies, and (3) urban computing.

Station Placement: Traditional gas station and hydrogen filling station and the recent charging station placement problems have been studied for a long time and dozens of papers have been pub-

lished on these topics so far. Unlike private facility location analysis, the objectives of public facility location (e.g., gas and hydrogen filling stations) analysis are more difficult to embrace and to quantify. Church et al. propose maximal service distance concept in [5] to measure the distance or time that a user need to travel to reach that facility. More specifically, Bapna et al. [3] propose to address the problem of optimally locating gas station facilities for developing nations, which are in the process of converting from leaded to unleaded fuel. Meanwhile, Nicholas gives a case study about the hydrogen station siting and refueling analysis by using the geographic information systems and study the station sitting problem in [12]. In recent years, electric vehicles are becoming more and more popular now. Some of the works have been done on studying the charging station placement problem. Li et al. propose to study the charging station network growth problem for electric vehicles based on the trajectory data of the vehicles in [9].

Bicycle-Sharing System: The bicycle-sharing systems are an important part in urban computing. Many research works have been done on bicycle-sharing systems and other transportation systems to study the system design problem [11], load balance problem [13], and bicycle traffic prediction problem [10]. Lin et al. [11] introduce a strategic design problem for bicycle sharing systems incorporating bicycle stock considerations, which is formulated as a hub location inventory model. Pavone et al. develop methods for maximizing the throughput of a mobility-on-demand urban transportation system and introduce a rebalancing policy that minimizes the number of vehicles performing rebalancing trips [13]. The optimal rebalancing policy can be found as the solution to a linear program effectively in the proposed model. Li et al. propose a hierarchical prediction model to predict the number of bikes that will be rent from/returned in a future period [10].

Urban Computing: In recent years, urban computing has become an emerging research area, and lots of works have been done in the area already. Generally speaking, urban computing aims at improving individuals' living environment in large cities by utilizing the urban sensing, data management and data analysis techniques. Zheng et al. have done lots of works in the area in the past years, and he also provides a review about the urban computing concepts, methodologies, and applications in [17]. Various problems are covered in urban computing, e.g., urban air quality prediction and forecast [18], urban noise diagnosis [19], urban travel and transportation prediction [9].

6. CONCLUSION

In this paper, we have studied the CSSE problem to re-deploy the bike stations within the new service region in the BSSs expansion. Two sub-problems are covered in CSSE simultaneously, which are the station deployment and station capacity assignment problems. A novel station re-deployment framework, named CROWDPLANNING, is introduced in this paper. CROWDPLANNING fuses the information from the crowd suggestions, historical bike usages and station construction costs together and formulates the sub-tasks as optimization problems respectively. Extensive experiments have bee conducted on real-world bicycle-sharing system and the crowd suggestion datasets. The experimental results demonstrate the effectiveness of the CROWDPLANNING framework in addressing the CSSE problem.

7. ACKNOWLEDGEMENT

This work is supported in part by NSF through grants IIS-1526499 and NVIDIA Corporation with the donation of the Titan X GPU used for this research.

This research is also partially supported by the grant from the Natural Science Foundation of China (No. 61303017), the Natural Science Foundation of Hebei Province (No. F2014210068), the Hebei Education Department (No. QN2016083), and the Fourth Outstanding Youth Foundation of Shijiazhuang Tiedao University.

8. REFERENCES

[1] Expansion! woo hoo! http://citibikeblog.tumblr.com/post/101182979267/expansion-woo-hoo. [Online; accessed 30-November-2015].

[2] Tell bay area bike share where you want stations in the tenfold expansion. http://sf.streetsblog.org/category/issues-campaigns/bay-area-bike-share/. [Online; accessed 30-November-2015].

[3] R. Bapna, L. Thakur, and S. Nair. Infrastructure development for conversion to environmentally friendly fuel. *European Journal of Operational Research*, 2002.

[4] D. Brabham. Crowdsourcing as a Model for Problem Solving. *Convergence: The International Journal of Research into New Media Technologies*, 2008.

[5] R. Church and C. ReVelle. The maximal covering location problem. *Papers of the Regional Science Association*, 1974.

[6] J. Howe. The rise of crowdsourcing. *Wired Magazine*, 2006.

[7] A. Lam, Y. Leung, and X. Chu. Electric vehicle charging station placement: Formulation, complexity, and solutions. *CoRR*, abs/1310.6925, 2013.

[8] J. Leimeister. Crowdsourcing: Crowdfunding, crowdvoting, crowdcreation. *Zeitschrift für Controlling und Management (ZFCM)*, 2012.

[9] Y. Li, J. Luo, C. Chow, K. Chan, Y. Ding, and F. Zhang. Growing the charging station network for electric vehicles with trajectory data analytics. In *ICDE*, 2015.

[10] Y. Li, Y. Zheng, H. Zhang, and L. Chen. Traffic prediction in a bike-sharing system. In *SIGSPATIAL*, 2015.

[11] J. Lin, T. Yang, and Y. Chang. A hub location inventory model for bicycle sharing system design: Formulation and solution. *Computers and Industrial Engineering*, 2013.

[12] M. Nicholas. Hydrogen station siting and refueling analysis using geographic information systems: A case study of sacramento county. http://www.its.ucdavis.edu/wp-content/themes/ucdavis/pubs/download_pdf.php?id=199., 2004. [Online; accessed 5-January-2016].

[13] M. Pavone, S. Smith, E. Frazzoli, and D. Rus. Load balancing for mobility-on-demand systems. *International Journal of Robotics Research*, 2012.

[14] H. Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys*, 1984.

[15] J. Zhang, X. Pan, M. Li, and P. Yu. Bicycle-sharing system analysis and trip prediction. In *MDM*, 2016.

[16] J. Zhang and P. Yu. Trip route planning for bicycle-sharing systems. In *IEEE CIC*, 2016.

[17] Y. Zheng, L. Capra, O. Wolfson, and H. Yang. Urban computing: Concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology*, 2014.

[18] Y. Zheng, F. Liu, and H. Hsieh. U-air: When urban air quality inference meets big data. In *KDD*, 2013.

[19] Y. Zheng, T. Liu, Y. Wang, Y. Zhu, Y. Liu, and E. Chang. Diagnosing new york city's noises with ubiquitous data. In *UbiComp*, 2014.