

## **A SIMULATION FRAMEWORK FOR THE REBALANCING AND MAINTENANCE OF BICYCLE-SHARING SYSTEMS**

Yu Jin  
Cesar Ruiz  
Haitao Liao  
Harry Pierson

Department of Industrial Engineering  
University of Arkansas  
Bell Engineering Building  
Fayetteville, AR 72701, USA

### **ABSTRACT**

Bicycle-sharing system (BSS) has attracted much attention due to its great success in providing a low-cost and environment-friendly alternative to traditional public transportation systems. In some BSSs comprised of stations with fixed docks, customer satisfaction can be measured by the availability of bikes for pick-ups and/or open docks for returns. However, it is quite common that the spatial balance of bike inventories will be broken due to customers' behavior or frequent failures of bikes and docks. As a result, constant rebalancing and maintenance services are required to sustain adequate levels of customer satisfaction. In this research, a simulation framework is developed to optimize the rebalancing and maintenance activities while satisfying customers' needs over a service area. An optimization model solved by Ant Colony Optimization is applied on the Citibike in New York City, which is considered as an example to validate the effectiveness and efficiency of the proposed simulation framework.

### **1 INTRODUCTION**

Bicycle-sharing system (BSS) has attracted much attention worldwide due to its great success in providing a low-cost and environment-friendly alternative to traditional public transportation systems (Si et al. 2019). Some BSSs are composed of multiple stations, each of which has a fixed number of docks. Customers pick up and return bikes to these stations. Due to customers' behavior, the spatial balance of bike inventory will gradually be broken over time, and consequently, customer satisfaction would reach low levels due to the lack of available bikes or open docks. In practice, a fleet of trucks is often deployed to transfer bikes across stations to meet the demand (i.e., inventory rebalance). This is referred to as the bike repositioning problem (BRP), and it is usually solved either statically (rebalancing at night) or dynamically (rebalancing during the daytime) (Li et al. 2016; Cruz et al. 2017; Schuijbroek et al. 2017).

However, low customer satisfaction levels result not only from the unbalanced bike inventory but also from broken items at the stations. In practice, bicycle and dock failures are intensive due to the high usage rates, and frequent maintenance services are required to sustain the operational condition of the BSS. The usual maintenance activities include sending technicians to repair broken docks at stations and deploying vehicles to transport broken bikes from stations to a repair center. Therefore, the same fleet of trucks can be deployed to conduct both bike rebalancing and collecting broken bikes for maintenance. Nevertheless, of all published studies on BSS, only Kaspi et al. (2017), Alvarez-Valdes et al. (2016), and Wang and Szeto (2018) have considered the impact of broken bikes and related maintenance activities on the performance of a BSS. Kaspi et al. (2017) evaluate the level of user dissatisfaction in the presence of unusable bikes.

The expected number of broken items and the probability that a specific bike is unusable is estimated with a Bayesian model. In Alvarez-Valdes et al. (2016), an optimization model for minimizing the overall cost of unsatisfied demand was used to determine the optimal route of vehicles, and the amount of usable and unusable bikes to be loaded in each station. A simple simulation approach was mentioned to imitate customers' behavior. Wang and Szeto (2018) also consider broken bikes in a BRP by minimizing the  $CO_2$  emission of all vehicles used for rebalancing. In practice, a decision should be evaluated via simulation due to the complexity of a real-world BSS. However, none of the previous studies have provided a simulation framework that incorporates rebalancing and maintenance of a BSS to accurately validate the impact of an optimization model in the daily operation of a BSS.

In this paper, a simulation framework is proposed to evaluate policies on optimizing rebalancing and maintenance for a BSS. The optimization model for rebalancing and maintenance modifies the original model proposed by Schuijbroek et al. (2017). The model is solved by a heuristic algorithm based on the k-medoids clustering and Ant Colony Optimization (ACO). The model is implemented in a simulation system developed for performance evaluation in terms of the repairman utilization, bike availability, and dock availability in a BSS. The contribution of the proposed simulation framework is to offer a valuable tool to the operator of a BSS for making the most efficient decision on rebalancing and maintenance. New York City's BSS (CitiBike, reporting 9000 bike checks per month on average) is considered as an instance to illustrate the proposed simulation framework.

The remainder of this paper is organized as follows. Section 2 presents a brief literature review on BSS simulation. Section 3 describes the proposed simulation framework. Section 4 provides the details on the proposed rebalancing and maintenance strategy. Section 5 provides an application of the proposed framework in CitiBike of New York City. Finally, section 6 draws conclusions and outlines future work.

## **2 LITERATURE REVIEW OF BSS SIMULATION**

Previous studies have developed different simulation frameworks to investigate the effects of rebalancing in BSS. Lin and Liang (2017) simulate the BSS in Taiwan using Arena to determine the optimal number of vehicles for the repositioning that minimizes customers' waiting time. Rebalancing is conducted dynamically with a priority following a first-in-first-serve rule. Saltzman and Bradford (2016) also develop a simulation system in Arena to optimize the configuration of a BSS in San Francisco. However, due to the limitations of Arena, the size of BSS studied in these two papers are comparatively small (e.g., less than 100 stations in the system). Kek et al. (2006) employ discrete event simulation to test different relocation techniques in a car sharing systems in Singapore which has only ten stations. Ji et al. (2014) use Monte Carlo simulation to evaluate the availability of a proposed e-bike sharing system under various scenarios involving different numbers of batteries and e-bikes in the system; however, in their small, two-station pilot system, rebalancing is not considered since it allows riders to make only round trips. Caggiani and Ottomanelli (2013) simulate bike rebalancing in a dynamic case, aiming to minimize vehicle repositioning costs for the BSS operator while maintaining a high level of user satisfaction. The simulation model is a decision support system, which is activated at constant time interval based on the demand predicted by an artificial neural network. A similar decision support system is applied to a free-floating BSS by Caggiani et al. (2018).

One of the important simulation models of BRP is proposed by O'Mahony (2015). The BSS is modeled by a discrete-event simulator implemented in Python. The simulation model assumes that 1) the number of trips taken at a given minute starting at a station is a Poisson random variable, 2) the destination station of each trip follows a multinomial distribution, and 3) the trip duration is also exponentially distributed. This work utilizes the rebalancing optimization strategy proposed by Schuijbroek et al. (2017) and is currently applied in the Citibike operation. In addition, Jian et al. (2016) continue this line of research by using simulation optimization to improve bike and dock allocation. It is worth pointing out that none of the existing simulation models consider maintenance activities indispensable in the BSS. To fill the research gap, a simulation framework that models bike borrowing and returning, bike inventory rebalancing, and maintenance of the BSS is developed in this paper.

### 3 SIMULATION FRAMEWORK

The simulation of BBS is comprised of three major processes: bike borrowing and returning, bike inventory rebalancing, and system maintenance. Bike borrowing and returning consists of generating demand for bikes and docks in each station. Bike inventory rebalancing is made up of all bike transportation activities between stations and between a central location (controller) and the stations. The system maintenance involves activities done by technicians to repair the broken docks in the stations and broken bikes in the controller. Details of each process are provided below.

#### 3.1 Bike Borrowing and Returning

This process follows the same assumptions as O’Mahony (2015). Figure 1 shows the process. It starts with creating bike pick-up demand using a non-homogeneous Poisson process (NHPP) with piece-wise constant arrival rates. The arrival rate at each station varies in a 24-hour cycle, which can be obtained from historical trip data. Then, if the start station has available bikes, a bike will be assigned to the user; otherwise, the user will give up on borrowing a bike and will leave the system. Once a bike is assigned, the destination station is randomly selected from all stations using a multinomial distribution. Selection probabilities can be also obtained from historical trip data. The trip duration is calculated as the distance between the origin and destination stations divided by a fixed bike-speed. Upon arrival at the destination station, the user returns the bike if there is an available dock. If no docks are available, the user goes to the nearest station with available docks. In reality, the user can do this based on the real-time BSS data provided by a smartphone application. After the bike is returned, the age of the bike is updated based on the trip duration, and a Bernoulli trial is conducted to determine whether the bike is still usable with a success probability being the bike’s reliability at the current age. Similarly, dock failures are generated by conducting a Bernoulli trial when the user tries to pick up a bike at the station.

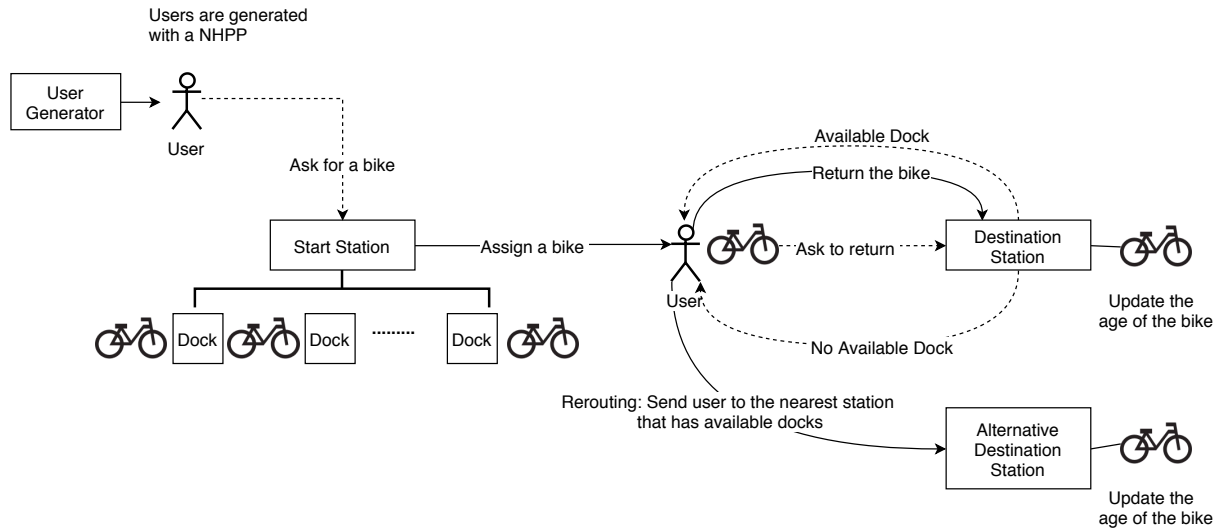


Figure 1: Bike borrowing and returning process.

#### 3.2 Bike Inventory Rebalancing

In the proposed simulation framework, vehicles perform relocating operations during the night (static rebalance) on the usable bike inventory between the controller and stations, and collect broken bikes from the stations for repair at the controller. As shown in Figure 2, the controller gathers the information of bike inventory levels and the number of broken items at each station, and saves such information in a “.txt” file. Then, the optimization module reads the file and solves the rebalancing problem based on the service level requirements and the current state of the BSS. The best solution consists of 1) a route that

determines the stations and the order in which they will be visited by each vehicle, 2) the amount of usable bikes to be picked up/dropped off at each station, and 3) the number of broken bikes to be picked up at each station. Details on the implementation of the optimization module are provided in the next section. Then, the controller reads the optimal solution found and assigns the routing tasks to each vehicle in the fleet. Note that the simulation framework is flexible enough to accommodate a variety of optimization techniques as long as they can be called from the simulation software. Lastly, after the vehicle receives a routing task, it will load an initial inventory of usable bikes from the controller if the total inventory of stations in each route cannot fulfill the demand. At the end of the route, these vehicles will take all broken bikes and any remaining usable bikes back to the controller.

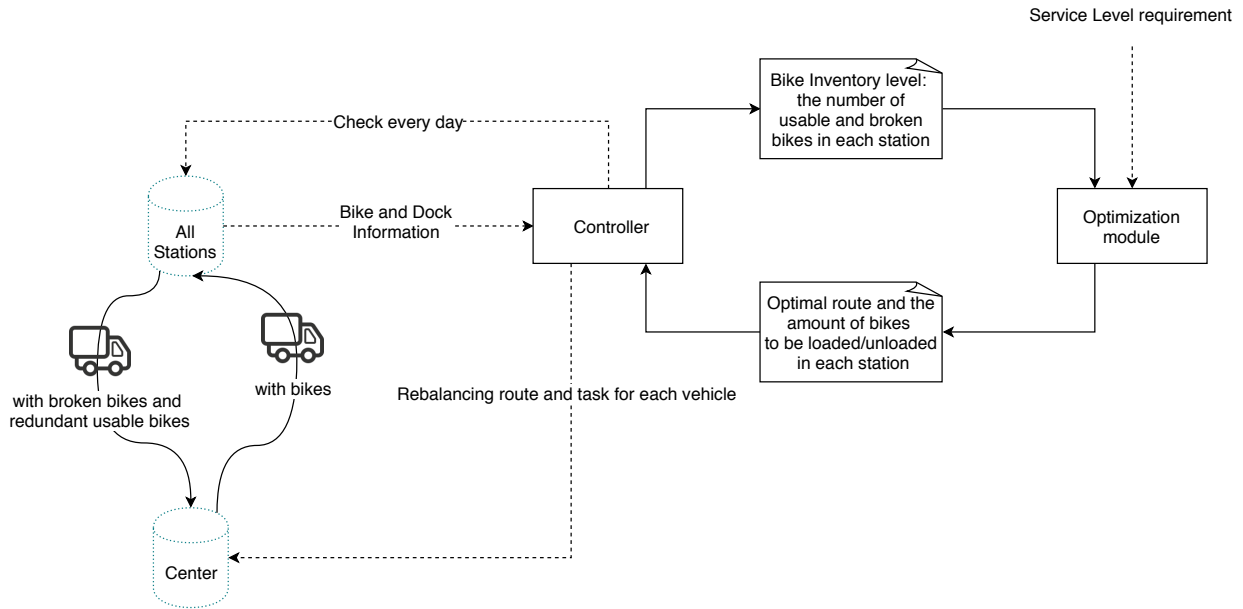


Figure 2: Bike rebalancing process.

### 3.3 System Maintenance

Figure 3 shows the system maintenance process. A group of repairmen repair both bikes and docks. The controller assigns an idle repairman to either repair a bike or go to a station to repair docks with the latter having higher priority. Stations report immediately to the controller whenever a dock fails. Stations with broken docks are served on a first-in-first-serve basis. The travel time of the repairman between stations and the controller is ignored in this work. Bike repairs start as soon as broken bikes arrive to the controller. Each broken bike is assigned to a repairman also following a first-in-first-serve rule. Upon the completion of repair, the age of each bike is set to zero and a new cycle starts. The repaired bikes are used as initial inventory for rebalancing operations. The controller continues to assign available repairmen to tasks unless no items are broken in the system.

## 4 REBALANCING OPTIMIZATION HEURISTIC

The optimization model adopted in the work is a modification of the one proposed in Schuijbroek et al. (2017) for static vehicle-based rebalancing. The makespan of rebalancing is minimized while achieving an inventory level  $s_i$  such that  $s_i^{min} \leq s_i \leq s_i^{max}$  for all station  $i \in S$ , where  $s_i$  is the inventory after rebalancing and  $S$  is the station set. The minimum inventory level  $s_i^{min}$  and maximum inventory level  $s_i^{max}$  are calculated based on the service level requirement by the Kolmogorov forward equation provided in O'Mahony (2015).

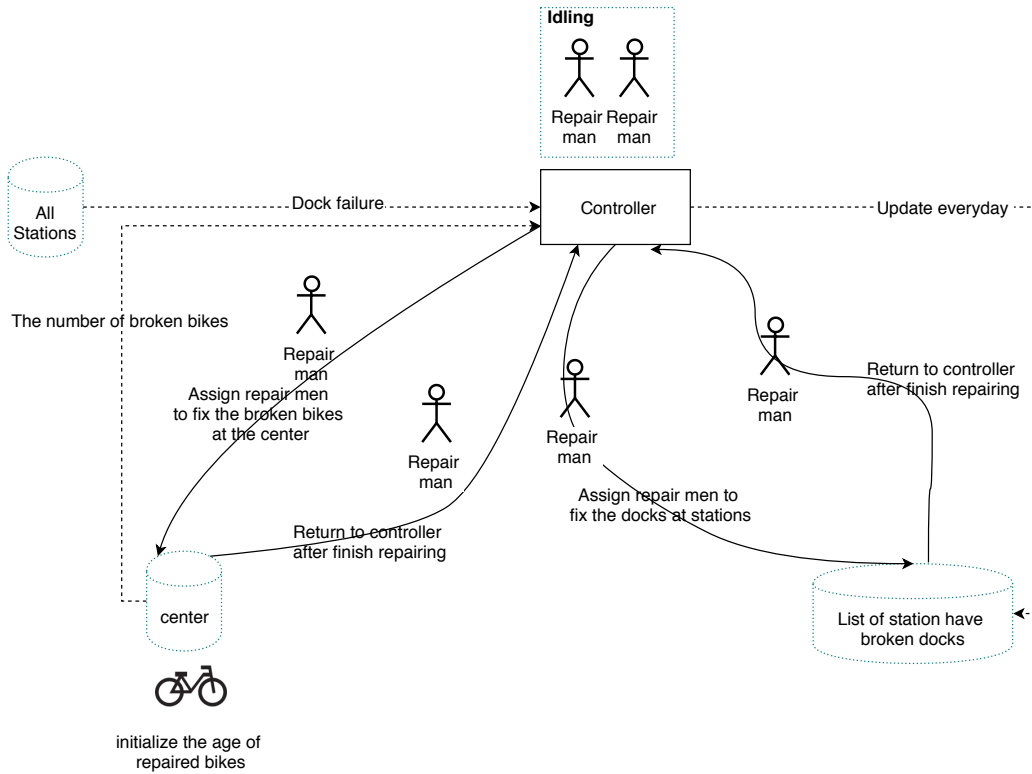


Figure 3: System maintenance process.

In this paper, instead of minimizing the completion time of the rebalancing operations (makespan), the total routing cost, including the travel time and loading/uploading cost, is minimized. The constraints on the service level requirements are incorporated as a penalization term to the objective function. In addition, the influence of broken items on the station and vehicle capacities are also taken into consideration.

It is worth noting that the proposed optimization model and selected heuristic algorithm are provided as an example to show that the simulation framework is able to be integrated to any optimization model for vehicle-based rebalancing and maintenance.

#### 4.1 Formulation for Rebalancing Optimization

In order to reduce the problem complexity, self-sufficient stations are not considered for rebalance operations. Stations are classified as self-sufficient if they have no broken bikes ( $b_i^0 = 0$ ) and  $s_i \in [s_i^{min}, s_i^{max}]$ . The rebalance considering the remaining stations is solved in a two-step process. First, the station set is partitioned into multiple clusters using the k-medoids algorithm where the number of clusters  $k$  is equal to the minimum between the number of vehicles and the number of stations for rebalancing. Second, a route to visit each station within each cluster is determined using ACO (i.e., the route of each vehicle). The objective function of the routing problem is given:

$$\min \sum_{(i,j) \in A_v} d_{ij} + \sum_{i \in R_v} c^-(y_i^- + z_i^-) + \sum_{i \in R_v} c^+(y_i^+) + \sum_{i \in R_v} (\rho_1 \max\{s_i - s_i^{max}, 0\} + \rho_2 \max\{s_i^{min} - s_i, 0\} + \rho_3 \max\{b_i^0 - z_i^-, 0\} - \rho_4 R_i) \quad (1)$$

which considers the total travel time, loading time and unloading time, the penalty of violating the service level requirement, and reward of save inventory. In the formulation,  $R_v$  is an arbitrary route created for cluster  $v$ , which starts at the controller, visits stations assigned to the cluster and comes back to the controller.  $A_v$  consists of two adjacent stations  $(i, j)$  in  $R_v$ . The travel cost of the trip can be found in the travel time matrix  $D$ , where  $d_{ij}$  is the travel time between station  $i$  and  $j$ . The unit bike loading and unloading cost is denoted as  $c^-$  and  $c^+$ , respectively. The number of usable bikes and broken bikes to be picked up at station  $i$  are denoted as  $y_i^-$  and  $z_i^-$ . For the controller,  $z_i^- = 0$ , and  $y_i^-$  is the number bikes for initial inventory. For station  $i$ ,  $z_i^-$  is the minimum of the number of broken bikes at that station and the remaining capacity of the vehicle, and  $y_i^-$  is the minimum of the number of bikes to meet the service level requirement and the remaining capacity of the vehicle. The number of usable bikes to be picked up at station  $i$  is denoted as  $y_i^+$  and it is equal to the minimum among the number of bikes to meet the service level requirement, the number of usable bikes on the vehicle, and the number of available docks at the station. The objective function is penalized if the service level requirements are not satisfied or there are still broken bikes remaining at those stations. The coefficients of penalty are denoted as  $\rho_1, \rho_2$ , and  $\rho_3$ . Moreover, a reward term  $R_i$  calculated by Equation (2) is included with coefficient  $\rho_4$  to allow stations to keep additional bikes without violating the constraints set by  $s_i^{min}$  and  $s_i^{max}$ .

$$R_i = (1 - \max\{s_i^{min} - s_i, 0\} - \max\{s_i - s_i^{max}, 0\}) * \max\{|s_i - \frac{s_i^{min} + s_i^{max}}{2}|, 0\} \quad (2)$$

#### 4.2 Use of Ant Colony Optimization Method

In each cluster, the vehicle routing problem is similar to the traveling salesman problem, which can be effectively solved using the ACO heuristic. The ACO algorithm might not be the best option to solve this optimization problem, but it provides a benchmark performance of the simulation system by using one of the most common techniques to solve the vehicle-based rebalancing and maintenance problem. Details on ACO can be found in Dorigo, Birattari, and Stutzle (2006), and the specifications used in this work are as follows:

- The pheromone on arc  $(i, j)$  in iteration  $t$  is denoted as  $\tau_{ij}(t)$ .
- Exploitation with transition probability: For arc  $(i, j)$  at iteration  $t$ ,  $P_{ij}(t) = \frac{(\tau_{ij}(t)^\alpha (1/d_{ij})^\beta)}{\sum \tau_{ij}(t)^\alpha (1/d_{ij})^\beta}$ , where  $\alpha$  and  $\beta$  are the pheromone exponential rate and heuristic exponential rate, respectively.
- Exploration: 30% of possibility to generate the next node randomly.
- The population size of ant is determined by  $N_v^{ant} = \lfloor |S_v| * p \rfloor$ , where  $p$  is the percentage of the station sequence length.
- Pheromone update:
  - Global update: only the best solution/ant will add pheromone to the path:  $\tau_{ij}(t+1) = \frac{(1-\rho)\tau_{ij}(t)+Q}{(z^*(t))}$  where  $z^*(t)$  is the current optimal,  $\rho$  is the evaporation rate, and  $Q$  is the contribution rate which is a constant  $Q=2$  in this paper.
  - Local update: every ant will add pheromone to the path:  $\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\tau_0$ , where  $\tau_0$  is the initial pheromone level.

### 5 CASE STUDY ON CITIBIKE

The proposed simulation framework is implemented in Java based on the JSL (Rossetti 2008). This implementation includes the classes *Bike*, *Dock*, *Station*, *UserGenerator*, *RepairMan*, *Controller*, and *Vehicle*, for simulating the bike borrowing and returning, bike inventory rebalancing, and system maintenance processes. The optimization module of vehicle routing for bike rebalancing is implemented in Matlab.

The BSS of New York City (CitiBike) is used to illustrate the proposed simulation framework. CitiBike consists of 628 stations with 12,625 bikes in total. The spatial distribution of the stations is shown in Figure

4. We consider only the stations located in Manhattan and Brooklyn (i.e., the station located at Sandy Hook island is excluded). Station location, capacity, and demand rates are learned from the trip data from July 2016 to June 2017 provided by the Citibike data website. The reliability of bikes is inferred from the trip data based on the time between failures. First, a bike is assumed to be failed if it does not start from the destination station of its last trip and has not been used for at least two days considering the high demand in New York City. The time when the last trip ended is considered as the failure time and the total time between failures is the total trip time between two failures. Figure 5 shows the bike reliability based on the historical trip data. In the simulation, the average 24-hour demand rate learned from the trip data is used for generating the demand of each station. The service level requirement is defined as the ratio of satisfied demand over the total demand for bikes and docks. At the end of the first day in the simulation, the bike inventory level of each station is demonstrated in Figure 6. In this work, we assume that 50 trucks are available for bike rebalancing, and each vehicle can transport up to 120 bikes at a time. Moreover, the dock's reliability is assumed to be 0.999 (i.e., one failure for every 1,000 times of usage).

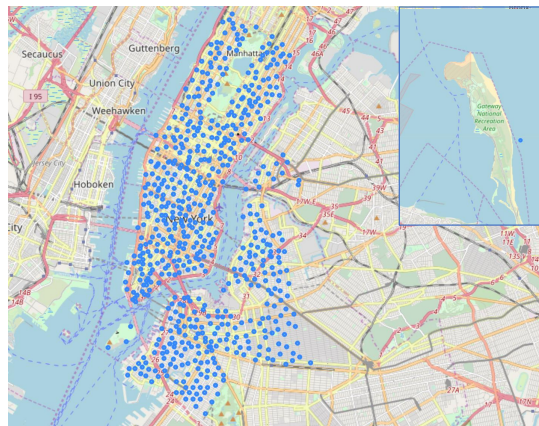


Figure 4: Stations map of Citibike in New York City.

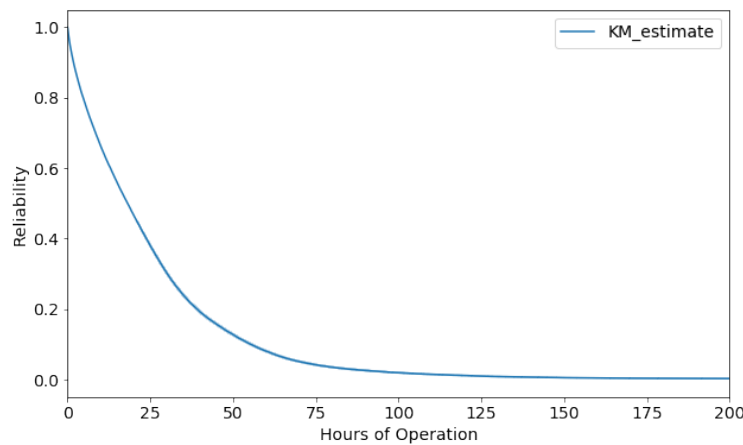


Figure 5: Kaplan-Meier estimate of bike reliability.

### 5.1 Rebalancing Optimization

The stations that require rebalancing are classified into 50 clusters using the k-medoids clustering algorithm. In each cluster, the routing cost can be calculated by Equation (1). The travel cost can be obtained from

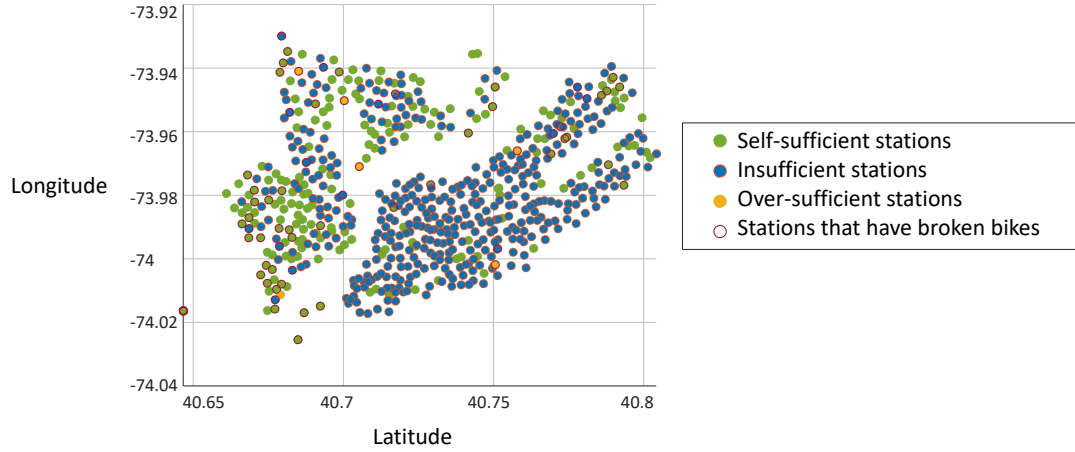


Figure 6: Station inventory level.

the geodesic distance calculated based on the latitude and longitude of stations. The velocity of vehicles is assumed to be 20 km/hr. The loading/uploading cost  $c^-$  and  $c^+$  are 30 seconds. The punishment parameters  $(\rho_1, \rho_2, \rho_3)$  are 100, 500, and 500 per bike, respectively. The punishments for the insufficiency and broken bikes are higher than the over-sufficiency. The reward coefficient is 10 per bike.

After extensive numerical experiments, we found that the best results are obtained when applying 500 iterations of the ACO with local pheromone update and exploration. The population of ants for each cluster is equal to 80% of the number of stations in the cluster. The initial pheromone level  $\tau_0 = 0.1$ . The pheromone exponential rate  $\alpha = 0.5$ , heuristic exponential rate  $\beta = 0.5$ , and the evaporation rate  $\rho = 0.1$ .

The performance of the k-medoids combined with ACO heuristic algorithm is evaluated in terms of the optimal objective value calculated by Equation (1), inventory sufficiency level  $(I, r_I, O, r_O)$ , and broken bike level  $(B, r_B)$  as shown in Figure 7. For the inventory sufficiency level,  $I$  and  $O$  are the total numbers of insufficient and over-sufficient stations, respectively, and  $r_I$  and  $r_o$  are the corresponding average unsatisfied percentages calculated as follows:

$$r_I = \frac{1}{|S|} \sum_{i \in S} (s_i - s_i^{min}), \quad r_O = \frac{1}{|S|} \sum_{i \in S} (s_i - s_i^{max}). \quad (3)$$

It is observed that all the over-sufficiencies are resolved and all broken bikes in the stations are transported to the repair center. However, there are still 35 stations that have not been served. It might be due to the fact that in the route of some vehicles, the broken bikes occupy too many spaces on the vehicle and the vehicle does not have spare for rebalancing usable bikes. This issue can be improved by applying more advanced clustering method to rebalance the workload of transporting broken bikes.

## 5.2 Simulation Result

We evaluate the effects of the rebalancing strategy on the performance measures of CitiBike when 50 vehicles and 15 repairmen are available. In particular, we change the bike availability  $(\beta^-)$  and dock availability  $(\beta^+)$  which are service level requirements defined by Schuijbroek et al. (2017). Note that the formulation in Schuijbroek et al. (2017) does not consider bike or dock failures such that the system performance will not be able to achieve those nominal values. For each configuration, we simulate 30 replications of 10 continuous days of operation and use a warm-up period of 5 days.

Table 1 presents the experimental results. The first column separates the results by the reliability function used into (1) nominal, using the reliability function as obtained in the previous subsection, and (2) improved, reducing the age of the bike by a factor of 100. Under the nominal reliability, bike availability is



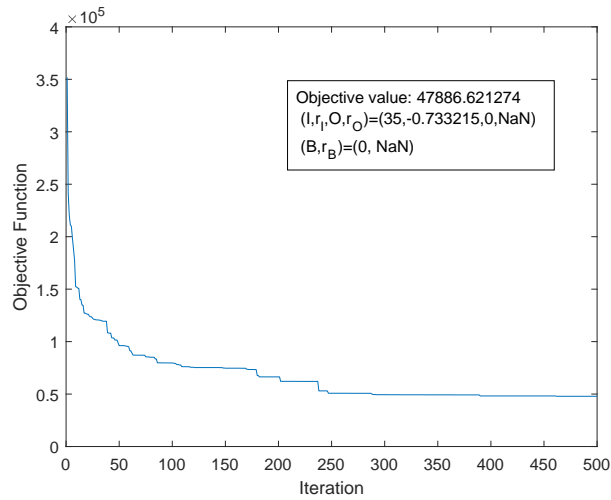


Figure 7: Rebalancing optimization.

at best 52% while the dock availability is always over 98%. This is due to a high number of bikes failing every day such that there are not enough bikes in the system to satisfy the bike demand while users almost always find available docks. In addition, decreasing  $\beta^-$  to be less than 0.95 causes the system to run out of bikes since the minimum required inventory  $s_i^{min} = 0$  for almost all stations. Under the improved reliability function, having low nominal availability requirements produces the best balance on the performance of both bike and dock availability, both over 91%. In contrast, having high requirements on both results in a significant reduction on the bike availability to be less than 85%. Moreover, bike availability is more sensitive to the nominal requirements dropping as low as 52% when  $\beta^+ = 0.95$  and  $\beta^- = 0.75$ . Lastly, the repairmen utilization suggests that using 15 repairmen is enough to serve the current system.

Table 1: Average (standard deviation) of each performance metric for 50 trucks and 15 repairmen in 24 hours.

Bike reliability	$(\beta^-, \beta^+)$	Bike Availability	Dock Availability	Repairmen Utilization
Nominal	(0.75, 0.75)	0.0074 (0.0005)	0.9994 (0.0008)	0.0374 (0.0027)
	(0.95, 0.95)	0.4321 (0.0038)	0.9957 (0.0003)	0.7366 (0.0068)
	(0.95, 0.75)	0.5162 (0.0048)	0.9836 (0.0009)	0.8567 (0.0090)
	(0.75, 0.95)	0.0032 (0.0004)	0.9991 (0.0013)	0.0159 (0.0013)
Improved	(0.75, 0.75)	0.9289 (0.0023)	0.9179 (0.0083)	0.1033 (0.0023)
	(0.95, 0.95)	0.8431 (0.0037)	0.9896 (0.0005)	0.1157 (0.0033)
	(0.95, 0.75)	0.9441 (0.0018)	0.8957 (0.0095)	0.1001 (0.0023)
	(0.75, 0.95)	0.5289 (0.0103)	0.9963 (0.0003)	0.1464 (0.0049)

## 6 CONCLUSION

In this research, a simulation framework is developed to optimize the vehicle-based rebalancing and maintenance activities in BSS while satisfying customers' needs over a service area. An optimization model and a heuristic approach based on the ACO algorithm is applied as an example to show how to solve the rebalancing optimization problem using the proposed simulation framework. A case study on

New York City's BSS (Citibike) is presented to illustrate the effectiveness and efficiency of the proposed simulation framework to help decision-makers evaluate different policies. The simulation results show the importance of considering items failures while making decisions on BSS operations such as rebalancing policies. The simulation framework developed in this paper is not limited to a certain model, but it can be accommodated to any optimization formulations that consider the rebalancing and maintenance of BSS. In our future research, a more advanced rebalancing strategy will be studied and incorporated into the simulation framework to improve the bike and dock availability and labor utilization in a BSS. In addition, more advanced heuristic algorithms will be implemented to solve the vehicle-based rebalancing and maintenance optimization problem. Moreover, a simulation optimization algorithm will be applied to obtain the optimal system configuration.

## REFERENCES

- Alvarez-Valdes, R., J. M. Belenguer, E. Benavent, J. D. Bermudez, F. Muñoz, E. Vercher, and F. Verdejo. 2016. "Optimizing the Level of Service Quality of a Bike-Sharing System". *Omega* 62:163–175.
- Caggiani, L., R. Camporeale, M. Ottomanelli, and W. Y. Szeto. 2018. "A Modeling Framework for the Dynamic Management of Free-Floating Bike-Sharing Systems". *Transportation Research Part C: Emerging Technologies* 87:159–182.
- Caggiani, L., and M. Ottomanelli. 2013. "A Dynamic Simulation Based Model for Optimal Fleet Repositioning in Bike-Sharing Systems". *Procedia-Social and Behavioral Sciences* 87:203–210.
- Cruz, F., A. Subramanian, B. P. Bruck, and M. Iori. 2017. "A Heuristic Algorithm for a Single Vehicle Static Bike Sharing Rebalancing Problem". *Computers & Operations Research* 79:19–33.
- Dorigo, M., M. Birattari, and T. Stutzle. 2006. "Ant Colony Optimization". *IEEE Computational Intelligence Magazine* 1(4):28–39.
- Ji, S., C. R. Cherry, L. D. Han, and D. A. Jordan. 2014. "Electric Bike Sharing: Simulation of User Demand and System Availability". *Journal of Cleaner Production* 85:250–257.
- Jian, N., D. Freund, H. M. Wiberg, and S. G. Henderson. 2016. "Simulation Optimization for a Large-Scale Bike-Sharing System". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 602–613. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Kaspi, M., T. Raviv, and M. Tzur. 2017. "Bike-sharing Systems: User Dissatisfaction in the Presence of Unusable Bicycles". *IIEE Transactions* 49(2):144–158.
- Kek, A., R. Cheu, and M. Chor. 2006. "Relocation Simulation Model for Multiple-Station Shared-Use Vehicle Systems". *Transportation Research Record: Journal of the Transportation Research Board* 1986(1):81–88.
- Li, Y., W. Szeto, J. Long, and C. Shui. 2016. "A Multiple Type Bike Repositioning Problem". *Transportation Research Part B: Methodological* 90:263–278.
- Lin, Y., and F. Liang. 2017. "Simulation for Balancing Bike-Sharing Systems". *International Journal of Modeling and Optimization* 7(1):24–27.
- O'Mahony, E. 2015. *Smarter Tools for (Citi) Bike Sharing*. Ph.D. thesis, Department of Computer Science, Cornell University, Ithaca, New York. <https://hdl.handle.net/1813/40922>.
- Rossetti, M. D. 2008. "Java Simulation Library (JSL): An Open-Source Object-Oriented Library for Discrete-Event Simulation in Java". *International Journal of Simulation and Process Modelling* 4(1):69–87.
- Saltzman, R. M., and R. M. Bradford. 2016. "Simulating a More Efficient Bike Sharing System". *Journal of Supply Chain and Operations Management* 14(2):36–47.
- Schuijbroek, J., R. C. Hampshire, and W.-J. Van Hoesve. 2017. "Inventory Rebalancing and Vehicle Routing in Bike Sharing Systems". *European Journal of Operational Research* 257(3):992–1004.
- Si, H., J. Shi, G. Wu, J. Chen, and X. Zhao. 2019. "Mapping the Bike Sharing Research Published from 2010 to 2018: A Scientometric Review". *Journal of Cleaner Production* 213:415–427.
- Wang, Y., and W. Szeto. 2018. "Static Green Repositioning in Bike Sharing Systems with Broken Bikes". *Transportation Research Part D: Transport and Environment* 65:438–457.

## AUTHOR BIOGRAPHIES

**YU (CHELSEA) JIN** received her B.S. degree in 2014 from Department of Information Science and Technology at Jinan University, Guangzhou, China. In 2016, she received her M.S. degree in manufacturing engineering at the University of Michigan, Ann Arbor. Currently, she is pursuing a Ph.D. degree at the University of Arkansas, Fayetteville. Her research focuses on sensing and analytics, smart manufacturing, as well as data mining and machine learning for manufacturing and service applications. She received the IISE Gilbreth Memorial Fellowship in 2018 and Kuroda Graduate Fellowship in Engineering

*Jin, Ruiz, Liao, and Pierson*

from the University of Arkansas in 2019. She is a student member of IISE and INFORMS. Her email address is [yujin@uark.edu](mailto:yujin@uark.edu).

**CESAR RUIZ** is a Ph.D. candidate in the Industrial Engineering Department at the University of Arkansas. He received his Bachelors degree in Business Engineering from Superior School of Business and Economics, El Salvador. His research interests are in reliability modeling, Bayesian statistics, selective maintenance, and spare-parts inventory control. His reasearch has been published in IISE Transactions and Quality Reliability Engineering International. He was a finalist in the QCRE best student paper competition during the IISE annual conferences in 2017 and 2018. He is a student member of IISE, SRE, and INFORMS. His email address is [caruizto@uark.edu](mailto:caruizto@uark.edu).

**HAITAO LIAO** is a Professor, and John and Mary Lib White Endowed Systems Integration Chair in the Department of Industrial Engineering at University of Arkansas - Fayetteville. He received a Ph.D. in Industrial and Systems Engineering from Rutgers University in 2004. His research has been sponsored by the U.S. National Science Foundation, Department of Energy, Nuclear Regulatory Commission, Oak Ridge National Laboratory, and industry. The research findings of his group have been published in IISE Transactions, European Journal of Operational Research, Naval Research Logistics, IEEE Transactions on Reliability, IEEE Transactions on Cybernetics, The Engineering Economist, Reliability Engineering & System Safety, etc. In 2014, he served as Chair of INFORMS Quality, Statistics and Reliability (QSR) Section, and President of IISE Quality Control and Reliability Engineering (QCRE) Division. He served as Associate Editor for Journal of Quality Technology and IEEE Transactions on Reliability, and currently serves as Associate Editor for IISE Transactions on Quality and Reliability Engineering. He received the U.S. National Science Foundation CAREER Award in 2010, the IISE QCRE William A.J. Golomski Award for three times, 2013 QCRE Track Best Paper Award, Stan Ofsthun Best Paper Awards in 2015 and 2019, and the prestigious 2017 Alan O. Plait Award for Tutorial Excellence. His email address is [liao@uark.edu](mailto:liao@uark.edu).

**HARRY A. PIERSON** is an Assistant Professor in the Department of Industrial Engineering at the University of Arkansas and directs the AT&T Manufacturing Automation Laboratory. He received his Ph.D. from The Ohio State University in 2012 and has over 20 years of combined academic and industrial experience. He also holds an M.S. in engineering management and a B.S. in mechanical engineering from Missouri S&T. He studies collaborative robotics, additive manufacturing, and advanced manufacturing, and his research has been supported by the Air Force Research Laboratory's Materials Directorate and the Army Material Command. His email address is [hapierso@uark.edu](mailto:hapierso@uark.edu)