(Re)Configuring Bike Station Network via Crowdsourced Information Fusion and Joint Optimization

Suining He
The University of Michigan–Ann Arbor
suiningh@umich.edu

Kang G. Shin
The University of Michigan–Ann Arbor
kgshin@umich.edu

ABSTRACT

Thanks to their great success as a green urban transportation means of first/last-mile connectivity, bike sharing service (BSS) networks has been proliferating all over the globe. Their station (re)placement and dock resizing has thus become an increasingly important problem for bike sharing service providers.

In contrast to the use of conventional labor-intensive user surveys, we propose a novel optimization framework called `CBikes`, (re)configuring the BSS network with crowdsourced station suggestions from online websites. Based on comprehensive real data analyses, we identify and utilize important global trip patterns to (re)configure the BSS network while balancing the local biases of individual feedbacks. Specifically, crowdsourced feedbacks, station usage history, cost and other constraints are fused into a *joint* optimization of BSS network configuration. We further design a semidefinite programming transformation to solve the bike station (re)placement problem efficiently and effectively. Our evaluation has demonstrated the effectiveness and accuracy of `CBikes` in (re)placing stations and resizing docks based on 3 large BSS systems (with more than 900 stations) in Chicago, Twin Cities, and Los Angeles, as well as related crowdsourced feedbacks.

CCS CONCEPTS

• **Information systems** → *Spatial-temporal systems*; *Data mining*;

KEYWORDS

Bike Sharing, Urban Planning, Urban Computing

ACM Reference Format:
Suining He and Kang G. Shin. 2018. (Re)Configuring Bike Station Network via Crowdsourced Information Fusion and Joint Optimization. In *Mobihoc '18: The Nineteenth International Symposium on Mobile Ad Hoc Networking and Computing, June 26–29, 2018, Los Angeles, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3209582.3209583

1 INTRODUCTION

With the advent of smart cities/communities and Internet of Things (IoTs), the urban sharing economy has been evolving very rapidly. In particular, bike sharing service (BSS) has emerged as one of the most popular and revolutionary powers that change the people's urban life/health. Bike sharing enables the first/last-mile urban travel to be more economic, greener and healthier than traditional gasoline-engine-powered vehicle riding. City transportation also benefits from additional network of bike stations with less hassle of traffic planning.

Experiencing deployment successes and receiving positive feedbacks, many BSS providers have begun expanding their BSS networks. For example, Divvy bicycle sharing program in Chicago, IL is adding another 40 stations and 400 new bicycles in 2017. Meanwhile, Citi Bike in New York City is embracing another 2,000 bikes and 140 stations starting from September, 2017. On the other hand, there exist BSS network shrinkages (at a micro or macro scale) for financial, event, seasonal or meteorological reasons. With dynamic bike usage and complexity of urban environments, how to expand and shrink, or *(re)configure* the existing network of BSS stations becomes increasingly important for the service providers. As stations and bicycles are dynamically added/deleted/resized during the BSS (re)configuration, the station relocation, or "station (re)placement" (*i.e.,* add, move or remove a station), as well as their dock resizing becomes challenging, involving more thorough site investigation and labor-intensive user surveys.

To leverage the collective knowledge from the BSS users [15, 33], service providers have attempted to crowdsource various station placement comments via their own websites. Interested users can easily pinpoint, comment and vote for various potential station locations on an interactive map. This way, the BSS systems can easily and timelily obtain many online feedbacks for their next stage expansion or shrinkage, while reducing their traditional survey and investigation costs significantly.

Despite its importance, however, how to (re)configure the BSS network based on the aforementioned crowdsourced comments is still very challenging and remains an open problem. From the *data* perspective, the first challenge lies in the *heterogeneity* of information inputs. Crowdsourced feedbacks usually provide local, fragmented suggestions due to each individual's limited geographic scope, while network (re)configuration needs global knowledge of user mobility and station-to-station dynamics. How to incorporate the local suggestions/comments together is important and should thus be considered carefully. As all stations are "linked" by users' trips, from the *user*'s perspective, the second challenge stems from their *trip tendency*. Overcrowded or inadequate network placement and ignorance of popular station-station pairs for users' commute may discourage cyclists, thus lowering usage. From the *platform*'s perspective, since web crowds are enabled with large freedom to label locations they want, addressing such naturally-noisy/biased crowdsourced inputs becomes the third challenge, which should be considered by a *joint* fusion formulation.

Mobihoc '18, June 26–29, 2018, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5770-8/18/06…$15.00
https://doi.org/10.1145/3209582.3209583

To address above challenges, we propose CBikes, a **C**rowdsourced **Bike** sharing **S**tation network (re)configuration framework using information fusion and joint optimization. Specifically, CBikes converts BSS network (re)configuration into a graph matching problem. CBikes is a centralized framework and integrates local crowdsourced suggestions with global historical bike usage data upon a geographical map. Each vertex (station) of the graph (network) is matched against this spatially and temporally-varying map of fused knowledge, subject to edges (links) or trips from others. We then formulate a joint optimization problem to balance among crowd satisfaction, platform utility, and (re)configuration cost.

CBikes makes the following major contributions:

- *Comprehensive (re)configuration analysis & data-driven designs*: We analyze extensive real data of several BSS (re)configuration cases. We derive important and practical data-driven designs, including user trip tendencies and inter-station distance constraints, and integrate them in CBikes.
- *Information fusion & joint optimization for BSS (re)-configuration*: We propose a novel optimization framework which jointly considers multi-modal data from crowdsourcing and platform-usage statistics for BSS (re)configuration. We first formulate a grid-based candidate selection and graph matching problem, then transform it into a novel semidefinite programming (SDP) form, and finally solve it efficiently and effectively.
- *Extensive experimental evaluation*: Our CBikes prototype has been evaluated with significant amounts of real data (of more than 900 stations) from 3 premium BSS systems in Chicago, IL, Twin Cities, MN and Los Angeles, CA. These comprehensive studies validate the effectiveness and accuracy of CBikes in optimizing station (re)configuration given crowdsourced inputs.

Despite focus on BSS systems, CBikes can be extended to other sharing/connected vehicle network (re)configuration, including parking lot decisions for car-sharing [35], gas station redeployment [30] and charging station expansion for electric vehicles [17].

This paper is organized as follows. After reviewing related work in Sec. 2, we overview the system framework and important concepts for our problem in Sec. 3. Then, Sec. 4 presents (re)configuration analysis and data-driven designs, followed by the core problem formulation and novel optimization framework in Sec. 5. Sec. 6 provides experimental evaluations, while Sec. 7 discusses some deployment limitations. The paper finally concludes with Sec. 8.

2 RELATED WORK

We briefly review the related work in the areas of urban computing, station placement and bike sharing systems.

Urban computing & information fusion: Urban computing [35] aims to improve social life quality under the trend of speedy urbanization. With faster computing, smarter IoTs and more sensing data, many urban transportation problems have been redefined intelligently and efficiently. CBikes serves as a novel cross-domain knowledge fusion technique [34], unleashing the data-driven and crowdsourcing power [9, 16, 33] to look at traditional site (re)configuration for emerging bike sharing [10, 11].

Site placement & expansion: Due to the recent boom of intelligent transportation, site placement, including gas stations [30], ambulance points [35], and electric vehicle charging docks [17] has been investigated to improve their social and business values.

Note that our work is different from the problems of placing stores [27], gas or electric charging stations [17], since we are given crowdsourced comments and usage statistics from already-deployed stations to (re)configure the BSS network, thus making their initial station placement not directly applicable to our problem. Our joint optimization and crowdsourced fusion are also complementary to emerging urban dynamics [29] and functional zone inference [20], and their studies can be integrated with ours for further refinement of results. Unlike others estimating geographical dependencies of real estate [12], CBikes considers users' trip tendency (pick-up/drop-off) between the bike stations.

Bike sharing systems & services: Recent popularity of BSS has triggered many interesting studies, such as mobility and demand prediction [20, 28, 31], station re-balancing [19], lane planning [5], trip recommendation and station deployment [18, 20]. However, few of state-of-the-art studies considered optimizing the (re)configuration of existing BSS network with crowdsourced knowledge. Orthogonal to the important spatial-temporal modeling for real-time bike demand prediction (including dynamic geographical, meteorological or seasonal factors) [19, 20, 28], CBikes focuses on fusing long-term batched station usage [25, 36] with aggregated crowdsourced feedbacks, for periodic network (re)configurations. Note that our (re)configuration can be done monthly, seasonally or annually subject to the urbanization process, profit, cost and the service provider's own customization.

Many factors may influence the success of (re)configuration [31, 36], including human-built facilities (quality/availability), natural environments (like topography, season or weather [28]), socio-economic or psychological considerations (say, social norms or habits), and utility (cost and travel time). Though it is very challenging to design a complete model, incorporating historical spatial-temporal usages, large-scale crowdsourced preferences and refined cost metric would be a good way to accommodate these factors.

In contrast to recent approaches to BSS deployment [20, 32], we propose a generic optimization framework that accommodates both network expansion and reduction using data-driven designs and novel semidefinite programming [7]. CBikes adopts a flexible formulation fusing crowdsourced knowledge with historical usage statistics *jointly*, and accounts for interactions of users and stations, thus adapting much better to complex station correlations.

Our study is also orthogonal to emerging station-free BSS systems [5]. CBikes can be used for station-free BSS if each parked bike is considered a "dock-less station". However, as unregulated parking may still prevent its wide acceptance by social-norm, we will not consider it any further in this paper.

3 SYSTEM & CONCEPTS

We present the basic CBikes framework (Sec. 3.1) and introduce important definitions of CBikes (Sec. 3.2).

3.1 System Framework

Fig. 1 shows the components and layers of CBikes. Specifically, CBikes consists of 4 consecutive layers for computing bike station (re)configuration: input, design, core and action layers. At the *input* layer (Sec. 4.1), historical station-usages, crowdsourced feedback of station expansion/shrinkage suggestions, as well as predefined costs are collected and delivered to a central server, pre-processed and then stored into databases. Note that other practical geographic

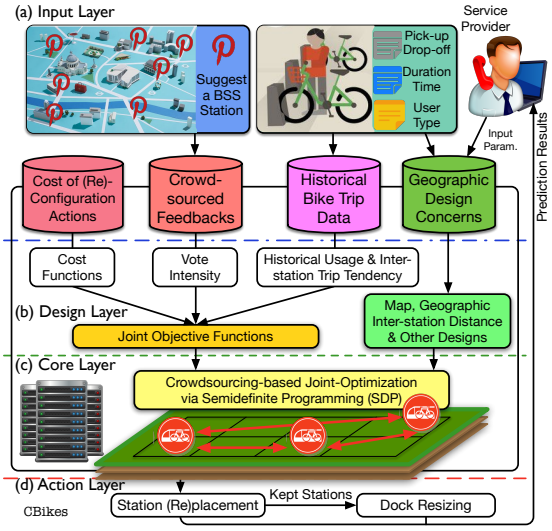

Figure 1: The system framework flow of CBikes.

design concerns or constraints, including the number of service bikes and accessible station deployment areas, are also inputted by the service provider, processed and stored. Our focus here is to develop a generic optimization framework, given the above primary and secondary information.

At *design* layer (Sec. 4), we then form the joint objective functions, and integrate map information and station geographic distances into constraints. Finally, we formulate a joint optimization framework, transform and solve it at *core* layer (Sec. 5), optimizing station sites with respect to predefined map grids. Guided by the results of the *action* layer, the service provider may (re)place stations and resize their docks. In case results are not satisfactory, the parameters can be tuned interactively for another optimization trial.

3.2 Important Concepts

We elaborate on the important terms, concepts or definitions for our mathematical formulation. Formally, we have

Definition 1. *Bike station network (BSN): Each station i is represented by $\mathcal{S}_i = (lat_i, lon_i, \kappa_i)$ ($i \in \{1, \ldots, M\}$), where tuple $[lat_i, lon_i]$ denotes its geographic coordinates and κ_i is its capacity ($\kappa_i \geq 0$). Denote the location of each $\mathcal{S}_i$ as a 2×1 vector $\boldsymbol{l}_i = [lat_i, lon_i]^T$, and let $\mathbf{L} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_M]^T$ be the M×2 coordinate matrix of all stations on the map. Given a set of M geographical nodes $\mathbf{L}$ and their links $\mathbf{E} \subseteq \mathbf{L} \times \mathbf{L}$ connecting them, a network of BSS stations is represented by a graph $\mathcal{G} = (\mathbf{L}, \mathbf{E})$.*

Given an already-deployed BSN, after a certain period we obtain

Definition 2. *Historical bike trip data: Each trip corresponds to a user's bike ride which happens at a certain time from a station to another. Specifically, a set of bike trips from a start station $\mathcal{S}_i$ to an end $\mathcal{S}_j$ can be represented as $\boldsymbol{\tau}(i,j) = \{i, j, (t_i, t_j)'s\}$, where $(t_i, t_j)'s$ are the set of pick-up/drop-off timestamps of each trip in $\boldsymbol{\tau}(i,j)$. Note that $\boldsymbol{\tau}(i,j)$ is symmetric if and only if riders return their bikes at the same station as they were rented, i.e., $\boldsymbol{\tau}(i,j) = \boldsymbol{\tau}(j,i)$ iff $i = j$.*

Based on the deployment results the service provider may initiate

Definition 3. *Bike station network (re)configuration (BSNR): A phase of BSNR basically consists of station (re)placement and dock resizing. At each BSNR, the service provider can place new stations, remove or move existing ones, or just keep them, and resize the docks. We consider two consecutive stages of a BSNR, i.e., two sets of station status before and after a (re)configuration. For ease of description, denote the $\widetilde{M}$ stations before BSNR as $\widetilde{\mathcal{S}}_i$'s, and let the old (prior to the (re)configuration) network be $\widetilde{\mathcal{G}} = (\widetilde{\mathbf{L}}, \widetilde{\mathbf{E}})$. Each $\widetilde{\mathcal{S}}_i$'s location before BSNR is denoted as $\widetilde{\boldsymbol{l}}_i = [\widetilde{lat}_i, \widetilde{lon}_i]$, with the pre-(re)configured capacity $\widetilde{\kappa}_i$. At each BSNR, we consider (re)placing M stations and resizing the dock capacity to accommodate a total of $\mathcal{K}$ bikes.*

BSNR decisions should also involve public engagement and cater to users' demand. Before a BSNR, via certain media/platform (like a website) interested users may easily suggest station sites, *i.e.,*

Definition 4. *Crowdsourced station feedbacks: Each feedback indexed by n on the interactive map is represented as $f_n = (lat_n, lon_n, t_n, text_n)$, where the pair (lat_n, lon_n) is the location/site coordinate, t_n is its timestamp, and $text_n$ is the related posted comment, if any.*

We briefly introduce the actions of BSNR. Station (re)placement is to find their appropriate locations. As searching in continuous geo-space may lead to a computation complexity problem, we discretize the entire map into multiple grids. This way, we have finite candidate sets for efficient computation, whose granularity can be determined via task customization [8, 20]. Formally, we have

Definition 5. *Station (re)placement grid: The entire city map is discretized into a set of R regular grids (rectangle grid in our case), i.e., $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_R]^T$, an R×2 matrix where each grid is given by a coordinate (2×1 vector) of its center, $\mathbf{g}_r = [lat_r, lon_r]^T$ ($r \in \{1, \ldots, R\}$).*

After station (re)placement, CBikes further resizes their docks.

Definition 6. *Dock resizing: The total dock capacity equals (or at least) the total number of bikes, i.e., $\sum_{i=1}^{M} \kappa_i = \mathcal{K}$. CBikes resizes the dock κ_i (enlarge, decrease or maintain) at each station i to satisfy both incoming crowdsourced needs and historical demands.*

Note that the cost of dock resizing only considers those stations staying at the same locations as in $\widetilde{\mathcal{G}}$. Dock-related costs of other newly-added/removed stations are included in their subtotals of creation and removal.

Profit, cost and station usage are critical from the platform perspective, while matching request and convenience may matter to the users. To accommodate both, we study in this paper:

Definition 7. *Crowdsourcing-based BSNR (CBSNR): Given historical bike trip data, crowdsourced feedbacks, cost of actions, and other practical BSS design constraints, CBSNR problem is to (re)configure the existing network to jointly match crowds' feedbacks and station usage statistics at minimum cost.*

4 (RE)CONFIGURATION ANALYSIS & DESIGN

The inherent complexity of CBSNR calls for careful and practical designs based on usage data and users' feedback. Via comprehensive analysis of real data (Sec. 4.1), we present important designs for CBikes, *i.e.,* historical station usages (Sec. 4.2), inter-station trip tendency (Sec. 4.3), geographic distance constraint (Sec. 4.4), and finally crowdsourced feedbacks (Sec. 4.5). For each of these, we make the important observations from the data (of periods before (re)configuration), and present a quantitative design formulation.

4.1 Overview of Datasets Studied

We consider the following BSS data (including map information) for our CBSNR analysis here and evaluation in Sec. 6:

- *Divvy at Chicago, IL*, which consists of total 582 stations by 2017 (2^{nd} quarter). 3 major expansions with total 282 new stations were recorded since 2013. Overall, 11,544,750 trips are studied.

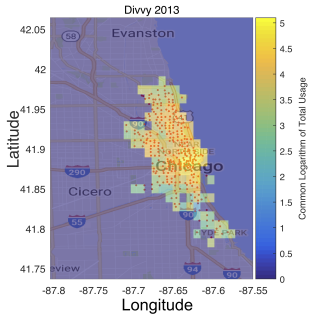

Figure 2: Distribution of total usage in Chicago 2013.

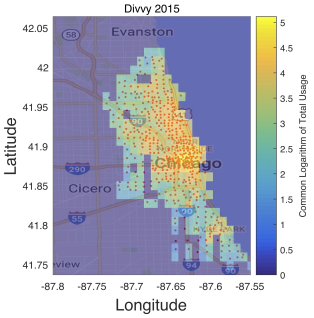

Figure 3: Distribution of total usage in Chicago 2015.

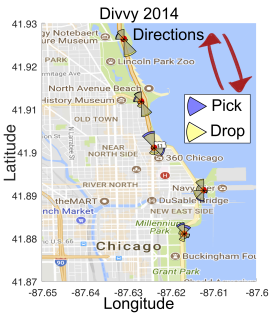

Figure 4: Flow directions of 5 stations in Chicago 2014.

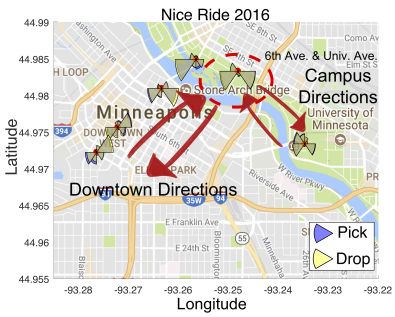

Figure 5: Flow directions of 6 stations in Minneapolis, MN 2016.

- *Nice Ride at Twin Cities, MN*, which includes a total of 202 stations in Minneapolis-St. Paul Metropolitan area until 2016. 5 major expansions with 134 new stations are recorded since 2013. Overall, 2,857,027 trips are analyzed.
- *Metro Bike at Los Angeles County, CA*, which consists of total 119 stations in Los Angeles (LA) County by 2017 (3^{rd} quarter). 2 major network expansions with total 56 new stations are recorded since 2016. Overall, 277,195 trips are evaluated.

This massive trip data includes start/destination stations, related pick-up/drop-off timestamps (or trip durations), user type (say, day-pass holders or annual subscribers) or even age/gender/birthday information. We further scrape the crowdsourced feedbacks from "*Suggest a Station*" website of each BSS provider [2–4]. For each CBSNR, we use the 1,100 latest feedbacks f_n's with $[lat_n, lon_n]$'s (with t_n before the BSNR). As most observations are qualitatively similar, we focus on Divvy and Nice Ride.

4.2 Historical Usage at Each Station

Observation: Intuitively, the more often a station was used at a certain location, the more likely it is preserved there. We first briefly summarize the spatial station usage *w.r.t.* BSNR. Figs. 2 and 3 visualize the spatial distribution of usage in a "heat-map" form. The warmer the color, the more pick-up/drop-off events are recorded ($\log_{10}$(usage)). Due to BSNR, clear configuration changes can be seen between 2013 and 2015. More city areas are covered, and higher usages can be observed among the points of interests (including the skyline and lake coast) in Chicago as the network expands. Similar patterns can be observed from Twin Cities and LA County.

Design: To better differentiate historical usages of different stations, we design a usage-related measure for each $\mathcal{S}_i$ *w.r.t.* each $\mathbf{g}_r$. Let $\mathrm{T}_r = \{\boldsymbol{\tau}(i,j) | (\mathcal{S}_i \text{ is at } \mathbf{g}_r) \bigcup (\mathcal{S}_j \text{ is at } \mathbf{g}_r)\}$ be the aggregated set of trips starting or ending at grid r. We define the *usage importance* of $\mathbf{g}_r$ for a station location candidate $\boldsymbol{l}_i$ as

$$\mathcal{U}_r^i \triangleq \frac{\exp(\lambda_r^i |\mathrm{T}_r|)}{1 + \exp(\lambda_r^i |\mathrm{T}_r|)}, \tag{1}$$

where $\lambda_r^i = \left(\widetilde{\boldsymbol{l}}_i \cdot \mathbf{g}_r\right) \Big/ \left(\left\|\widetilde{\boldsymbol{l}}_i\right\| \cdot \|\mathbf{g}_r\|\right)$. Here $0 < \lambda_r^i \leq 1$ characterizes the normalized affinity or closeness of station i with grid r in previous geographic space, *i.e.,* the closer $\mathcal{S}_i$ was with $\mathbf{g}_r$ before BSNR, the larger λ_r^i gets. We consider $0 < \mathcal{U}_r^i < 1$, the scale of which can be easily integrated with other formulations, and the exponential function strengthens the effect of large usage and physical closeness. Clearly, the more station i is used at grid r, the larger $\mathcal{U}_r^i$ is, and the more likely its location is kept or (re)placed there.

4.3 Inter-station Trip Tendency

Observation: Despite its importance, considering total usage only may not be sufficient. For example, a BSS user may frequently commute between a pair of stations (say, her/his home and office or school). Individually considering each station without inter-station trip tendency may overlook such frequently commuting users (which yields a stable platform income) and remove those stations having strong links $\mathbf{E} \subseteq \mathbf{L} \times \mathbf{L}$ with others.

To further illustrate this, Fig. 4 shows an example of trip tendency among 5 stations in Chicago in 2014. We summarize their pick-up/drop-off flows *w.r.t.* each outgoing/incoming direction (*i.e.,* a vector between start and destination). Dark blue sectors indicate the volume of outgoing bike flows while light yellow represents incoming bikes. Volumes in all directions are normalized to $[0, 1]$ for each $\mathcal{S}_i$. The larger radius of a sector, the more proportion of its bike flows start or end in that direction. We can observe that a strong north–south trip pattern *w.r.t.* stations along Lake Michigan beaches mainly because the tourists' recreational rides create a large trip tendency at stations along the lake shore.

Similarly, Fig. 5 shows the trip tendency to/from several stations in Minneapolis, MN. We can see strong bike flows between west downtown and university area, indicating bike commutes by students, staff and faculty. In particular, we can observe significant south–west and south-east flows at the station of 6th Ave. SE & University Ave. (circled), which likely bridges the downtown and campus. Despite its less total usage (lower $\mathcal{U}_r^i$ in Eq. (1)) than others, CBSNR should also value importance of this station.

In summary, inter-station trip tendency is highly correlated with purposes of users' trip choice (*start*, *end*), including commutes between home and school or recreational sightseeing. Further, its strength characterizes the volume/tendency of urban flows. Therefore, we incorporate the tendency in our optimization model.

Design: Recall that $\boldsymbol{\tau}(i,j)$ represents the set of bike trips from $\mathcal{S}_i$ to $\mathcal{S}_j$ ($i \neq j$). To focus on the connectivity and trip-tendency, we adapt the *link probability* in theories of network embedding [23], and define a new *tendency metric* $p(i,j)$ between $\mathcal{S}_i$ and $\mathcal{S}_j$ as

$$p(i,j) = \left(1 + \exp\left(-\vec{a}_i^j \cdot \vec{a}_j^i\right)\right)^{-1}, \tag{2}$$

where the vector $\vec{a}_i^j$ represents the proportion of trips from i to j, *i.e.,* $|\boldsymbol{\tau}(i,j)|$, as well as that of the remaining trips, *i.e.,*

$$\vec{a}_i^j = \left[\frac{|\boldsymbol{\tau}(i,j)|}{\sum_{k=1,k\neq i}^{M} |\boldsymbol{\tau}(i,k)|}, 1 - \frac{|\boldsymbol{\tau}(i,j)|}{\sum_{k=1,k\neq i}^{M} |\boldsymbol{\tau}(i,k)|}\right], \tag{3}$$

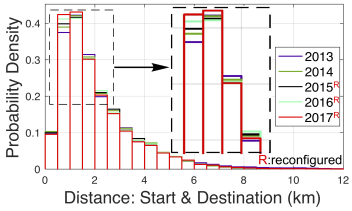

Figure 6: Trip dist. distributions *w.r.t.* years (Divvy), with [0.5km, 2.5km] zoomed in.

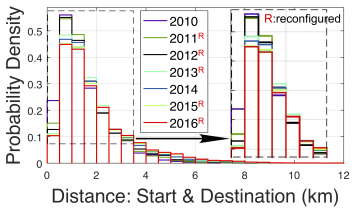

Figure 7: Trip dist. distributions *w.r.t.* years (Nice Ride), with [0.0km, 3.5km] zoomed in.

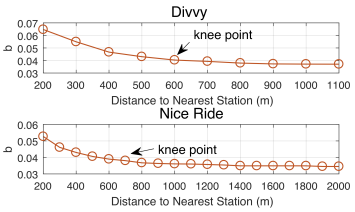

Figure 8: Regression parameter b vs. distance to the nearest station (Divvy & Nice Ride, 2016).

and similarly for $\vec{a}^i_j$. Note that $p(\cdot,\cdot)$ is symmetric, *i.e.,* $p(i,j) = p(j,i)$. In other words, the larger proportion of bikes are commuting between stations i and j, the larger $p(i,j)$ is $(0 < p(i,j) < 1)$, implying more important connectivity of these two stations. Then, we find $\sum_{j=1,j\neq i}^{M} p(i,j)$ for each $\mathcal{S}_i$, further indicating its overall connectivity with other stations. This way, we may characterize the complex network structure efficiently [23], highlighting the connectivity and trip-tendency between stations. Considering the frequent usage and travel patterns of bike users, BSNR should preserve interactive connectivities between these stations.

From the data management's point of view, the total usage and the trip tendency of stations are inherently correlated, as the former is the result of aggregating the latter. To highlight station connectivity and mitigate inherent redundancy, as shown in Eq. (3) we normalize the usage in the model. Besides, our evaluation (Sec. 6) shows that inclusion of tendency beyond usage improves the performance, which has not yet been considered in previous siting studies [8, 18, 20].

4.4 Geographic Inter-station Distance

The BSS is designed to provide first-/last-mile commute, and a user is allowed to return the bike at any station near her/his destination. Thus, the density of deployed stations is a critical design consideration, *i.e.,* the network should be neither too dense nor too sparse.

Observation 1: We first overview the histograms of outgoing trip distances, which characterize the tendency of a user when deciding on a trip. We do not show round trips as they are included in single station usage (Sec. 4.2). Figs. 6 and 7 show the outgoing trip distance distribution *w.r.t.* years for each BSS system. We can observe that a clear "last-mile" traffic flow, *i.e.,* more than 65% outgoing users tend to drop off bikes within 2km (around 1.5miles).

Interestingly, as BSS expands, increasingly more percentage (88% in 2013→90% in 2016) of users take short-distance (<4km) trips in Chicago while in Twin Cities, this part is decreasing (97.34% in 2010→93.3% in 2013→89.81% in 2016). It is likely due to the difference in network density. With markedly more nearby stations and available bikes, it is more convenient for Chicagoans to ride between near stations. For Nice Ride, as average distance to nearest station is larger (0.47km in Divvy vs. 0.58km), under such nearby stations of a sparser network may take less usage percentage.

Unlike its peers, Metro Bike in LA County is distributed in LA, Santa Monica, Pasadena and Long Beach. Distances between nearest stations are much smaller within each city (often 0.25km~0.39km), showing much denser urban networks. Hence, much more short-distance trips are expected.

Observation 2: We also show the bike usage of each station versus the distance to its nearest neighbor. This way, we can characterize the impact between stations due to service coverage overlap. Specifically, we conduct negative binomial regression (NBR) [14] on single station usage |T| (the number of trips) against different distances $\mathcal{D}$ (m) to the nearest peers. Considering the probability $\mathcal{P}(|\mathrm{T}| = a|\mathcal{D}) = (e^{-z}\cdot z^a)/(a!)$ and mean of |T| is z [14], NBR finds the set of b's which maximize the log-likelihood for $\ln z = b_0 + b\mathcal{D}$.

Fig. 8 shows the regression parameter b versus $\mathcal{D}$. b characterizes sensitivity of station usage towards network density. Overall, we observe in both systems a positive effect ($b > 0$) of the distance to the nearest neighbor over the station usage, implying that usage generally increases with distance from the nearest neighbor. A strong counter-effect upon a station can be inferred within a close distance from others (say, less than 400 or 500m) which may lower its usage. It is mainly because of a competitive effect [25] that close-by stations may serve the same group of users and prevent each other from being fully utilized. As a short-range effect, it saturates quickly after a certain range (say, 600m in Divvy and 700m in Nice Ride), due to discouraged usage of distant sites.

Design: To reflect the above observations, over $\mathbf{E} \subseteq \mathbf{L}\times\mathbf{L}$ we set lower/upper bounds $[\underline{d}_{ij}, \overline{d}_{ij}]$ for the distance between two neighboring stations $\mathcal{S}_i$ and $\mathcal{S}_j$ (in a neighborhood set $\boldsymbol{\mathcal{N}}$), *i.e.,*

$$\underline{d}^2_{ij} \le \|\boldsymbol{l}_i - \boldsymbol{l}_j\|^2 \le \overline{d}^2_{ij}, \quad \forall i \neq j, (i,j) \in \boldsymbol{\mathcal{N}}, \tag{4}$$

We apply heuristic local search [6] around all $\mathcal{S}_i$'s in $\mathcal{G}$ based on historical usage statistics, crowd feedbacks or their fused map (Sec. 5.2) to determine a rough neighborhood set of $\boldsymbol{\mathcal{N}}$. As `CBikes` is a general framework, geographic distances other than the Euclidean metric (like the Manhattan distance for metropolitan cities like New York City [31]) can be easily applied. Note that we consider locally constraining neighboring station candidates in close grids (say, within 2 to 3 grids), making differences of metrics rather small in practice. Similar to many state-of-the-art studies [20, 32], for prototype and illustration purposes we consider the Euclidean distance here.

For convenience and utility, the upper bound caters to the majority of travel distance preferences, while the lower bound mitigates conflicts between neighboring stations. We consider distance at the 65-percentile of cumulative usage distributions from Figs. 6 and 7 for $\overline{d}_{ij}$, and distance at the "knee point" (where the plotted curve "turns", or formally where a curve is best approximated by a pair of lines) in Fig. 8 for $\underline{d}_{ij}$. Note that all derived parameters for each test are only based on periods before (re)configuration takes place. Despite the global bound setting here, one may easily customize $[\underline{d}_{ij}, \overline{d}_{ij}]$ further *w.r.t.* each station pair.

In summary, including links of stations (including inter-station trip tendency and distance) is important as simple scalar quantification and local feedbacks of crowds who have limited scopes may ignore the actual trip tendency. Their introduction helps assist the

global optimization, and we will further validate their importance and effectiveness via evaluation of real data (Sec. 6).

4.5 Crowdsourced Feedbacks

Observation: Crowds are essential to CBSNR, and Fig. 9 visualizes the spatial distribution ("heat-map") of aggregated crowd feedbacks before BSNR. The warmer color means more feedbacks. We also plot the initial station locations in 2013 (before expansions). From the spatial distribution of crowdsourced feedbacks, we may observe that strong sociodemographic factors. For example, many suggestions are made to the central business district and skyline (say, Magnificent Mile) of Chicago, matching intensive commuting needs there. Besides, anticipation also comes from south and west, probably due to student commuter demands around the university campus and introduction of metro stations. We also observe similar patterns in feedbacks of the other two systems. The crowdsourced feedbacks have potential and power in identifying latent factors (qualitatively and quantitatively) for network (re)configuration, and serve as an important supplement to many other GIS databases [36].

Note that the local and dispersed crowds' feedbacks could not always directly reveal the overall trip tendency connecting the start and the destination, mainly because each individual usually recommends new stations closest to either her/his own work place or residence. Besides, one may not reveal both the start and end of each trip due to his privacy and identity concerns. The global inter-station trip tendency has been modeled in our optimization to account for the above biases or insufficiency.

Pre-processing the crowdsourced data is essential. For example, we have noticed and filtered out some hilarious input locations in Lake Michigan for Divvy. Via comprehensive map boundary and building constraints, we can easily identify those noisy feedbacks. As users may vote for more reasonable labels for themselves, and `CBikes` jointly considers historical usage and geographic constraints, these noisy inputs can be suppressed further.

Design: Given Defs. 4 and 5, we consider crowds' feedbacks in a discretized manner, *i.e.,* we aggregate the number of feedbacks f_n's falling into each rectangle grid. Intuitively, the more crowdsourced pin-points go into a grid, the more likely it would be selected. This way, we consider the aggregated feedbacks $\mathcal{V}_r$ for each $\mathbf{g}_r$, and define a measure of *vote intensity* as a penalty function $\phi(\mathcal{V}_r)$ for our optimization input. A larger $\phi(\mathcal{V}_r)$ due to more votes implies a heavier "penalty" to be minimized by the solver. Specifically, given input $|\mathcal{V}_r|$ votes at $\mathbf{g}_r$, we have

Definition 8. *Deadzone-linear penalty (DLP): the DLP function with a deadzone width $\beta \geq 0$ is given by*

$$\phi(\mathcal{V}_r) = \begin{cases} 0 & : if \quad |\mathcal{V}_r| \leq \beta; \\ |\mathcal{V}_r| - \beta & : if \quad |\mathcal{V}_r| > \beta. \end{cases} \tag{5}$$

In other words, our DLP de-emphasizes the grids with crowds' votes less than β, mitigating outlier effect, and focuses on others with more support, which is also reasonable in traditional user surveys for BSS expansion [15, 21, 25]. Using a linear $|\mathcal{V}_r| - \beta$, `CBikes` also mitigates sensitivity towards large but noisy votes than other higher-order penalty functions [7]. After calculating for all $\mathbf{g}_r$'s, we normalize each $\phi(\mathcal{V}_r)$ ($r \in \{1, \ldots, R\}$) into the range $[0, 1]$.

In summary, as a joint optimization framework, `CBikes` fuses heterogeneous sources of information and data-driven designs, instead of single-point knowledge input, for final joint decisions, thus mitigating the noisiness of crowd feedbacks. The effectiveness of our proposed information fusion will be validated in Sec. 6.

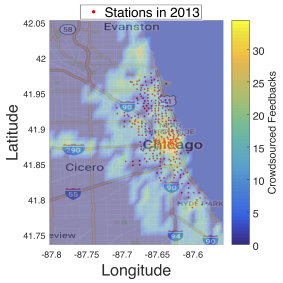

Figure 9: Crowd feedback distribution, and station locations in Chicago 2013.

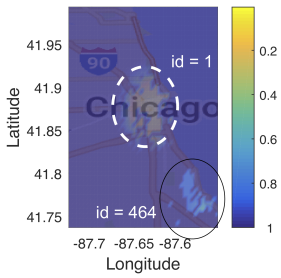

Figure 10: Spatial distribution of Δ_r^i's for two selected stations of Divvy.

5 CORE FORMULATION & METHODOLOGY

We present the problem formulation to integrate the above designs. We first present the grid matching basics (Sec. 5.1), and then provide objective functions (Sec. 5.2). We then discuss the formulation (Sec. 5.3), followed by semidefinite programming transformation (Sec. 5.4). We finally give a complexity analysis (Sec. 5.5).

5.1 Station (Re)Placement & Grid Matching

Station (re)placement is more challenging than dock resizing. We convert the BSS (re)placement problem to the problem of estimating affinity (closeness) of each station with predefined geographic grids. Each $\mathcal{S}_i$'s location is considered as the weighted average of grid coordinates (Def. 5). Consider M stations are to be (re)placed. Let h_r^i be the weight of grid r in determining $\mathcal{S}_i$'s location $\boldsymbol{l}_i$, *i.e.,*

$$\boldsymbol{l}_i = \sum\nolimits_{r=1}^{R} h_r^i \mathbf{g}_r, \quad \forall i \in \{1, \ldots, M\}, \tag{6}$$

where each h_r^i follows normalization and nonnegative constraints,

$$\sum\nolimits_{r=1}^{R} h_r^i = 1, \quad h_r^i \geq 0, \quad \forall r \in \{1, \ldots, R\}. \tag{7}$$

For ease of presentation, we define $\mathbf{H}$, an $M \times R$ matrix consisting of all h_r^i's. Then, the set of location coordinates of all stations is

$$\mathbf{L}_{M\times 2} = \mathbf{H}_{M\times R}\mathbf{G}_{R\times 2}. \tag{8}$$

We want to determine the grid weights for station (re)placement.

5.2 Objective Function Design

To incorporate heterogeneous sources of data, we present a novel information-fusion technique in our joint optimization. Specifically, we present the joint difference functions fusing crowds and historical usage, and the cost measures for (re)configuration actions. Combining these leads to our final objective function.

Metric of joint difference: To quantify the matching of knowledge fusion, we further design a *generic* metric, *i.e., joint difference of grid matching*, denoted as Δ_r^i, for each candidate station i at a grid r. Specifically, given V feature metrics $F_v(i, r) \geq 0$ showing the fitness of matching, we may define $\Delta_r^i \triangleq \left(\prod_{v=1}^{V}(1 + F_v(i, r))\right)^{-1}$. In our prototype, $F_v(i, r)$'s come from available historical usage (Secs. 4.2 & 4.3) and crowd feedbacks (Sec. 4.5), *i.e.,*

$$\Delta_r^i \triangleq \frac{1}{\left(1 + \mathcal{U}_r^i\right)\left(1 + \sum_{j=1, j\neq i}^{M} p(i, j)\right)\left(1 + \phi(\mathcal{V}_r)\right)}. \tag{9}$$

The inverse function in Eq. (9) means that the more historical usage $\mathcal{U}_r^i$, total trip tendency $\sum_{j=1, j\neq i}^{M} p(i, j)$ and votes $\phi(\mathcal{V}_r)$, the smaller Δ_r^i and the more favored $\mathbf{g}_r$ for $\mathcal{S}_i$. It guarantees $0 < \Delta_r^i \leq 1$, and adapts to cases of either with little historical usage or few

crowds' votes (say, any $F_v(i,r) \to 0$). For those grids without any of the above knowledge, one may conduct further spatial-temporal prediction (or extrapolation) [24] on $F_v(i,r)$ based on nearby peers.

We also illustrate and visualize the spatial distribution of *joint difference* $\boldsymbol{\Delta}_r^i$'s in Eq. (10), *i.e.,* "heat map" of fused knowledge. Fig 10 shows $\boldsymbol{\Delta}_r^i$'s of two station candidates in Divvy (dashed circle: $id = 1$; solid circle: $id = 464$). The warmer the color, the smaller the $\boldsymbol{\Delta}_r^i$, indicating a higher matching potential there for that station.

Note that for further grid differentiation, the joint difference modeling in Eq. (9) is general to be integrated with other external information (other feature metrics $F_v(i,r)$'s) if available, including distance to the central business district, closeness to rail stations and other interesting sociodemographic factors (estate price, income or point of interest number) [25, 36] affecting the station functionality.

Given the joint difference for each station, we further look at the entire network. Let $\boldsymbol{\Delta}$ be an $M \times R$ matrix consisting of all $\mathcal{S}_i$'s joint differences. We define an operator $\psi(\mathbf{H}, \boldsymbol{\Delta})$ returning *sum of entry-wise products* of elements in matrices $\mathbf{H}$ and $\boldsymbol{\Delta}$, or formally, the *trace* (denoted as $\mathrm{Tr}(\cdot)$) of product $\mathbf{H}\boldsymbol{\Delta}^T$. Then, the *total joint difference* of CBSNR estimates and the map of fused knowledge is

$$\psi\left(\mathbf{H}_{M\times R}, \boldsymbol{\Delta}_{M\times R}\right) \triangleq \mathrm{Tr}(\mathbf{H}\boldsymbol{\Delta}^T) \triangleq \sum\nolimits_{i=1}^{M}\sum\nolimits_{r=1}^{R} h_r^i \Delta_r^i. \tag{10}$$

Specifically, the smaller the Δ_r^i, the higher h_r^i assigned to $\mathbf{g}_r$, and the more likely $\mathcal{S}_i$ is (re)placed there (Eq. (6)), *i.e.,*

$$h_r^i \geq h_q^i, \quad \text{if } \Delta_r^i \leq \Delta_q^i, \quad \forall r \neq q \in \{1, \ldots, R\}, \quad \forall i. \tag{11}$$

Cost of station (re)placement: Considering the feasibility of CBSNR, we integrate the estimates of potential (re)placement cost. Let $c_\circ \geq 0$ and $c_\times \geq 0$ be the costs of adding and removing a station, respectively (customizable *w.r.t.* each $\mathbf{g}_r$ and each $\mathcal{S}_i$). The move action is considered as a removal followed by an add. Then, we define the costs of all actions for each $\mathcal{S}_i$ at $\mathbf{g}_r$ as:

$$\theta_r^i = \begin{cases} 0 & \text{: if no action is imposed;} \\ c_\circ & \text{: if a new station is added;} \\ c_\times & \text{: if an existing station is removed;} \\ c_\times + c_\circ & \text{: if a station is moved to other place.} \end{cases} \tag{12}$$

Recall that we consider $\boldsymbol{l}_i = \sum_{r=1}^{R} h_r^i \mathbf{g}_r$, the weighted average of closely-matched grids. For existing stations, let $\widetilde{h}_r^i = 1$ if $\widetilde{\mathcal{S}}_i$ was at $\mathbf{g}_r$ and $\widetilde{h}_r^i = 0$ vice versa. For newly-added ones, $\widetilde{h}_r^i = 0$, for $\forall r$. Increasing or decreasing h_r^i at grid r implies a higher potential of adding or removing $\mathcal{S}_i$. To fit these in our formulation, we characterize the two changes for each θ_r^i as

$$\left(h_r^i\right)_\circ = \max\left\{h_r^i - \widetilde{h}_r^i, 0\right\}, \quad \left(h_r^i\right)_\times = \max\left\{\widetilde{h}_r^i - h_r^i, 0\right\}. \tag{13}$$

Then, we set the total cost of (re)placing all M stations in R grids as

$$C^* \triangleq \sum_{i=1}^{M}\sum_{r=1}^{R} \theta_r^i = \sum_{i=1}^{M}\sum_{r=1}^{R} \left(\left(h_r^i\right)_\circ \cdot c_\circ + \left(h_r^i\right)_\times \cdot c_\times\right). \tag{14}$$

Cost of dock resizing: Let $M' \leq M$ be the number of stations staying at their same locations without (re)placement (moved/removed). Recall in Def. 6, dock resizing considers only the cost of these M' stations, where each resizing action for an $\mathcal{S}_i$ costs

$$\eta_i = \begin{cases} 0 & \text{: if dock size is unchanged;} \\ c_\uparrow & \text{: if dock size is increased by 1;} \\ c_\downarrow & \text{: if dock size is decreased by 1.} \end{cases} \tag{15}$$

If a dock needs to be enlarged, we have $\kappa_i \geq \widetilde{\kappa}_i$, and vice versa. Similar to Eq. (13), we define the changes at each station as

$$(\kappa_i)_\uparrow = \max\{\kappa_i - \widetilde{\kappa}_i, 0\}, \quad (\kappa_i)_\downarrow = \max\{\widetilde{\kappa}_i - \kappa_i, 0\}. \tag{16}$$

We design the cost function to capture the change *w.r.t.* each station's location weight assignment in (re)configuration. Similarly, we may set the total cost of dock resizing as

$$C^\dagger \triangleq \sum_{i=1}^{M'} \eta_i = \sum_{i=1}^{M'} \left((\kappa_i)_\uparrow \cdot c_\uparrow + (\kappa_i)_\downarrow \cdot c_\downarrow\right). \tag{17}$$

5.3 Problem Formulation

Station (re)placement problem in CBSNR is formulated as: *given* the crowds' site suggestions and the historical usage, the *objective* is to (re)place stations such that *total joint difference* (in crowdsourced feedbacks and historical usage), as well as the total *cost of station (re)placement* are *jointly* minimized.

To accommodate both grid matching and (re)placement cost, we form the final objective as $\psi(\mathbf{H}, \boldsymbol{\Delta}) + \alpha C^*$, where $\alpha > 0$ is a tunable parameter (we empirically set $\alpha = 0.5$). Formally, we have

$$\begin{aligned} \arg\min_{\mathbf{H}} \quad & \psi(\mathbf{H}, \boldsymbol{\Delta}) + \alpha C^*, \\ \text{s.t.} \quad & \text{Constraints in Eqs. (4), (7), (8) \& (13).} \end{aligned} \tag{18}$$

We further present the formulation of **dock resizing**. Intuitively, more capacity should be assigned to stations with lower $\Delta^i \triangleq \sum_{r=1}^{R} h_r^i \Delta_r^i$ ($i \in \{1, \ldots, M'\}$), *i.e.,* more crowd supports and historical usage. In other words, $\kappa_i \geq \kappa_j$ if $\Delta^i \leq \Delta^j$. In practice, the dock size may not be too large due to space constraint in some city areas. We may pose an upper limit $\kappa_{\max}$ for each dock, and it may vary with local street environment or customization.

Specifically, the dock resizing is to minimize the *dock resizing cost* $C^\dagger$ and match *the frequently-used and popular stations*, *i.e.,*

$$\begin{aligned} \arg\min_{\{\kappa_i\}} \quad & C^\dagger, \\ \text{s.t.} \quad & \kappa_i \geq \kappa_j, \quad \text{if } \Delta^i \leq \Delta^j, \quad \forall i \neq j, \quad 0 \leq \kappa_i \leq \kappa_{\max}, \\ & \Delta^i = \sum\nolimits_{r=1}^{R} h_r^i \Delta_r^i, \quad \sum\nolimits_{i=1}^{M'} \kappa_i + \sum\nolimits_{i=M'+1}^{M} \kappa_i = \mathcal{K}. \end{aligned} \tag{19}$$

Total capacity $\mathcal{K}$ can be slightly larger than actual bike number in order to be more resilient to bike flow dynamics.

5.4 SDP Transformation

Note that $\underline{d}_{ij}^2 \leq \|\boldsymbol{l}_i - \boldsymbol{l}_j\|^2$ in Formulation (18) is a non-convex constraint [7], making its solving rather difficult. To address this difficulty, we introduce a novel semidefinite programming (SDP) technique [7, 13, 22] in order to solve the station (re)placement problem efficiently. Our basic idea is to introduce interim variables representing the station candidate locations, and then relax the lower bound constraints via matrix transformation of SDP [22].

Mathematically, we first define an indicator vector $(\mathbf{o}_{ij})_{M\times 1}$ with M elements, among which the i-th element is 1, the j-th is -1 and all others are 0. Let $d_{ij}^2 = (\boldsymbol{l}_i - \boldsymbol{l}_j)^T(\boldsymbol{l}_i - \boldsymbol{l}_j)$ be the resultant distance (squared) from predictions of $\mathcal{S}_i$ and $\mathcal{S}_j$, and we may further have

$$d_{ij}^2 = \mathbf{o}_{ij}^T \mathbf{L}\mathbf{L}^T \mathbf{o}_{ij}, \quad \forall i \neq j, (i,j) \in \boldsymbol{\mathcal{N}}. \tag{20}$$

We then introduce a transition matrix $\mathbf{Z} \in \mathbb{R}^{M\times M}$ as $\mathbf{Z} = \mathbf{L}\mathbf{L}^T$, or

$$\mathbf{Z} - \mathbf{L}\mathbf{L}^T = \mathbf{0}. \tag{21}$$

Then, we rewrite the aforementioned bound constraint into

$$\underline{d}_{ij}^2 \leq \mathbf{o}_{ij}^T \mathbf{Z} o_{ij} \leq \overline{d}_{ij}^2. \tag{22}$$

Next we relax Eq. (21) into a semidefinite form [7], *i.e.,*

$$\mathbf{Z} - \mathbf{L}\mathbf{L}^T \succeq \mathbf{0}. \tag{23}$$

We aim at transforming Eq. (21) into one with *linear matrix inequality* (LMI) [7, 22] which turns out to be convex and solvable.

Therefore, we introduce a block matrix form called *Schur complement* [7] for transformation, which is formally defined as follows.

Definition 9. *Schur Complement: Let $\mathcal{A}$ be a matrix which is partitioned into four matrix blocks $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$ and $\mathcal{E}$, i.e.,*

$$\mathcal{A} = \begin{bmatrix} \mathcal{B} & \mathcal{C} \\ \mathcal{D} & \mathcal{E} \end{bmatrix}, \tag{24}$$

where $\mathcal{B}$ and $\mathcal{E}$ are symmetric and nonsingular matrices. Then, Schur complement of block $\mathcal{E}$ in matrix $\mathcal{A}$, denoted as $\mathcal{A}/\mathcal{E}$, is given by

$$\mathcal{A}/\mathcal{E} = \mathcal{B} - \mathcal{C}\mathcal{E}^{-1}\mathcal{D}. \tag{25}$$

According to related theory of matrices [7], we have $\mathcal{A} \succeq \mathbf{0}$ if $\mathcal{A}/\mathcal{E} \succeq \mathbf{0}$. Recall that $\mathbf{Z} - \mathbf{L}\mathbf{I}_{2\times2}\mathbf{L}^T = \mathbf{I}_{2\times2}/\mathbf{Z} \succeq \mathbf{0}$ (Eq. (23)), where $\mathbf{I}_{2\times2}$ is a 2×2 diagonal unit matrix. We then have its $(M+2)\times(M+2)$ LMI form:

$$\begin{bmatrix} \mathbf{Z}_{M\times M} & \mathbf{L}_{M\times2} \\ (\mathbf{L}^T)_{2\times M} & \mathbf{I}_{2\times2} \end{bmatrix} \succeq \mathbf{0}. \tag{26}$$

This way, a semidefinite programming solver [7, 22] can be applied upon the LMI, and the non-convex problem can be solved efficiently and effectively. In summary, the final formulation is given by

$$\begin{aligned} \arg\min_{\mathbf{H}} \quad & \psi(\mathbf{H}, \boldsymbol{\Delta}) + \alpha \mathrm{C}^*, \\ \text{s.t.} \quad & \text{Constraints in Eqs. (7), (8), (13), (22), \& (26).} \end{aligned} \tag{27}$$

Then, CBikes rounds each station estimation $\boldsymbol{l}_i$ to its nearest grid. Service providers may customize and enforce extra constraints (some inaccessible area, *e.g.*, $h_r^i = 0$, or region boundary, *e.g.*, $A \cdot lon_i + B \cdot lat_i + C \geq 0$) given geographical areas where a dock is not supposed to be deployed (say, a building or a river).

In practice, SDP relaxation renders Eq. (23) a slightly flexible design instead of an over-rigid one, helping adapt to more sophisticated network structures underneath. Other refinements, if needed, can be applied to fine-tune those relaxed distance bounds. One may also check on over-relaxed pairs and adjust using the gradient descent approach [7] to re-satisfy their constraints. We observed only a very small proportion (say, usually less than 1.85%) out of all station pairs need a cosmetic refinement, making our SDP design applicable in most cases.

5.5 Complexity Analysis

We briefly analyze the computational complexity of CBikes. Given M stations and total N_f feedbacks, finding Δ_r^i's of all R grids takes $O(N_f + MR)$. With M stations and R grids, the complexity of SDP is $O(M^3R^3)$ [7, 22], and the total sums to $O(N_f + M^3R^3)$ for CBikes. Further computation reductions can be made in several ways. For example, for each $\mathcal{S}_i$, out of all grids we may only consider the top several location candidates, which have lower Δ_r^i's, and locally search its potentially-nearby neighbors [6, 20] for fewer mutual distance constraints in the optimization. Using the above methods, R and constraints (say, Eqs. (7), (11) and (22)) can be reduced significantly, thus achieving better computational efficiency.

6 EXPERIMENTAL EVALUATION

We first present the evaluation setups in Sec. 6.1 and then illustrate the experimental results in Sec. 6.2.

6.1 Evaluation Setups & Schemes Compared

We compare CBikes with the following schemes in BSNR design:

- *BSNR-w/o-Cost*: which greedily considers crowds and historical usage, without considering the cost for CBSNR.
- *BSNR-w/o-Crow*: which focuses on only historical usage [8, 20], without crowd feedbacks, to (re)place or resize the BSS stations.
- *BSNR-w/o-Hist*: which greedily considers only crowdsourced feedbacks without historical usage, to (re)configure the stations.
- *BSNR-w/o-Tend*: which considers no inter-station trip tendency, and independently (re)configures each station [18, 32].
- *BSNR-w/o-Dist*: which does not consider any distance bound constraint [17].
- *HEU*: a *heuristic* scheme, instead of joint optimization, adopted by some BSS providers (e.g., Capital Bikeshare [1]). Site candidates are first filtered by some heuristic criteria [1] (like utility). Top-ranked candidates are selected and further fine-grained.
- *RAND*: which *randomly* (re)places the BSS stations into grids and resizes them without using any design metrics in Sec. 4.

We evaluate the above algorithms based on the datasets (*i.e.,* Divvy, Nice Ride and Metro Bike) described in Sec. 4.1. We compare the station networks before and after each CBSNR phase, *i.e.,* $\widetilde{\mathcal{G}}$ and $\mathcal{G}$, including each station's status, *i.e.,* $\widetilde{\mathcal{S}}_i = (\widetilde{lat}_i, \widetilde{lon}_i, \widetilde{\kappa}_i)$ against $\mathcal{S}_i = (lat_i, lon_i, \kappa_i)$. We analyze (re)placement of stations and their capacity change. With the timestamps (t_m in Def. 4), crowdsourced feedbacks before this CBSNR (or between two consecutive expansions, if any) are used as optimization inputs. At each CBSNR phase, we use the following evaluation metrics:

- *Accuracy, precision, f-measure & recall*: We compare the difference with the ground-truth station distribution. Specifically, we determine *accuracy* by checking whether each station is matched with its ground-truth grid. We measure the latter three well-known metrics of binary prediction *w.r.t.* the grids, *i.e.,* a value 1 (0) represents that a station is (not) placed inside a grid.
- *(Re)configuration cost*: we compare the costs of all schemes, *i.e.,* station (re)placement (C^*) and dock resizing ($C^\dagger$). For the purpose of reference, we also show the *ground-truth* (GT) costs derived from the actual (re)configuration done by service providers.
- *Mean absolute error (MAE) & mean squared error (MSE)*: differences between predicted size $\{\widehat{\kappa}_i\}$ and ground-truth $\{\kappa_i\}$.

Unless otherwise stated, the default parameter values are set as follows. For each CBSNR phase, by analyzing trips and stations before it happens, we set the $[\underline{d}_{ij}, \overline{d}_{ij}]$ as described in Sec. 4.4; $\alpha = 0.5$; $\beta = 10$. To balance computation efficiency and (re)placement granularity, we set a 90×90 grid mesh (each grid is 0.23×0.40km^2) for Divvy (Chicago), with a bounding box $[-87.80°W, -87.55°W; 41.74° N, 42.06°N]$. For Nice Ride (Twin Cities), we use a 60×60 grid mesh (each is 0.32×0.26km^2), within a box $[-93.32°W, -93.08°W; 44.89° N, 45.03°N]$. As LA county is much larger, a 120×120 mesh (each is 0.29×0.42km^2) comes with a box $[-118.49°W, -118.12°W; 33.71°N, 34.17°N]$ for Metro Bike. All computation is done on a desktop of Intel Core i7-6700 and 32GB RAM. Based on the existing public market analysis [1], we consider $c_\times = 80$, $c_\circ = 100$ (station (re)placement) and $c_\uparrow = c_\downarrow = 10$ (dock resizing).

Our parameter settings are based only on historical data of periods prior to each CBSNR to make the evaluation bias-free. We have also conducted empirical studies on the selection of other important system parameters, but omit them due to space limit.

6.2 Evaluation Results

Station (re)placement: We first show the (re)placement performance in Figs. 11, 12 and 13. Each bar is provided with the mean

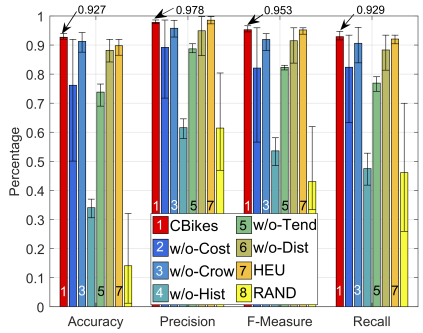

Figure 11: Station (re)placement for Divvy in Chicago (for each metric, left to right: (1)-(8)).

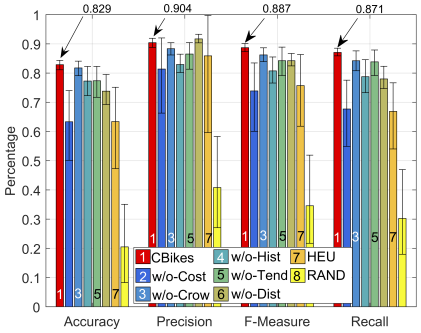

Figure 12: Station (re)placement for Nice Ride in Twin Cities.

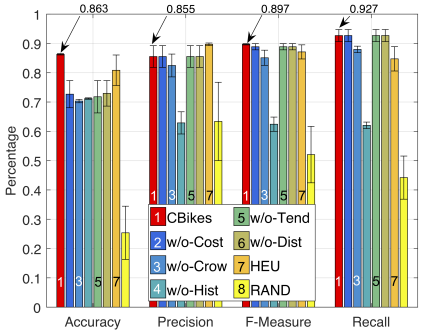

Figure 13: Station (re)placement for Metro Bike in LA County.

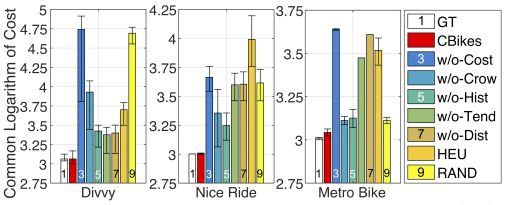

Figure 14: Station (re)placement cost ($\log_{10}(C^*)$).

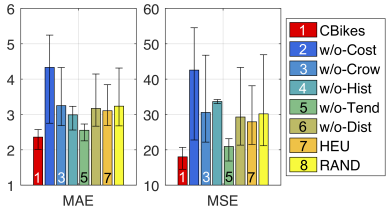

Figure 15: Dock resizing at Divvy.

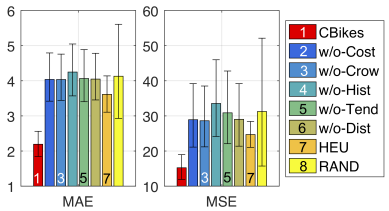

Figure 16: Dock resizing at Nice Ride.

and 75-th/25-th percentiles of all CBSNR phases. We also indicate values with arrows pointing to the means for CBikes. Note that accuracy is based on station index, while others are for binary grid mapping. As wrong matches of stations may still cause similar grid coverage, the accuracy value can in general be stricter and smaller.

Without mutual constraints, *BSNR-w/o-Dist* may get similar grid coverage, but lower matching *w.r.t.* each station. It may hence introduce a much higher moving cost (see Fig. 14). Overall, without support of historical data and joint fusion-based optimization, *BSNR-w/o-Hist* may be easily affected by noisy feedbacks, and achieves much worse and varied performance. Lacking crowdsourced feedbacks, *BSNR-w/o-Crow* cannot determine placement of new stations well, especially for the case of extensive expansion, causing larger variations. *HEU* (heuristic) adjusts stations without joint optimization and global pictures, and thus more post-processing is required before better results can be achieved. In contrast, with joint information fusion and optimization CBikes outperforms others.

Due to a much larger volume of trip data and denser network with more stations, CBikes in Chicago is optimized better and slightly outperforms those in other two cities. Considering the coupling of users and stations (trip tendency and distance bounds) makes CBikes outperform *BSNR-w/o-Tend* and *BSNR-w/o-Dist*. Divvy may witness stronger effect of inter-station trip tendency (more commute and recreational trips) and there is a slightly larger gap between CBikes and *BSNR-w/o-Tend*. Besides, as more CBSNR phases (total 5) are involved in Twin Cities, all schemes experience more performance variations than in other cases.

Fig. 14 summarizes the total (re)placement costs. Clearly, one may expect a huge cost to be incurred by *BSNR-w/o-Cost*. With more information fused, CBikes achieves much lower costs and outperforms others. Besides, its differences with ground-truth (actual (re)placement costs) are also much smaller.

Dock resizing: Due to space limit and similarity of results, we focus on dock resizing of Divvy and Nice Ride here. Figs. 15 and 16 compare the different schemes in terms of resizing MAEs and MSEs *w.r.t.* ground-truth κ_i's. Large resizing error may lead to underutilization or underprovisioning of docks, causing waste and imbalance of BSS resources. CBikes is shown to achieve much lower errors (usually more than 20%) than other schemes. Overall, dock resizing may be easier in Chicago than in Twin Cities due to more trip data and better optimized (re)placement results.

Compared to Divvy, historical usage at Nice Ride is more important in dock resizing than crowd popularity. Due to a sparse network at Nice Ride, most crowds' feedbacks focus on the issues of adapting coverage or density, without paying attention to the resizing of existing stations. Thus, without sufficient historical usage information, *BSNR-w/o-Hist* could not effectively determine the importance of each station's capacity, and hence larger error occurs to it at Nice Ride than *BSNR-w/o-Crowd* and others.

Fig. 17 summarizes the dock resizing costs ($\log_{10}(C^\dagger)$). Note that similar costs may occur when wrong subsets of docks are resized at a similar scale. With better accuracy and lower adjustment cost (often by half an order of magnitude), CBikes helps effectively adapt to bike demands with better feasibility.

Visualization & computation: We visualize (re)configuration prediction and ground-truth results in Fig. 18 for Chicago, Twin Cities and LA County. One can see that the predictions via information fusion and joint optimization markedly resemble the actual values. In terms of computation, the optimization time *w.r.t.* datasets of Divvy, Nice Ride and Metro Bike are 93.71s (due to much more stations), 19.7s and 7.27s, which are suitable for periodic (monthly or annual) network (re)configuration. Parallelization and GPU can be easily applied, which is outside the scope of this paper.

7 DISCUSSIONS

Network Shrinkage: As most existing BSS systems are growing in recent years, our evaluation data in hand mainly contains expansions, and does not include any shrinkage only. However, the data we studied includes removed/moved stations (say, around 21.25% of all stations). Our model is general enough to accommodate both expansion and shrinkage of BSN, and can achieve good accuracy.

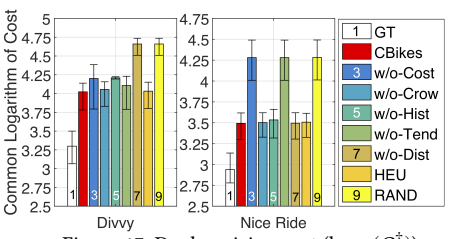

Figure 17: Dock resizing cost ($\log_{10}(C^{\dagger})$).

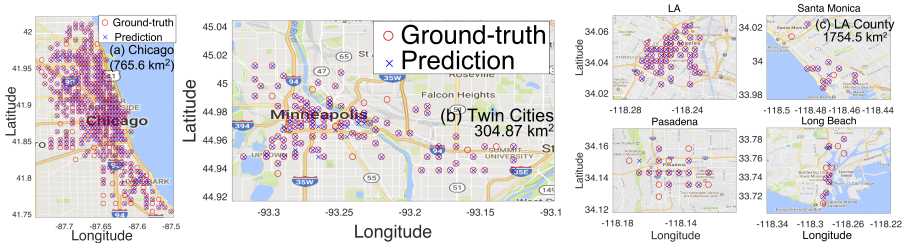

Figure 18: Matching visualization in (a) Chicago, (b) Twin Cities and (c) Los Angeles County.

Incorporating Other Information: Due to resource limit, a myriad of other factors, such as demographic distribution and city management regulation [19, 20, 28], may not be well considered in our current prototype. Their absence might also account for the discrepancy from actual results. However, as a generic framework, `CBikes` can easily integrate them if and when given. Note that we focused on urban-level BSNR, reducing the initial search scope and facilitating decision-making on management of BSNs. Given our results as a reference, secondary fine-grained adjustments of dock locations inside grids may be made subject to various constraints, including bike accessibility, user visibility and space compatibility, which are orthogonal to our focus.

Further Denoising: Large error in using crowds' feedbacks only (*BSNR-w/o-Hist* in Sec. 6) indicates the severity of "noisy" crowdsourcing. `CBikes` can exploit many state-of-the-art approaches [15, 21, 26] to filter the comments or incentivize better suggestions from the crowds. Besides, service providers periodically conduct formal panels or seminars [1] where citizen representatives could discuss BSNR. One may design weighting schemes to assess the quality of various feedbacks for better accuracy.

8 CONCLUSION

BSS network (re)configuration – *i.e.,* station (re)placement and dock resizing – has become very important for many BSS providers. We have proposed a novel optimization framework, `CBikes`, to (re)configure bike station networks with crowdsourced station suggestions. A comprehensive data analysis first derives inter-station trip tendency and distance constraints. Crowds' feedbacks, historical usage, costs and designs are then fused into a joint optimization formulation. We further leverage SDP transformation to solve the nonconvex (re)placement problem efficiently and effectively. Extensive experiments with 3 premium BSS systems, supported by related crowds' feedbacks, have validated the accuracy and effectiveness of `CBikes`.

9 ACKNOWLEDGMENT

This work was supported in part by the DGIST Global Research Laboratory Program through NRF funded by MSIP of Korea.

REFERENCES

[1] City of Falls Church: Bikeshare Ridership Analysis. http://www.fallschurchva.gov/DocumentCenter/View/8694, 2017.
[2] Divvy Suggestion Map. http://suggest.divvybikes.com, 2018.
[3] Nice Ride MN – Station Locations Suggestion Map. http://wikimapping.com/wikimap/Nice-Ride-Suggestions.html, 2018.
[4] Suggest a Location – Metro Bike Share – Los Angeles. https://bikeshare.metro.net/suggest-a-location/, 2018.
[5] J. Bao, T. He, et al. Planning bike lanes based on sharing-bikes' trajectories. In *Proc. ACM KDD*, pages 1377–1386, 2017.
[6] M. d. Berg, O. Cheong, et al. *Computational Geometry: Algorithms and Applications.* Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd edition, 2008.
[7] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge Univ. Press, 2004.
[8] L. Chen, D. Zhang, et al. Bike sharing station placement leveraging heterogeneous urban open data. In *Proc. ACM UbiComp*, pages 571–575, 2015.
[9] M. H. Cheung, F. Hou, and J. Huang. Make a difference: Diversity-driven social mobile crowdsensing. In *Proc. IEEE INFOCOM*, pages 1–9, May 2017.
[10] P. DeMaio. Bike-sharing: History, impacts, models of provision, and future. *Journal of Public Transportation*, 12(4):3, 2009.
[11] Z. Fang, L. Huang, and A. Wierman. Prices and subsidies in the sharing economy. In *Proc. WWW*, pages 53–62, 2017.
[12] Y. Fu, H. Xiong, et al. Exploiting geographic dependencies for real estate appraisal: A mutual perspective of ranking and clustering. In *Proc. ACM KDD*, pages 1047–1056, 2014.
[13] S. He, S.-H. G. Chan, et al. Fusing noisy fingerprints with distance bounds for indoor localization. In *Proc. IEEE INFOCOM*, pages 2506–2514, April 2015.
[14] J. M. Hilbe. *Negative Binomial Regression.* Cambridge Univ. Press, 2011.
[15] H. Jin, L. Su, and K. Nahrstedt. Theseus: Incentivizing truth discovery in mobile crowd sensing systems. In *Proc. ACM MobiHoc*, pages 1:1–1:10, 2017.
[16] M. Karaliopoulos, I. Koutsopoulos, and M. Titsias. First learn then earn: Optimizing mobile crowdsensing campaigns through data-driven user profiling. In *Proc. ACM MobiHoc*, pages 271–280, 2016.
[17] Y. Li, J. Luo, et al. Growing the charging station network for electric vehicles with trajectory data analytics. In *Proc. IEEE ICDE*, pages 1376–1387, April 2015.
[18] J. Liu, Q. Li, et al. Station site optimization in bike sharing systems. In *Proc. IEEE ICDM*, pages 883–888, Nov 2015.
[19] J. Liu, L. Sun, et al. Rebalancing bike sharing systems: A multi-source data smart optimization. In *Proc. ACM KDD*, pages 1005–1014, 2016.
[20] J. Liu, L. Sun, et al. Functional zone based hierarchical demand prediction for bike system expansion. In *Proc. ACM KDD*, pages 957–966, 2017.
[21] S. Liu, Z. Zheng, et al. Context-aware data quality estimation in mobile crowdsensing. In *Proc. IEEE INFOCOM*, pages 1–9, May 2017.
[22] Z. q. Luo, W. k. Ma, et al. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3):20–34, May 2010.
[23] J. Tang, M. Qu, et al. LINE: Large-scale information network embedding. In *Proc. WWW*, pages 1067–1077, 2015.
[24] J. Wang, J. Tang, et al. Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach. In *Proc. IEEE INFOCOM*, pages 1–9, May 2017.
[25] X. Wang, G. Lindsey, et al. Modeling bike share station activity: Effects of nearby businesses and jobs on trips to and from stations. *Jour. Urban Planning & Development*, 142(1):04015001, 2016.
[26] H. Xiao, J. Gao, et al. A truth discovery approach with theoretical guarantee. In *Proc. ACM KDD*, pages 1925–1934, 2016.
[27] M. Xu, T. Wang, et al. Demand driven store site selection via multiple spatial-temporal data. In *Proc. ACM SIGSPATIAL GIS*, 2016.
[28] Z. Yang, J. Hu, et al. Mobility modeling and prediction in bike-sharing systems. In *Proc. ACM MobiSys*, pages 165–178, 2016.
[29] C. Zhang, K. Zhang, et al. Regions, periods, activities: Uncovering urban dynamics via cross-modal representation learning. In *Proc. WWW*, 2017.
[30] F. Zhang, N. J. Yuan, et al. Sensing the pulse of urban refueling behavior: A perspective from taxi mobility. *ACM TIST*, 6(3), Apr. 2015.
[31] J. Zhang, X. Pan, et al. Bicycle-sharing system analysis and trip prediction. In *Proc. IEEE MDM*, volume 1, pages 174–179, June 2016.
[32] J. Zhang, X. Pan, et al. Bicycle-sharing systems expansion: Station re-deployment through crowd planning. In *Proc. ACM SIGSPATIAL GIS*, 2016.
[33] X. Zhang, Z. Yang, et al. Free market of crowdsourcing: Incentive mechanism design for mobile sensing. *IEEE TPDS*, 25(12):3190–3200, 2014.
[34] Y. Zheng. Methodologies for cross-domain data fusion: An overview. *IEEE Trans. Big Data*, 1(1):16–34, March 2015.
[35] Y. Zheng, L. Capra, et al. Urban computing: Concepts, methodologies, and applications. *ACM TIST*, 5(3):38:1–38:55, Sept. 2014.
[36] X. Zhou. Understanding spatiotemporal patterns of biking behavior by analyzing massive bike sharing data in chicago. *PLOS ONE*, 10:1–20, 10 2015.