Hybrid Rebalancing with Dynamic Hubbing for Free-floating Bike Sharing Using Multi-objective Simulation Optimization

Vahid Mahmoodian[a], Yu Zhang *[b], and Hadi Charkhgard[a]

[a]*Department of Industrial and Management Systems Engineering, University of South Florida, Tampa, FL 33620, USA*
[b]*Department of Civil and Environmental Engineering, University of South Florida, Tampa, FL 33620, USA*

January 21, 2019

Abstract

For rebalancing problem of free-floating bike sharing systems, we propose dynamic hubbing (i.e. dynamically determining geofencing areas) and hybrid rebalancing (combining user-based and operator-based strategies) and solve the problem with a novel multi-objective simulation optimization approach. Given historical usage data and real-time bike GPS location information, dynamic genfenced areas (hubs) are determined to encourage users to return bikes to desired areas towards the end of the day through user incentive program. And then for remaining imbalanced bikes, an operator-based rebalancing operation will be scheduled to take care of that. The proposed strategy determines the number of hubs, their locations, the start time for initiating the user incentive program, and the amount of incentives by considering two conflicting objectives, i.e. level of service and rebalancing cost (weighted incentive credits and operating cost for rebalancing the remaining imbalanced bikes). We implement the proposed method to the Share-A-Bull free-floating bike sharing system at University of South Florida. The results show that incentivising the users to return the bikes to the hub dynamically determining according to the understanding of imbalance of the system can significantly reduce the total rebalancing cost and improve the level of service.

Keywords. Hub-based rebalancing,User incentive program, Free-floating bike sharing, Simulation, Multi-objective optimization

1 Introduction

In recent years, bike sharing has received a lot of attention [1, 2, 3, 4]. This is highlighted by the study conducted by Shaheen et al. [5] that indicates the existence of 150 bike sharing systems across more than 30 countries in the world until 2010. This number is expected to increase to 1608 programs in all over the world by the end of 2018 [6] which indicates a sharp rise in this business. This attention can be partly because of two reasons: first, taking environmental issues into account

*Corresponding author. E-mail address: *yuzhang@usf.edu*

has become even a more important topic in recent years [7], and second, new concepts such as first-mile and last-mile have emerged in multi-modal transportation [8, 9].

It is known that for bike sharing systems, the flow of customers can completely change the tempral and spatial distribution of the bikes and cause an imbalance of demand and supply. Thus rebalancing/redistribution of bikes is critical to ensure the efficiency of bike sharing systems [10]. As described in existing literature (see for instance [11, 12]), rebalancing of bikes can be done either by users with incentive program or by operator with a fleet of rebalancing vehicles. In an operator-based rebalancing method, the operator collects and repositions bikes in order to balance certain number of bikes to predetermined locations. The rebalancing can be static or dynamic or a combination of the static and dynamic [13]. Static rebalacning means that the bikes are rebalanced without the interference of users' activities. Such rebalancing is usually operated during the night when no customers borrow or return bikes. In contrast, dynamic rebalancing is operated periodically in the day when the borrowing and returning of bikes continuously occur.

It is not surprising that most existing studies have focused on conventional bike sharing systems, the so-called docking/station-based bike sharing systems. In these systems, users are only allowed to take a bike from/to a station equipped with special racks and a locking system. However, a new type of bike sharing systems, the dockless/free-floating bike sharing system, has emerged. For example, in North America, the first free-floating bike sharing system introduced by SoBi in 2013 [14]. Such systems do not need docking stations, which cost a large percentage of start-up investment of station-based bike sharing [8]. With the built-in GPS device, the free-floating bike sharing system allows users to leave a bike almost anywhere. The locations of bikes are tracked by the GPS and displayed on smart-phone or web-based APP. The users can use the APP to locate the bikes and even reserve the bikes if the program is designed to allow the users to do so. Given these advantages, free-floating bike sharing has been expanded dramatically around the world. However, the advantages of flexibility also raise operational challenges; the demand patterns in the implemented region typically prevents the system from self-rebalancing. Compared to station-based bike sharing, the rebalancing of a fully free-floating bike sharing system is more difficult to manage. Thus, many bike sharing companies actually are running free-floating bike sharing system in a station-based way or partially station-based way. Even worse, some start-up bike sharing companies do not consider the reblancing at all.

Observing the needs, research community has put effort to develop methods tackling on the rebalancing problem of free-floating bike sharing [15, 16]. For example, Pal and Zhang [8] present a mixed integer linear programming model for solving static complete rebalancing problem in a free-floating bike sharing system. They solve this problem by a hybrid nested large neighborhood search algorithm for both single and multiple vehicles. In another study, Pal et al. [17] extracts the mobility patterns and imbalance of a free-floating bike sharing system by considering the interaction of independent variables in a statistical model on historical trips and weather data.

In light of the above, we further the rebalancing research of free-floating bike sharing system. The main contributions of our research are (1) developing hybrid rebalancing by combining user-based incentive program and operator-based rebalancing to take the advantage of both; (2) introducing dynamic hubbing (geofencing areas) for incentive program design; (3) solving the problem with novel multi-objective simulation optimization method. Specifically, on daily basis, our approach makes three important decisions. It determines the optimal number and location of *hubs*, as well as the starting time for offering an incentive to customers, and finally, the amount of incentive on daily basis. In order to make these three decisions, the approach considers two conflicting objectives including the total rebalancing cost and service level. The cost will be computed by assuming that at the end of the day, an operator has to rebalance the bikes if the incentive program has not already resulted in the perfect redistribution of the bikes between hubs. The service level is defined as the amount of walking required for the customer to reach the bikes.

It is worth mentioning that the idea of identifying hubs in free-floating bike sharing systems is not new in itself. For example, in two very recent studies, Caggiani et al. [18, 19] introduce methods for clustering the region under consideration into some hubs. However, our proposed approach is quite different because it is a multi-objective simulation optimization technique that makes three important highly-related decisions at the same time in which only one of them is identifying the number of hubs.

The rest of this paper is organized as follows. In Section 2, we provide a description of the problem. In Section 3, the proposed approach is explained . In Section 4, the numerical results and analysis on a real-world dataset are presented. Finally, in Section 5, some final remarks are provided.

2 Problem description

As mentioned in the introduction, redistribution of bikes plays a major role in the total cost of a bike sharing system and customers' satisfaction. In general, because uncertainty in free-floating bike sharing systems is higher than station-based bike sharing systems, the complexity of complete rebalancing problem in these systems is higher. Hence, the focus of this study is on the rebalancing problem in free-floating bike sharing systems. We attempt to find the best strategy for redistributing the bikes using both user-based and operator-based rebalancing methods. Specifically, it is assumed that at some point towards the late hours of each day, the operator starts to stimulate the users to redistribute the bikes in a free-floating bike sharing system. Also, we assume that once the user-based operator is complete, i.e., at the end of each day, the (static) operator-based rebalancing is performed. In light of these assumptions, this study attempts to answer the following key questions for a given day:

- What time of the day should the user-based rebalancing method be started? The start time is denoted by ST in this study.
- What should be the incentive for the user-based rebalancing method? The incentive rate is denoted by IR in this study and shows the amount of the incentive. It can be interpreted as the specific units of the ride in the system, such as a 2-minutes free ride.
- What are the number and place of the hubs for rebalancing the bikes in both user-based and operator-based rebalancing methods in a daily fashion? We assume that the region under consideration is gridded/partitioned into (equal-size) rectangular areas which are referred to as zones. One goal of this study is to select a subset of these zones identified as "hubs". The hubs will be used as the predetermined locations for rebalancing the bikes in both user-based and operator-based rebalancing methods in this study. The number of hubs is denoted by NH.

The optimal values of ST, IR and NH will be determined using a multi-objective simulation optimization approach for every single day. In another word, we will find specific hubs and an incentive offering strategy considering two objectives including customer satisfaction and cost.

Customer satisfaction is measured by the average amount of walking required for the customers to reach the bikes during the time the incentive program starts and early hours of the following day. We do not consider the walking outside of the incentivized time frame because the incentive program does not affect the walking done before the it starts, i.e. ST. A lower value of this measure means that the bikes have a better distribution, and hence, a better service is provided. The other objective is minimizing the redistribution cost of the system for the same day. These two objectives are often conflicting and so there are trade-offs between them. Therefore, the proposed approach attempts to approximate the set of (the so-called) *Pareto-optimal* solutions of this problem, i.e., a solution in which it is impossible to improve the value of one objective without making the value of the other objective worse. The decision maker(s) can then see the trade-offs and choose the most beneficial solution. In this study, we also propose a typical approach for selecting a desirable solution.

3 The proposed method

The proposed approach is a simulation optimization method which utilizes a simulation to evaluate the objectives associated with the solutions of individuals in a population-based heuristic multi-objective optimization method (explained in Section 3.4). For each solution denoted by (NH, ST, IR), in the simulation process, we first determine the locations of the hubs based on the prediction of the demand for the early hours of the next day and the number of hubs (NH). Then, the system is simulated throughout a day for determining where the bikes will be located at the end of the day. For this purpose, the simulation is done only with regular users until the time for incentive offering, i.e. ST, comes. After that moment, the changes in the simulation outlined by the incentive program, explained in Section 3.2, are applied. The simulation will be run for a few times, i.e., 20, in our computational study, and at the end, the average of both objective values over all runs will be computed. The average values will be considered as the fitness, i.e., objective values, of an individual solution (NH, ST, IR).

3.1 Finding locations of hubs

The location of the hubs is determined based on the prediction of the demand in early hours of the next day. The locations of all NH hubs will be selected with the purpose of minimizing the customers' walking in the early hours of the next day. In other words, finding the locations of the hubs can be viewed as a clustering problem. In this study, we employ K-means [20] to cluster the demands and attempts to minimize the summation of Euclidean distances of the demand points from the center of the clusters. Recall that the number of clusters (k) in K-means method will be set to NH. After applying the K-means algorithm, the zones within which the center of the obtained clusters fall will be selected as hubs. So, each cluster results in selecting one hub. Moreover, we use the percentage of the total demand assigned to a cluster to determine the percentage of the bikes required in a hub in the next day. Specifically, if $\lambda\%$ of all demands are assigned to a cluster, $\lambda\%$ of all bikes should be in the associated hub in the next day. Consequently, the rebalancing methods should ensure that by the end of the day, $\lambda\%$ of all bikes will move into that hub.

3.2 Simulation

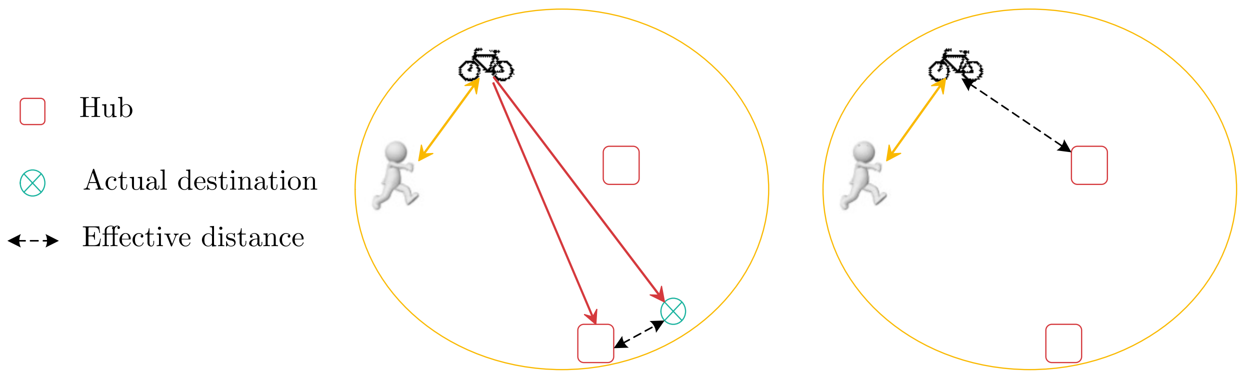
In this simulation, it is assumed that the region is partitioned into equal-size zones in which a customer can clearly see the bike and walk to it without feeling that it is far. For example, in our computational study, the size of each zone is 50×50 meters. We assume that the customers walk to the closest bike when they want to pick one. Moreover, if the closest bike is farther than two zones, a customer will give up and will be counted as a lost customer.

In this simulation, it is assumed that the demand for a ride in each zone follows a Poisson distribution, and its destination is generated based on the distribution of trips which start from this zone. As a result of this, the entrance of the bikes to each zones will follow Poisson distribution as well. Moreover, it is assumed that an occupied bike by a customer will be unavailable for the duration of time that is required to travel from customer's origin to destination. This travel time follows a normal distribution extracted from historical data. This simulation process normally continues until the incentives begin to be offered. The changes in the simulation after this time are further explained in detail.

When the incentive offering starts, the users are informed about our incentive for bikes which are out of hubs and are encouraged to drop them off in one of the predetermined hubs instead of his/her actual destination. At any time, the hubs with fewer bikes than what they planned to have in the clustering step, will be offered as the predetermined hubs in the incentive program. In the simulation, it is assumed that if a user accepts an incentive offer, he/she will drop off the bike within the closest predetermined hub to his/her destination. Overall, we assume that a user accepts an offer with a probability that depends on IR and the distance that they have to walk to their actual destination. Specifically, for a given value of IR, the acceptance probability of an offer decreases as the walking distance of a user to its actual destination increases. This distance is referred to as the *effective distance* and an illustration of that can be found in Figure 1 for two types of customers including regular and *opportunistic* customers. *opportunistic* customers are representative of the users who just like biking and appear only because of the incentives. It is assumed that these customers always pick up the closest out-of-hub bike (for which the incentive is offered) and leave it in one of the suggested hubs. For *opportunistic* customers, the effective distance is different and is defined as the distance that they have to bike to reach the closest hub. For these users, the acceptance probability of an offer is decreasing according to the distance that causes inconvenience to them. Overall, Figure 2 shows the acceptance probability of an incentive offer versus the effective distance that is considered in this study for $IR = 1$ and $IR = 1.5$. The effect of IR is assumed to be only on proportionally stretching or shrinking the curve along the horizontal axis in this study. Apparently, higher values of IR imply higher acceptance probability. This distribution seems completely unknown and different from one person to another. However, it can be easily learned for each specific user based on his/her behavior by running a preliminary incentive program.

In the simulation, when the end of the day reaches, the static operator-based rebalancing will be conducted. In order to do so, the last distribution of bikes and position of hubs will be given to one of the fast and effective static rebalancing algorithms in the literature called NLNS + VND [8]. This algorithm is one of the recent outputs of our research group. Recall that after the static operator-based rebalancing, the number of bikes in each hub will be exactly equal to the numbers planned in the clustering phase.

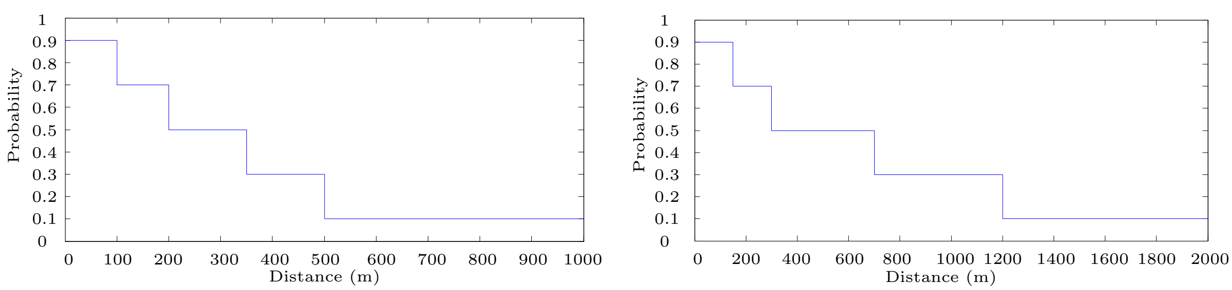
Figure 3 shows a detailed flowchart of the proposed simulation method. The notation used in the flowchart can be found in Table 1. First step of this flowchart addresses the demand prediction



(a) Effective distance for regular customers

(b) Effective distance for oppurtunistic customers

Figure 1: Effective distance based on the customers decide to accept an offer



(a) Acceptance probability with $IR = 1$

(b) Acceptance probability with $IR = 1.5$

Figure 2: Acceptance probability VS. distance

for tomorrow's first several hours and determining hub locations for a given *NH*. Moreover, we already know the current location of the bikes (*Bike.L*), their availabilities (*Bike.IsAct*), ST, and *IR*. Then, the users origin (*Dem.Location*) and destination (*Dem.Dest*) for a unit of simulation time step are randomly generated through function *Generate_customers*. The time step for simulation has been set to 1 minute in this study and users generation follows the distribution of demand associated to each moment of the day. Therefore, time counter, i.e. t, has been passed into function *Generate_customers* as an argument. After that, the number of generated users is computed by function *size* and for each user the closest available bike will be found. The simulation then evaluates if the distance to the closest bike is acceptable by the user, i.e. it is not greater than walking limit *WL*. If the user gives up from picking up the bike, meaning the closest bike is farther than WL then we fail to serve the customer and consequently LC will increase by one. Otherwise, depending on the time and the value of ST the simulation will either include incentive offering's steps or not. In other words, if the incentive offering has not started ($t < ST$) or it has so but the user, based on his/her *effective distance* decides not to accept it, the bike's availability (*Bike.IsAct*) will change to *true* to show that it is not available for a certain units of time. This time is a normally random time that is needed to traverse the distance between *Bike.L* and *Dem.Dest*. It should be mentioned that the user's decision is made randomly based on the value of *IR* and *EDR* in function named *accepted* as we explained before. Also, the location of the bike will be simply updated/set to the actual destination of the user, too. However, if the simulation has entered the incentive offering phase, i.e., $t \geq ST$ and the user accepts the offer, the demand's destination (*Dem.Dest*) will change to the closest hub to that destination, and the number of accepted offers (nAO) will increase by one.

After that all generated users have been processed, if the simulation time has not reached the moment we start offering incentives ($t<ST$), the timer will increase by one unit. Accordingly, the status of the bikes which must be dropped off by that time, i.e., $t \geq Bike.DT$, will be updated to available, i.e., *Bike.IsAct* will change to *false*. If $t > ST$ then we will generate some additional demands which represent oppurtunistic users. The number of oppurtunistic users is proportional to the number of regular demands at the moment t. As explained before, these users have their own *effective distance* based on which they may accept or reject the offered incentive. However, all opportunistic users who accept the offer will walk to pick up the closest incentivized bike no matter how far it is and will take it to the closest hub to their destination. Therefore, the bike's location will be updated to the closest hub. Also, bike's other properties, *Bike.IsAct* and *Bike.DT*, will get updated similar to what we did for regular users.

Finally, when the time horizon of the simulation reaches, *Walking* and *Cost* are computed. *Walking* is recorded from hub determination step,plus walkings of regular users happen during incentive offering. Additionally, we get the cost by running NLNS+VND algorithm on static relocation part of the problem plus the cost we incur for incentives. In continuation, this step will be explained in more detail.

3.3 Evaluation

In this study, two objectives are considered including the service level and cost. The service level is defined as the amount of walking required by customers. The service level is denoted by W and has two parts. The first part computes the (average) walking of customers in early hours of the next day and is denoted by W_1. Specifically, in order to calculate W_1, first, the average distance of (predicted) demands which are assigned to a hub from its center is calculated. We consider the

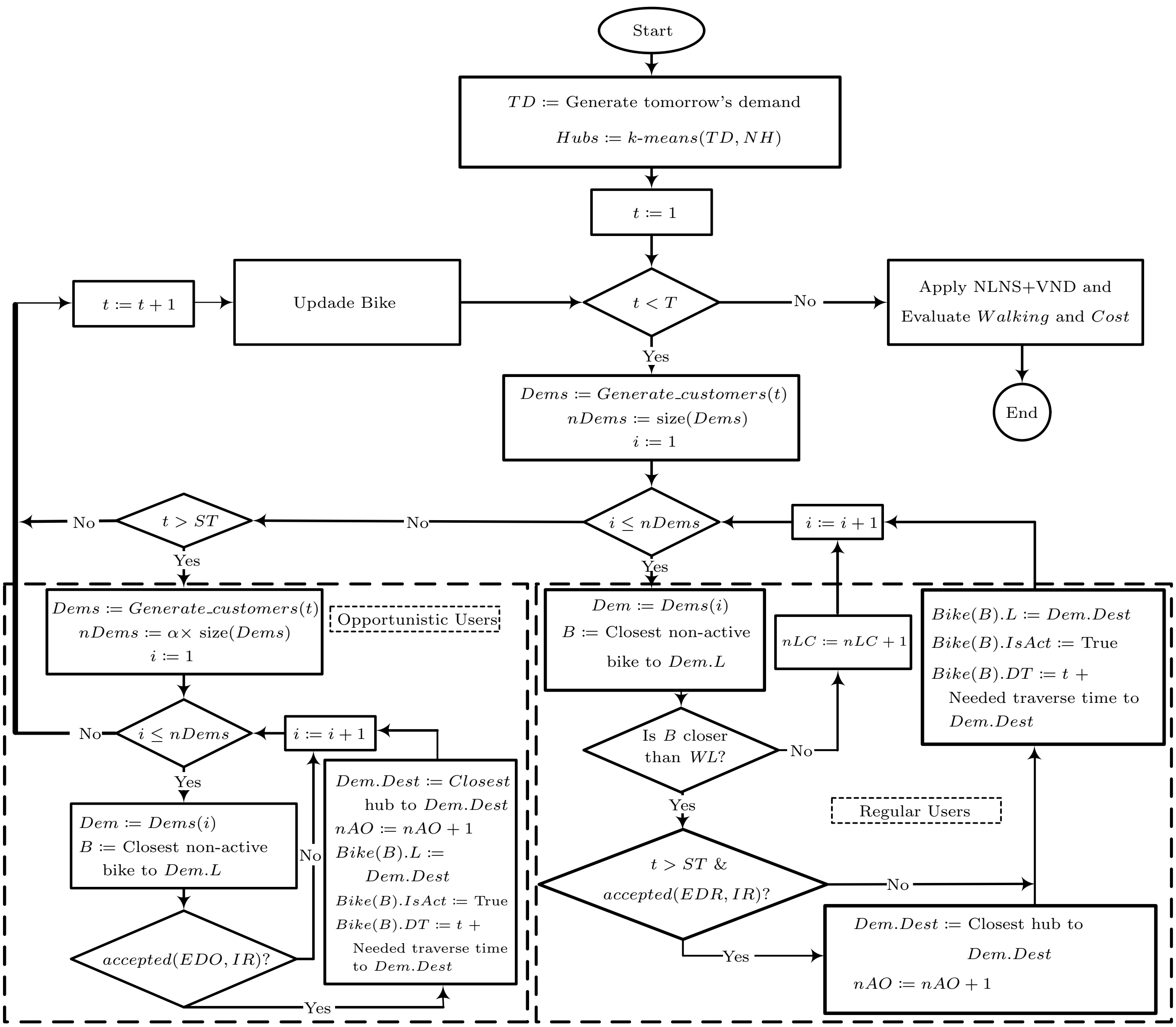


Figure 3: Flowchart of the proposed simulation

Table 1: The Notations used in the flowchart of Figure 3

H	the number of hubs,
ST	start time for incentive offering
IR	the incentive rate
α	the rate of opportunistic customers
T	time horizon of the simulation
WL	the maximum distance that a customer walks to find a bike before giving up
TD	tomorrow's predicted demand
nAO	the number of accepted offers
nLC	the number of customers who have given up
EDR	effective distance for regular customers
EDO	effective distance for opportunistic customers
$Bike.L$	the locations of the bikes
$Bike.IsAct$	the status of the bike which is true if it is under rent and false otherwise
$Bike.DT$	the time by which the rented bike will be dropped off
$Dem.L$	the geographical point in which a demand appears
$Dem.Dest$	the destination of a demand
$Cost$	the objective value for cost
$Walking$	the objective value for walking

average distance to the center in order to cover all possible permutations that these demands may have and affect the number of available bikes in the zone. Another reason is that the exact point of collected bikes in a zone is not certain and we just know that they fall within the borders of the hub zone. Then this average distance is multiplied by the number of available bikes in the hub. We define W_1 as the summation of these values over all hubs. The reason that we consider the (hourly) average is taking all combinations of showing up of predicted demands into account. The second part computes the (average) required walking the users, today and after the incentive offering starts, do through the simulation and is denoted by W_2. Specifically, W_2 is the summation of distances that regular and not-lost users, need to walk to reach the closest bike (after starting the incentive program) divided by the length of the incentive offering time period. The reason for considering hourly walking is that we add this walking to W_1, and therefore, we need to keep the dimension consistent. For example, if regular users walk 1000 meters to find the closest bike and take it (and not give up), and incentive offering starts 4 hours before the end to the day, W_2 will be 250 meters. To compute W_2, we ignore the *opportunistic* customers because the goal of the system is to satisfy regular customers and not the *opportunistic* users.

The second objective, i.e., the cost, includes several parts. The first and the major part is the cost of static operator-based rebalancing, i.e., the rebalancing time obtained from the algorithm NLNS + VND. The second part is the cost of the lost customers, i.e., the total number of lost customers during incentive offering period multiplied by the profit of IR of ride. Finally, the third part is the number of accepted offers multiplied by the cost of one ride.

3.4 Multi-objective optimization

In most real-world problems, there are more than one objective functions that must be optimized. For cases in which the objective functions have the same unit measure, some researchers argue that objectives can be added up to create a single objective function. Evidently such a method cannot be applied to cases in which the unit measure is different and the objectives are conflicting. Hence, many studies have focused on using multi-objective optimization techniques to solve such problems [21, 22, 23, 24]. The main goal in a multi-objective optimization problem is generating some/all *Pareto-optimal* solutions, i.e., solutions in which it is impossible to improve the value of one objective without deteriorating any other objective values. The map of Pareto-optimal solutions in the space of objectives is known as the Pareto-optimal frontier, which helps decision makers understand the trade off between objectives, meaning, they can understand how much they need to sacrifice one objective to improve the other.

There exist several exact methods that can obtain all or a part of Pareto-optimal frontier for some multi-objective optimization problems. However, for many others, computing even one Pareto-optimal solution is not possible in practice. For such cases, researchers attempt to develop heuristic solution approaches. Non-dominated Sorting Genetic Algorithm (NSGA II) is one of the approximate methods which has been widely used in solving multi-objective problems [25, 26, 27]. This algorithm utilizes genetic algorithm to generate a population of feasible solutions. Moreover, It is equipped to a fast subroutine that can produce non-dominated solutions in a given population. In general, this algorithm improves the quality of approximate Pareto-optimal frontier in each generation using the obtained Pareto-optimal frontier in previous generation.

In light of the above, we use NSGA II as the optimizer in our proposed simulation optimization approach. Specifically, the simulation model discussed in Section 3.2, plays the role of fitness evaluator in NSGA II, i.e., it calculates the value of objectives associated with each solution (NH, ST, IR). It should be mentioned that the results of a simulation will only be statistically reliable if their average after enough number of duplications tends to be a fixed value. In this study, we repeat the simulation process 20 times to evaluate the objectives for each solution.

Algorithm 1 summarizes the steps of the proposed simulation optimization approach. In this algorithm, first, the parameters of the algorithm are set. These parameters include the number of population ($nPop$), the number by which the simulation is duplicated to evaluate each solution ($Duplication$), and the termination condition of the algorithm ($Termination\_Condition$). The number of the population shows the number of solutions with the form (NH, ST, IR) we want to keep in the algorithm's pool. The larger this parameter is, the better the algorithm will perform and this in turn requires more valuable computational time. So, as long as the computation time is reasonable this parameter should be set to larger values. Note that the simulation is a stochastic process and the results obtained from one single run is not statistically reliable. Therefore, we should repeat the simulation for an identical solution several times and calculate the average to ensure the robustness of the results. *Duplication* indicates the number of times the simulation is repeated to evaluate a solution in this algorithm. Termination condition, also, determines when the algorithm will finish and can be set based on different measures in the algorithm. The most common termination condition is an upper bound for the number of generations or iterations ($Termination\_Condition$).

In the next step, $nPop$ number of solutions, i.e. each represented by (NH, ST, IR), are generated randomly within a given interval (Pop) and their objective values are evaluated by simulation. This interval for variables can be obtained by the sensitivity analysis or experts estimation. Function *Simulation*, moreover, gets the value of the variables and runs the process as we explained in Sections

3.2 and 3.3. Notice that this step is repeated *Duplication* times for i'th solution and the average values will be considered for objectives' values, *Pop(i).Obj*. Then, the population will be sorted by function *Sort_Frontiers* based on two measures in the context of multi-objective optimization; ranking and distribution in the space of objectives. This sorting algorithm is the main part of NSGA II and interested readers can refer to [26] in order to learn more about it.

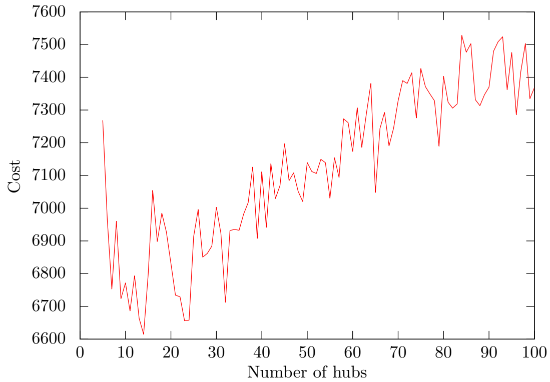
The algorithm's main loop, then, runs until the termination condition is met. The first step of this loop, is the selection of *Parents* which is a set of solutions selected from the population. In this study, the parents are chosen randomly from the population by function *Select_Parents*. Then, some new solutions (which their number is 80% of *nPop* in this study) are generated by applying combination operators on parent solutions. There are several combination operators in the literature of genetic algorithm which can be classified into two groups, crossovers and mutations [28]. In this study we have used continuous crossover operator, and for the mutation, we select a variable randomly and regenerate its value randomly. It should be mentioned that because all variables are integer (in this study), we round the value of the variables obtained from the continuous crossover. Then, these new solutions are evaluated and added to the population. Each iteration, finally, is completed by sorting the population and then removing those solutions which are not eligible to survive for the next generation. This is done by just keeping *nPop* best solutions.

Algorithm 1: Proposed simulation optimization method

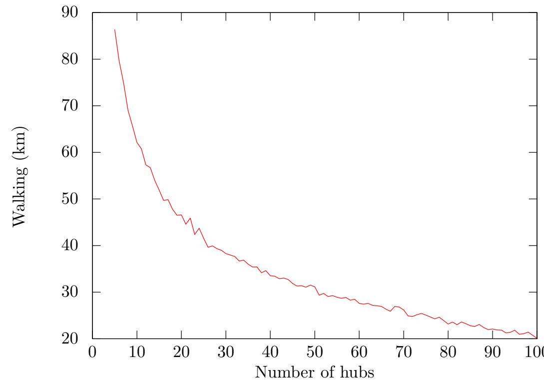
```
Initialize nPop, Duplication and Termination_Condition
for i = 1 : nPop do
    Pop(i).[NH,ST,IR] ← Generate solution [NH,ST,IR] randomly within a logical interval
    Cost ← 0; Walking ← 0; Obj ← [0,0]
    for j = 1 : Duplication do
        [Cost, Walking] ← Simulation(Pop(i).[NH,ST,IR])
        Obj ← Obj + [Cost, Walking]
    Pop(i).Obj ← Obj/Duplication
Pop ← Sort_frontiers(Pop)
while Termination_Condition is not met do
    Parents ← Select_parents(Pop)
    Children ← Combine(Parents)
    for i = 1 :size(Children) do
        Cost ← 0; Walking ← 0; Obj ← [0,0]
        for j = 1 : Duplication do
            [Cost, Walking] ← Simulation(Children(i).[NH,ST,IR])
        Obj ← Obj + [Cost, Walking]
    Children(i).Obj ← Obj/Duplication
    Pop ← [Pop, Children]
    Pop ← Sort_frontiers(Pop)
    Pop ← Pop(1:nPop)
```

4 Numerical results

The proposed method is implemented in Julia and applied on the real dataset obtained from a bike sharing system which is running in University of South Florida (USF). This university serves more than 38000 students with more than 1700 faculties and staff on an about 1500×3000 meters campus. SoBi is providing a free-floating bike sharing system, called Share-A-Bull, in USF since 2015. Now after three years, this system includes 300 bikes and covers all the main campus of USF



(a) Cost v.s. the number of hubs when $IR = 1$ and $ST = 08:00$ $p.m.$



(b) Walking v.s. the number of hubs when $IR = 1$ and $ST = 08:00$ $p.m.$

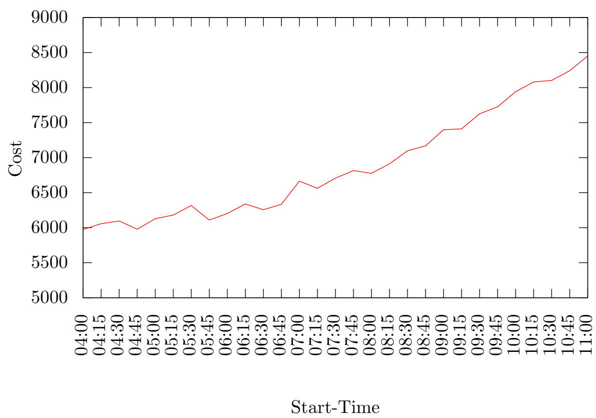
Figure 4: Sensitivity of objectives vs. the number of hubs

(located in Tampa, Florida) and student housing areas in the vicinity. Trip data of this system for one year was analyzed and used to extract the demand distribution in different zones of the area and each hour of the day. It should be mentioned that in this system the demand falls to almost zero after 00:00 a.m. and hence the operator starts doing static rebalancing at that time. Consequently, by this observation, the incentive offering can start at some time in the afternoon/evening. So, we assume that ST can take values between 4:00 and 11:00 p.m.

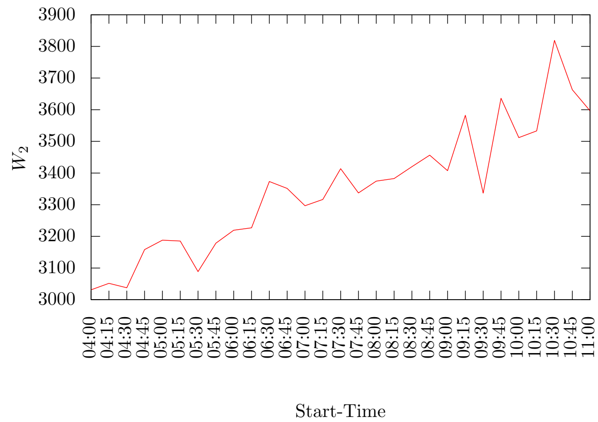
Figure 4 shows that by setting $IR = 1$ and $ST = 8$ p.m. how the number of hubs, i.e., NH, impacts the objective values. Although there are some fluctuations (because of the uncertain nature of the process), the cost first starts to decrease as the number of hubs increases until 12 hubs are available and then it increases. We also observe from Figure 4b that walking, i.e., W, decreases as the number of hubs increases. This is not surprising because by increasing the number of hubs, the dispersion of bikes in the region increases and hence the walking is expected to decrease.

Figure 5 shows that by setting $NH = 12$ and $IR = 1$ how the start time, ST, impacts the objective values. We observe from Figure 5a that the total cost decreases as the start time increases but the slope becomes smoother. One reason for this result can be obtained from Figure 6 in which the average number of lost customers (per hour) is reported for different start times when $NH = 12$ and $IR = 1$. We observe that if we start the incentive offering sooner, the number of lost customers increases. This is because (in that case) the high demand rate in early hours leads to many accepted offers and collecting the bikes in the hubs earlier. Therefore, a decrease in bike dispersion occurs which can increase the number of lost customers. Moreover, as the number of lost customers increases, customers' walking in the remaining hours of today, i.e., W_2, is expected to decrease and this can be observed from Figure 5b.

Finally, Figure 7 shows the trade-offs obtained by the (approximate) Pareto-optimal solutions in the space of objective values. For each point, the presented number indicates the number of hubs. Observe that the lowest cost is obtained by using 8 hubs. Also, as the cost increases, the number of hubs increases. Oppositely, as the walking increases, the number of hubs decreases. So, the main question is now how a desirable solution can be selected. Of course, this is a subjective question and a decision maker should decide about it. However, one typical recommendation is to select a point that has the minimum (Euclidean) distance to the imaginary ideal point, i.e.,



(a) Cost v.s. start time when $NH = 12$ and $IR = 1$



(b) Today's walking v.s. start time when $NH = 12$ and $IR = 1$

Figure 5: Sensitivity of objectives vs. the start time

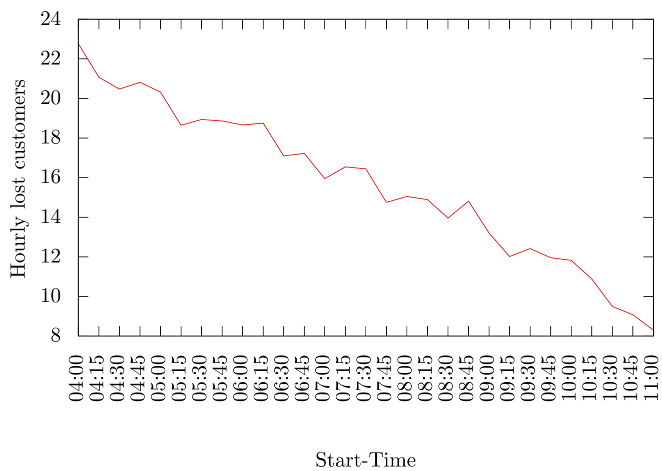


Figure 6: The number of lost customers v.s. start time when $NH = 12$ and $IR = 1$

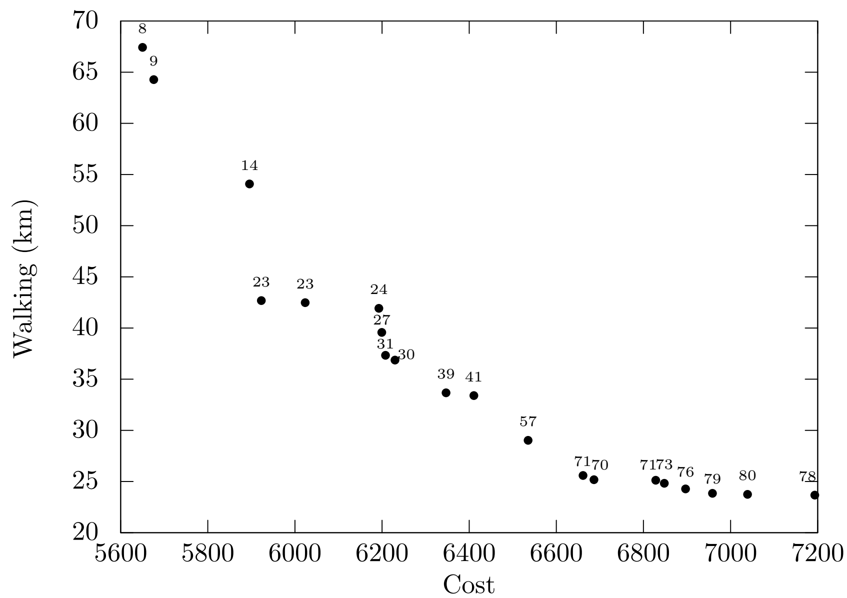


Figure 7: The trade-offs obtained by the (approximate) Pareto-optimal solutions

the point that can be reconstructed by considering the minimum values of both objectives over all Pareto-optimal points. For our problem, the ideal point is $(5650.11, 23671.02)$ and the point that has the minimum distance from it is $(5922.88, 42678.97)$. Therefore, the solution corresponding to this point is $NH = 23$, $ST = 7:00$ p.m., and $IR = 4.37$.

5 Conclusion

In this study a multi-objective simulation optimization method was presented for the rebalancing problem in free-floating bike sharing systems by considering two objectives including cost and service level. The proposed approach combines both the user-based and operator-based rebalancing methods in an effective way. The approach starts by applying a user-based rebalancing method and at the end of the day employs a static operator-based rebalancing method. The underlying idea of the proposed method is to determine an optimal number of hubs and their locations in the region under consideration. The approach then determines at what time an invective program should be started so that the users can possibly redistribute the bikes. Moreover, the approach determines the amount of incentive. We showed the effectiveness of the proposed approach on a real dataset in the University of South Florida. In future research, we consider applying user-based redistribution for rebalancing bikes during the whole day, according to the demand prediction of near future (for one-hour time periods). Additionally, obtaining and using offer-acceptance behaviour of each user, evaluated based on his/her record, is another direction of the future research, rather than one general form for all users. Using other methods for determining the hubs instead of clustering can be another strategy which might improve the performance of the model.

References

[1] Boyacı, B., K. G. Zografos, and N. Geroliminis, An integrated optimization-simulation framework for vehicle and personnel relocations of electric carsharing systems with reservations. *Transportation Research Part B: Methodological*, Vol. 95, 2017, pp. 214–237.

[2] Méndez, L., D. Hernández, G. Villarrubia, and J. F. de Paz, Multi-agent System for the Control and Monitoring of Shared Bicycle Fleets. In *International Symposium on Ambient Intelligence*, Springer, 2017, pp. 187–194.

[3] Caggiani, L., R. Camporeale, M. Marinelli, and M. Ottomanelli, User satisfaction based model for resource allocation in bike-sharing systems. *Transport Policy*, 2018.

[4] Vogel, P., J. F. Ehmke, and D. C. Mattfeld, Cost-Efficient Allocation of Bikes to Stations in Bike Sharing Systems. In *International Conference on Computational Logistics*, Springer, 2017, pp. 498–512.

[5] Shaheen, S., S. Guzman, and H. Zhang, Bikesharing in Europe, the Americas, and Asia: past, present, and future. *Transportation Research Record: Journal of the Transportation Research Board*, , No. 2143, 2010, pp. 159–167.

[6] Richter, F., *Bike-Sharing Clicks Into Higher Gear*, 2018.

[7] Sayarshad, H. R. and J. Y. Chow, A scalable non-myopic dynamic dial-a-ride and pricing problem. *Transportation Research Part B: Methodological*, Vol. 81, 2015, pp. 539–554.

[8] Pal, A. and Y. Zhang, Free-floating bike sharing: solving real-life large-scale static rebalancing problems. *Transportation Research Part C: Emerging Technologies*, Vol. 80, 2017, pp. 92–116.

[9] Chow, J. Y. and H. R. Sayarshad, Symbiotic network design strategies in the presence of coexisting transportation networks. *Transportation Research Part B: Methodological*, Vol. 62, 2014, pp. 13–34.

[10] DeMaio, P., Bike-sharing: History, impacts, models of provision, and future. *Journal of public transportation*, Vol. 12, No. 14, 2009, pp. 41–56.

[11] Caggiani, L. and M. Ottomanelli, A dynamic simulation based model for optimal fleet repositioning in bike-sharing systems. *Procedia-Social and Behavioral Sciences*, Vol. 87, 2013, pp. 203–210.

[12] Sayarshad, H., S. Tavassoli, and F. Zhao, A multi-periodic optimization formulation for bike planning and bike utilization. *Applied Mathematical Modelling*, Vol. 36, No. 10, 2012, pp. 4944–4951.

[13] Raviv, T., M. Tzur, and I. A. Forma, Static repositioning in a bike-sharing system: models and solution approaches. *EURO Journal on Transportation and Logistics*, Vol. 2, No. 3, 2013, pp. 187–229.

[14] Parkes, S. D., G. Marsden, S. A. Shaheen, and A. P. Cohen, Understanding the diffusion of public bikesharing systems: evidence from Europe and North America. *Journal of Transport Geography*, Vol. 31, 2013, pp. 94–103.

[15] Shen, Y., X. Zhang, and J. Zhao, Understanding the usage of dockless bike sharing in Singapore. *International Journal of Sustainable Transportation*, 2018, pp. 1–15.

[16] Kortum, K. and R. B. Machemehl, *Free-floating carsharing systems: innovations in membership prediction, mode share, and vehicle allocation optimization methodologies*. Citeseer, 2012.

[17] Pal, A., Y. Zhang, and C. Kwon, Analysis of Free-floating Bike Sharing and Insights on System Operations or Analyzing Mobility Patterns and Imbalance of Free Floating Bike Sharing Systems, 2018.

[18] Caggiani, L., R. Camporeale, and M. Ottomanelli, A dynamic clustering method for relocation process in free-floating vehicle sharing systems. *Transportation Research Procedia*, Vol. 27, 2017, pp. 278–285.

[19] Caggiani, L., R. Camporeale, M. Ottomanelli, and W. Y. Szeto, A modeling framework for the dynamic management of free-floating bike-sharing systems. *Transportation Research Part C: Emerging Technologies*, Vol. 87, 2018, pp. 159–182.

[20] MacQueen, J. et al., Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Oakland, CA, USA, 1967, Vol. 1, pp. 281–297.

[21] Saffari, H., A. Makui, V. Mahmoodian, and M. S. Pishvaee, Multi-objective robust optimization model for social responsible closed-loop supply chain solved by non-dominated sorting genetic algorithm. *Journal of Industrial and Systems Engineering*, Vol. 8, No. 3, 2015, pp. 42–58.

[22] Mohammadi, R., S. F. Ghomi, and F. Jolai, Prepositioning emergency earthquake response supplies: a new multi-objective particle swarm optimization algorithm. *Applied Mathematical Modelling*, Vol. 40, No. 9-10, 2016, pp. 5183–5199.

[23] Liu, D., K. C. Tan, S. Huang, C. K. Goh, and W. K. Ho, On solving multiobjective bin packing problems using evolutionary particle swarm optimization. *European Journal of Operational Research*, Vol. 190, No. 2, 2008, pp. 357–382.

[24] Marler, R. T. and J. S. Arora, Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, Vol. 26, No. 6, 2004, pp. 369–395.

[25] Srinivas, N. and K. Deb, Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, Vol. 2, No. 3, 1994, pp. 221–248.

[26] Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, Vol. 6, No. 2, 2002, pp. 182–197.

[27] Zhou, A., B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, Vol. 1, No. 1, 2011, pp. 32–49.

[28] Mitchell, M., *An introduction to genetic algorithms*. MIT press, 1998.