Improving on-Campus Bicycle Parking Facilities Through Community-assisted Traffic Navigation

Utkarsh Goel, Clint Cooper, MB Parsa, and David Millman

Gianforte School of Computing, Montana State University, Bozeman, MT 59717
{utkarsh.goel, clint.cooper, m.parsagharamaleki, david.millman}@montana.edu

1 Introduction and Overview of the Problem

With the growing levels of greenhouse gases in the atmosphere, many people have shown increased interests to embrace the environmental-friendly mode of transportation, biking [5]. Incidentally, biking offers numerous other benefits, such as a means of daily exercise while commuting to work, which is both economical and saves time. Montana State University (MSU) strives to be one of the leaders to motivate its students, faculty, and staff to prefer biking as their primary mode of transportation. However, in order for MSU to achieve a biker-friendly campus, receptacles to park bikes must be available at all times. As of the current date, such receptacles are either not placed appropriately across various campus locations or are not available in abundance at all times. This situation is evident by the increasing numbers of bikes being tied to vertical posts or even trees at several MSU campus locations.

Other university campuses, such as University of Montana, take several approaches to ensure that the receptacles are available at all times during peak school hours – resulting in University of Montana receiving the Silver Bicycle Friendly Award by League of American Bicyclists [11]. To the best of our knowledge, there is currently no mechanism available to MSU that enables efficient utilization and placement of bike receptacles to strongly promote a biker-friendly campus. As a result, MSU students, faculty, and staff often feel discouraged to use their bikes for commuting purposes.

2 Challenges to Obtain Data for Bike Parking

One of the biggest challenges that the MSU community faces today is the lack of available bike parking receptacles in the campus. While it is possible to throw huge amounts of money at the problem to install more bike racks outside of every university building and satisfy the needs of all the active bikers on campus today, such an approach will also require a continued investment as the university population grows. Therefore, we consider this approach to be unthoughtful of the university’s resources.

Another approach would be to arrange for a human task force to manually acquire information about the number of available receptacles for each bike rack at different times of the day, with the goal of moving the bike racks that remain under-utilized to the locations where bike racks are completely utilized. However, before a class begins, many students park their bikes and occupy many of the available bike parking receptacles. Further, after a class finishes, students take-off their bikes and ride to a different building for attending another class. Such behavior in utilizing bike parking facilities results in a high churn rate where bike receptacles are being occupied before the classes and made available after the classes throughout the whole day. Given the number of buildings where students attend classes, it remains challenging to manually collect data representing such high churn rates at different times of the day. Moreover, if such techniques were to see a wide-scale adoption in MSU, bike racks would need to move from one location to another everyday as the students commute from one class to another. Therefore, we argue that such approach does not offer a scalable placement of bike receptacles on campus, given the dynamically changing bike parking ecosystem everyday.

Finally, the publicly available bike parking dataset for MSU campus provides locations of several departmental buildings along with the maximum number of people allowed inside the building at any given point in time, and the locations of bike racks along with their capacities [6]. In our experience with the dataset, to compute an optimal placement of bike racks, we encountered the following six limitations.

- The dataset does not provide any insights as to how many bike racks or even bike receptacles outside a given building are occupied at different times of the day. Therefore it is not straightforward to
determine whether the number of bike racks installed today are sufficient to address the needs of daily bikers.

- The dataset also does not provide any insights as to what the churn rates are before and after classes, for the bike racks installed outside of different buildings. Therefore, it is difficult to determine whether or not some bike racks are completely utilized at different times of the day.

- While the dataset provides the capacity of a building, it does not give us an accurate view how the racks are utilized. For example, some long range commuters that we informally surveyed reported leaving their bikes at one rack all day and walking between various campus locations.

- The capacity of bike racks also does not provide a strong indication as to how many of the receptacles are available at different times.

- The list of department buildings and bike racks is not exhaustive, in that it provides locations and capacities of only a small portion of the university campus. Therefore, the current dataset ignores data for a substantial number of buildings and bike racks that may be useful when calculating optimal bike rack placement strategy.

In addition to the above mentioned limitations of the available dataset, we argue that while there are several scientific algorithms developed in the past, such as flow simulation [13, 21], load-balancing [20], and Markov Chains [18] which solve similar problems, these algorithms would not be effective when applied over the available data. This is because these algorithms would have to assume qualities that are not present in the data and thus consequently lead to inaccuracies in calculating strategies to place bike racks in the campus. Therefore, given the data we have, an optimal placement of bike racks is impractical to accomplish. Instead, what is needed is a more detailed and exhaustive dataset with information such as, the number of available receptacles at a given bike rack at different times of the day, the churn rate for bike racks outside every building on campus, and accurate geographical coordinates of bike racks. To address the above limitations and compute an optimal strategy to place bike racks in campus, we design an automated system that relies on bikers to contribute the needed data, which we discuss next.

3 Community-assisted Traffic Navigation

In order for MSU to promote a biker-friendly campus, we propose DuckSoup, a low-cost crowd-sourced system inspired by novel solutions to traffic navigation and sharing of available parking resources. DuckSoup is unique in that it allows the MSU biker population to participate in a cooperative ecosystem with the goal of finding an available receptacle for every biker. DuckSoup monitors the availability of bike receptacles at all times and redirects bikers to nearby bike racks with available receptacles. For the MSU biker population to actively participate, we design DuckSoup to offer incentives to its participants through an attractive reward system. DuckSoup is designed to be scalable in that as more and more bikers participate, it provides improved recommendations as to where bikers could park their bikes. Although, the Waze mobile application does not offer a parking service, DuckSoup is inspired from Waze to rely on community participation to make parking recommendations and that DuckSoup also rewards its participants for contributing bike rack availability data [4].

DuckSoup goes beyond current approaches to address bike parking problems and allows a constantly growing biking ecosystem to be monitored by participation from bikers. Specifically, DuckSoup’s novel approach to redirect bikers offers a short-term solution to help bikers find parking spots with minimal upfront costs, and a long-term solution to monitor the utilization of bike racks for optimizing their placements.

3.1 Improving Bike Parking Facilities in the Short Term

Our goal with DuckSoup is to allow a biker participant to notify the system as to where (or which bike rack) the biker parked their bike. One of the critical data points of recording availability of bike receptacles is the location of bike racks itself. While one could utilize Global Positioning System (GPS) on the participant’s mobile device to capture the locations of bike racks when the bike is parked, GPS signals may not be precise enough to accurately point to a bike rack [1]. Additionally, the accuracy of GPS signals rely on its direct line-of-sight communication with the GPS satellites, which makes it difficult for GPS to retrieve locations when obstacles are present in the path, such as tall buildings, a cloudy day, or tall trees [3, 12, 15, 19]. While new mobile devices use signal strength from cellular towers
to utilize triangulation-based techniques to find the location of the device, as the load on cellular towers increases, the device may not be able to find sufficient number of cellular towers to perform an accurate triangulation [17]. Therefore, in such cases, a mobile device location would represent a much larger area than the actual size of a bike rack – resulting in inaccuracies in the locations of bike racks.

We design DuckSoup’s framework to allows labeling of each bike rack using a unique Quick Response (QR) code attached to the bike rack [2]. Our immediate goal here is to avoid usage of biker’s mobile device to capture GPS coordinates and to use QR codes attached on bike racks as an indication of the locations of the racks. We plan to manually collect accurate locations of each bike rack and relate them with the QR codes attached in DuckSoup system.

At the start of the system, a DuckSoup participant finds an available bike receptacle on its own and scans the QR code using the DuckSoup application. The scanning of QR code allows DuckSoup to record the bike rack where the user parked her bike, and as well as the number of bike receptacles left available on that bike rack. As more participants scan QR codes attached to different bike racks, DuckSoup updates its knowledge of how many receptacles are available at which locations and at what times of the day. Especially when every biker scans the QR code, DuckSoup acquires detailed churn rates for available receptacles, before the classes begin and after the classes finish. Specifically, when bikers use a receptacle on a bike rack, they scan the QR code attached to the bike rack – indicating that a receptacle has been occupied by the biker. Similarly, when bikers take their bikes out of the receptacles, the biker would need to scan the same QR code again – indicating that a receptacle has been made available by the biker. We refer to the process of occupying a receptacle as check-in and the process of making a receptacle available as check-out.

Finally, as the DuckSoup realizes wider adoption, it would provide recommendations as to where a biker should park their bike, in case the bike rack nearest to the biker’s destination is estimated to be completely occupied. For example, a biker could request recommendations for an available parking spot based on her destination and the time to reach the destination. DuckSoup would then estimate an available bike receptacle nearest to the biker’s destination and navigate the biker to the bike rack. For interested readers, we make a prototype of DuckSoup accessible via Google Chrome browser at https://www.cs.montana.edu/~utkarsh.goel/DuckSoup/.

3.2 Improving Bike Parking Facilities in the Long Term

When a biker checks-in or checks-out, DuckSoup uploads several data points to its back-end server. The data uploaded consists of the timestamp when the bike was checked-in or checked-out, the biker’s MSU NetID, the ID of the bike rack (obtained from the unique QR code), the location of the bike rack (obtained from the publicly available bike rack data [6]), and whether the bike is checked-in or checked-out. We also leverage the data on bike rack capacities to increase or decrease the number of available receptacles during check-in or check-outs [6]. For example, when a biker checks-in a given bike rack, we reduce the number of receptacles available for that bike rack by one. Similarly, when a biker checks-out from a given bike rack, we increase the number of receptacles available for that bike rack by one.

In situations where bikers desire to obtain an available bike rack nearest to their current location, DuckSoup the utilization of a basic k-nearest neighbor algorithm to compute up to five nearest bike racks with available bike receptacles [14]. Specifically, the K-nearest neighbor algorithm computes the geographical distances between a given bike rack and a subset of other available bike racks on the campus to suggest up to five available bike racks nearest to the given bike rack.

The data contributed by participants allows DuckSoup to keep track of how many bikes are occupied for each bike rack at any given time in a day. Such data enables us to calculate the occupancy of each bike rack for time slices, which one could utilize to compute optimal placement of bike racks. For example, if the collected data indicates that a bike rack is occupied nearly all day, we suggest that more bike racks be installed nearby. Additionally, if a rack is observed to be available nearly all day, that bike rack could be placed at a different location with little to no impact on daily bikers.

One of the predictions we make here is that not every biker on the campus would check-out or be reliable in doing so. Additionally, many bikers may remain unaware of DuckSoup or refrain from installing the application on their mobile device. To mitigate any inaccuracies that result from such bikers, we could leverage two techniques. First, we plan to implement a pseudo-half-life function for each bike rack that periodically reduces the number of bikes at each rack to simulate people checking out – thus allowing us to compensate for participants that neglect to do so. The timing and relative magnitude of this adjustment will need to be empirically tested and tuned over time to accurately represent the real world behavior of the bikers. And second, when a bike rack is observed to be 80% occupied, DuckSoup could consider that
bike rack to be completely occupied by both participants and non-participants. The threshold of 80% occupancy is inspired from how load-balancing is performed on public cloud servers [16].

4 Rewarding Bikers for Participation

The sustainability of DuckSoup relies on active participation from bikers. To encourage early participation and quickly retain large numbers of bikers, we design DuckSoup to offer attractive incentives to the participants. For every check-in and check-out recorded for a given biker, we plan to reward the biker with a choice of either accepting getting a punch towards a mug of coffee in the MSU SUB/Library, or a point added to their CatCard balance. Additionally, to promote even more participation, we suggest to reward the first 100 participants with at least 30 check-in and check-outs with a 5 USD MSU Bookstore gift card.

5 Anticipated Deployment Costs

The deployment of DuckSoup requires minimal upfront costs, which we categorize as follows.

- **Printing QR codes**: There is absolutely no cost to generate QR codes. In fact, several online Web services provide techniques to generate QR codes at no cost [9]. There is only a minimal cost to print QR codes and attach them to each bike rack on campus. For example, the cost to print 100 QR codes on a sticker/waterproof sheet is less than 3 USD [10].

- **Server resources**: As DuckSoup records the participant-contributed data on a relational database, performs on-demand complex operations to calculate available bike receptacles, and computes optimal placement of bike racks as the data grows, maintaining DuckSoup on already running servers inside MSU campus will result in no additional deployment costs. For initial development and testing of DuckSoup, we plan to host the data contributed by the biker participants on servers maintained by the Networks Lab at Gianforte School of Computing [8]. Additionally, to perform complex operations on the large amounts of data for redirecting bikers to available bike receptacles in the short term and computing optimal placement of bike racks in the long term, we plan to leverage the Hyalite Cluster hosted by MSU ITC [7].

- **Application Development**: Inspired by the open-source community, we make the code base of DuckSoup publicly available for further development at https://github.com/ugoel/DuckSoup. We encourage MSU students to actively contribute code and evolve DuckSoup’s functionality to offer innovative features. We believe that contribution from students would enable MSU to lower down the cost of developing a full-fledged DuckSoup application.

DuckSoup does not require any upfront investments from bikers. Specifically, all participants need is a mobile phone with access to the Internet and camera. MSU offers free WiFi to every student, faculty, staff, and even outsiders across its whole campus. Further, with the increased popularity of mobile devices and Internet applications, it is reasonable to assume that most every biker has a mobile device capable of accessing the Internet via MSU’s WiFi.

6 Expected Impact

We expect that the impact of employing a biker-driven crowd-sourced mode of data collection on MSU campus would be twofold: First, such a system would be the first one-of-its-kind step that enables MSU to become a more biker-friendly campus, which in turn would encourage more people to choose biking as their preferred mode of transportation. And second, employing DuckSoup campus-wide would allow MSU to gain significant visibility within the League of American Bicyclists and potentially enable MSU to offer a much stronger competition to other universities in the coming years.

References


