American Journal of Operational Research 2019, 9(1): 12-17
DOI: 10.5923/j.ajor.20190901.03

A Markov Simulation Approach to Balancing Bike-Sharing Systems

Patrick R. McMullen

Wake Forest University, Winston-Salem, NC 27106, USA

Abstract This research presents a simulation-based approach to establishing stations for an urban bike-sharing system. Movements of bicycles from one station to another are simulated via a Markov process. During the simulation, some stations will lose bicycles, while other stations will gain bicycles. This simulation process is tantamount to the transient phase of a Markov process. Sparsely-used stations are closed, resulting in increased system utilization. The simulation is applied to several test problems. Experimentation shows 86.53% average system utilization, with a standard deviation of 6.81%. It was also discovered that more bikes in the system requires more simulation time, along with more sensitivity to minimum utilization thresholds.

Keywords Simulation, Markov, Optimization, Utilization

1. Introduction

Bike-sharing in urban areas has gained popularity in recent years [1]. Northern Europe was the first to pursue the practice [2,3], while China later pursued bike-sharing on a large scale [2,3]. Portland, Oregon was the first American city to pursue a bike-sharing program [4]. The reasons for bike-sharing are obvious and manifold: traffic is reduced, requiring less parking facilities; the carbon footprint is reduced, resulting in better environmental conditions [5]. Bikes take up less space as compared to motor vehicles, and bikes are nimble, making stops within the urban area quite manageable. Many approaches have been attempted to design and analyze bike-sharing systems. Some approaches are holistic, while some are station-based [6]. Some approaches treat the probabilities of bikes transferring from one station to another as constant, while other approaches treat these probabilities as variable [7,8]. Some of the dynamic transition probability approaches involve forecasting with the intent of accurate modeling.

Regardless of what approach is taken, a bike-sharing system needs to be well-designed, so that the demand for the services is met. If demand is not met, clients of the service are frustrated. If demand is met, but with subsequent excess supply, the management of the bike-sharing service experiences inefficiencies and unneeded costs. Bike-sharing services can be supported by municipal taxes, supported by user-subscription cost, or some combination of both.

It is important to have a bike-system in place that provides adequate stations (or “docks”) for all bikes in the system. In layman’s terminology, there needs to be adequate space to park the bikes, and a station, or “dock” provides an area to park the bikes. At the same time, the management of the bike-sharing system do not wish to have excess stations, as the subsequent inefficiencies result in increased costs. As such, a proper balance is desired – adequate parking spaces for the bikes while avoiding expensive excess parking space.

This research effort provides a holistic methodology for designing such a bike-sharing system. Bike traffic – that is, bikes moving from station to station, is simulated in accordance with a Markov process. During the simulation, stations are reduced by eliminating unused stations. This process continues until the overall system utilization meets some sort of threshold level. At that point, stations are no longer eliminated. Instead, the simulation is in a steady-state, and stops at the appropriate point.

More specifically, the simulation emulates the transient, or build-up, phase, of a Markov process. During this simulation, unneeded stations are eliminated, and bikes needing space are redirected to other stations with the global intent of seeking improved system utilization. In this context, utilization is the ratio of bikes in the system to the overall system capacity. After this transient phase is concluded, system statistics are reset, and the simulation continues without eliminating stations (considered a steady-state system), so that the system dynamics can be studied.

The authors consider the unique feature of this research the exploitation of the simulation’s transient phase in order to obtain a pre-specified service level.

* Corresponding author:
mcmullpr@wfu.edu (Patrick R. McMullen)

Published online at http://journal.sapub.org/ajor

2. Methodology

The following steps detail how the simulation process works. First, the following terms are defined:

Term	Description
n_0	number of stations (“docks”) in initial solution
n	number of stations (“docks”)
b_i	number bikes in station i
T	time units elapsed
T_S	time units elapsed until steady state is reached
T_F	time units elapsed until simulation completion
u_i	utilization of station i
C	capacity of each bike station
$a_{i,j}$	probability of a bike moving from station i to station j
S	successful bike transitions
F	failed bike transitions
U_μ	average system utilization
U_{min}	minimum station utilization
V_μ	average system utilization threshold
V_{min}	minimum station utilization threshold
M	state of process

The items above are largely self-explanatory, but additional details follow. The capacity of each bike station, C, is the number of bikes that a station can accommodate. All of the terms associated with utilization reflect the ratio of the bikes in the station (or system) to the number of bikes that can be accommodated by the station (or system).

Step 1: Initialization

First, the time elapsed, T, is initialized to one. Successful bike transitions and failed bike transitions are initialized to values of zero ($S = F = 0$). The number of initial stations (n_0) is initialized to some value and the number of stations (n) is also set equal to this value. Values for the number of bikes in each station (b_i) are initialized to some value, as are the capacities of each bike station (C). Whichever values are selected for these two entities are such that the utilization of all stations are at 50%. Mathematically, that is:

$$u_i = 0.50, \forall i \qquad (1)$$

Next, the Markov transition matrix is initialized. The transition matrix, $\boldsymbol{A}$, is shown below:

$$\boldsymbol{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix}$$

The transition matrix is engineered such that each row displays the probability of a bike moving from station i to station j, and all probabilities sum to one [9]. Mathematically, this is as follows:

$$\sum_{j=1}^{n} a_{i,j} = 1, \forall i \qquad (2)$$

The state of the process, M, is set to zero, which indicates a transient state.

Step 2: Transition of Bikes

Monto-Carlo simulation is used to select which station (j) receives a bike from the station that loses a bike (i). In the context of this research effort, there are three possible transition outcomes.

Step 2A: Bike Remains in its Original Station

The first scenario is when a bike will not move from station i to station j, because the Monte-Carlo simulation selects station i to be the recipient of the bike. This situation is mathematically the following:

$$i = j \qquad (3)$$

This possible scenario should make sense because as the simulation moves along, some stations will retain their bikes as time passes. In fact, the transition matrix values were randomly generated such that higher selection probabilities were assigned for the $a_{i,i}$ values. It should be noted, however, that for bike-sharing systems of a specific nature, great care should be taken to generate reliable transition matrices. That is, it is critical that the transition matrix reflect actual system dynamics [10].

Step 2B: Failed Transition

The second scenario is when station i has no bikes (if $b_i = 0$) to supply station j, or if station j cannot accommodate another bike ($b_j = C$). Either of these conditions constitute a “failed” transition. If this happens, no transition takes place, and the failed transitions counter is incremented as follows:

$$F = F + 1 \qquad (4)$$

Step 2C: Successful Transition

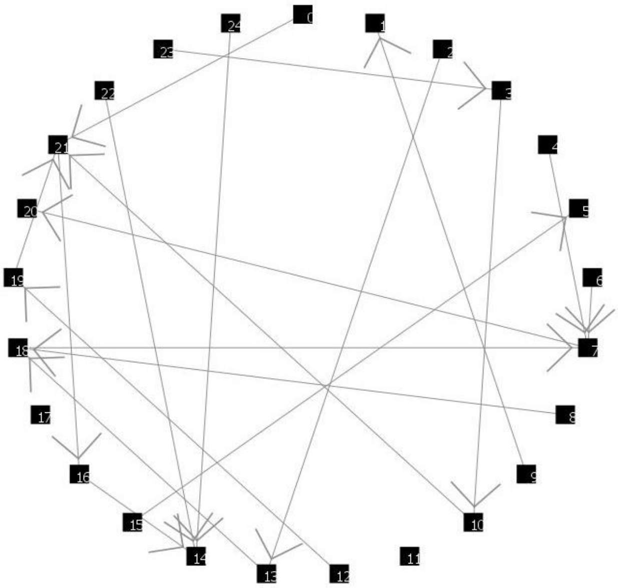

Figure 1. Transition for a Single Time-Step

The last possible scenario is when station i and j are unique ($i \neq j$), station i is not empty ($b_i > 0$), and station j is able accommodate another bike ($b_j < C$). When this occurs, the following updates occur:

$$b_i = b_i - 1 \qquad (5)$$

$$b_j = b_j + 1 \tag{6}$$

$$S = S + 1 \tag{7}$$

For each time increment (T), Monte Carlo selection and transition occurs for all n stations in the system. Figure 1 shows transition for one specific value of time, T.

In this figure, we see several bikes transitioning from one station to another. For example, Station 0 loses a bike to Station 21, while Station 7 loses a bike to Station 20, while gaining bikes from Stations 4, 6, and 18. Stations 11 and 17 neither gain nor lose bikes.

It is important to emphasize that when bikes move from one station to another, the transition times for bikes are considered negligible.

Step 3: Updates

After the above-described transitions are completed, utilization statistics are calculated – specifically, system utilization and minimum utilization. System utilization, the average utilization across all stations is as follows:

$$U_\mu = \left(\frac{1}{nC}\right)\sum_{i=1}^{n} b_i \tag{8}$$

The minimum utilization is the utilization of the station with the fewest bikes. Mathematically, this is as follows:

$$U_{min} = \left(\frac{1}{C}\right)\min\{b_1, b_2, \ldots b_k\} \tag{9}$$

Utilization values for each station update as follows:

$$u_i = \frac{b_i}{C}, \forall i \tag{10}$$

Step 4: Re-Distribution

This step is executed if and only if the system is in a transient state (that is, if $M = 0$). If not (that is, if $M = 1$), we continue to Step 5.

At this point, a check occurs to see if any stations are empty. That is, if $b_i = 0$, station i is added to a list of empty stations. From this list of empty stations, a station s is eliminated via random selection. One station is eliminated in the presence of any empty stations. When station s is eliminated, the following occurs:

$$\text{row } s \text{ is eliminated from } \boldsymbol{A} \tag{11}$$

$$\text{column } s \text{ is eliminated from } \boldsymbol{A} \tag{12}$$

$$n = n - 1 \tag{13}$$

rows of ***A*** re-numbered from 1 (top) to n (bottom) (14)

columns of ***A*** are re-numbered from 1 (left) to n (right) (15)

Step 5: Steady State Check

If the average system utilization exceeds its threshold ($U_\mu > V_\mu$) and the minimum station utilization exceeds its threshold ($U_{min} > V_{min}$), the simulation is considered to be in steady state, and we proceed accordingly. We first change the state variable, M:

$$M = 1 \tag{16}$$

We also set the steady-state time (T_S) stopping time, along with the finishing time (T_F) such that the steady-state simulation run length is equal to the transient-state simulation run length:

$$T_S = T \tag{17}$$

$$T_F = 2T_S \tag{18}$$

We also reset the success count and failure count at steady state as follows:

$$S = 0 \tag{19}$$

$$F = 0 \tag{20}$$

Additionally, U_μ and U_{min} values are recorded from the transient state.

Again, the above steps are executed only if the system is found to be in steady-state. Regardless of the value of the state variable M, the time value is incremented by one:

$$T = T + 1 \tag{21}$$

After incrementing the time value, we return to the transition phase, Step 2.

Step 6: Termination

When the time increment variable, T reaches T_F the simulation terminates, and we record the transition success rate, which is as follows:

$$\frac{S}{S+F} \tag{22}$$

There is no need to record the utilization-related statistics, as they have not changed during the transient state, because the number of stations has not changed during the steady-state portion of the simulation.

To generalize the entire simulation process, the following pseudo-code is provided:

```
Initialize();
While (($U_\mu < V_\mu$) or ($U_{min} < V_{min}$)){
    Transition();
    Update();
    Redistribution()}
RecordTransientStateStatistics();
S = 0;
F = 0;
$T_S$ = T;
$T_F$ = 2T;
For(i = $T_S$; i <= $T_F$; i++)Transition();
RecordSteadyStateStatistics();
```

3. Experimentation

Experimentation is pursued to study the effectiveness of the simulation described above. First, it should be noted that the simulation was performed via NetLogo v. 6.0.4, an agent-based simulation package written in the Java programming language [11]. In the context of an agent-based simulation, the stations or "docks" are treated as the *agents*. These station-agents expand and contract in terms of the number of bikes they accommodate, and some of these station-agents disappear when it becomes apparent that their use is not as important as other station-agents. The number of bikes in the system is initialized at the beginning of the simulation, and the total number of bikes in the system never changes. The number of stations in the system does change (in the transient phase), as a function of the bikes moving from one station to another.

Twelve test problems are studied to evaluate the simulation methodology. Specifically, the number of initial stations (n_0), the number of bikes in each station (b_i), the capacity of each station (C), stopping-threshold values (V_μ and V_{min}) were chosen for each test problem. These user-specified values for each test problem are presented in Table 1.

It should be noted from the table above that for all problems, the minimum threshold for average system utilization is set to 0.75, and the minimum threshold for the station with the lowest utilization if 0.25. What this means is that the transient phase of the simulation will end ONLY if the average system utilization must be at least 75%, and the station with the lowest utilization must be at least 25%. Otherwise, the transient phase of the simulation continues. It should also be noted that the utilization for all stations, and subsequently the entire system is initialized at 50% -- this is established by Equation (1).

4. Results

For each of the (12) test problems, the simulation was performed (25) times. This results in a data set of (300) total observations. Table 2 shows the average (and standard deviation) values of the output values for each of the (12) test problems. The bottom row of Table 2 shows the overall average (and standard deviation) of each output measure.

The average overall system utilization was 86.53%, with a standard deviation of 6.81%. The average minimum station utilization was 41.23%, with a standard deviation of 24.02%. The summary statistics from the other three outputs are reported accordingly.

Table 1. User-Specified Inputs for Test Problems

Problem	Initial Stations (n_o)	Bikes/Station (b_i)	Total Bikes (n_0*b_i)	Station Capacity (C)	Req. Avg. Util. (V_μ)	Min. Stat. Util. (V_{min})
A	12	25	300	50	0.75	0.25
B	12	50	600	100	0.75	0.25
C	12	75	900	150	0.75	0.25
D	12	100	1,200	200	0.75	0.25
E	25	25	625	50	0.75	0.25
F	25	50	1,250	100	0.75	0.25
G	25	75	1,875	150	0.75	0.25
H	25	100	2,500	200	0.75	0.25
I	50	25	1,250	50	0.75	0.25
J	50	50	2,500	100	0.75	0.25
K	50	75	3,750	150	0.75	0.25
L	50	100	5,000	200	0.75	0.25

Table 2. Experimental Results – Means (and Standard Deviations)

Problem	Compression	Avg. Util.	Min. Util	Success	Time
A	0.5600 (0.0382)	0.8971 (0.0655)	0.5448 (0.2982)	0.8563 (0.0605)	288.50 (122.63)
B	0.5633 (0.0363)	0.8914 (0.0623)	0.5164 (0.2854)	0.8708 (0.0702)	595.08 (399.78)
C	0.5400 (0.0425)	0.9314 (0.0729)	0.6736 (0.3499)	0.8473 (0.0551)	927.52 (378.39)
D	0.5467 (0.0422)	0.9120 (0.0724)	0.6238 (0.3462)	0.8608 (0.0501)	1,047.32 (315.77)
E	0.6112 (0.0337)	0.8206 (0.0481)	03208 (0.1138)	0.9233 (0.0262)	427.32 (132.48)
F	0.5984 (0.0408)	0.8394 (0.0591)	0.3556 (0.1425)	0.9228 (0.0224)	1,036.12 (284.53)
G	0.6112 (0.0356)	0.8209 (0.0508)	0.3021 (0.0858)	0.9215 (0.0254)	1,613.36 (624.58)
H	0.5984 (0.0316)	0.8378 (0.0442)	0.3150 (0.0694)	0.9111 (0.0228)	2,517.2 (915.59)
I	0.6256 (0.0329)	0.8015 (0.0442)	0.2880 (0.0476)	0.9623 (0.0118)	454.20 (132.12)
J	0.5864 (0.0350)	0.8556 (0.0502)	0.3204 (0.1070)	0.9558 (0.0162)	1,318.28 (403.37)
K	0.5768 (0.0287)	0.8689 (0.0429)	0.3211 (0.0983)	0.9554 (0.0114)	2,155.16 (636.18)
L	0.5576 (0.0267)	0.8986 (0.0420)	0.3664 (0.1396)	0.9511 (0.0134)	3351.60 (923.59)
Overall	0.5813 (0.0441)	0.8653 (0.0681)	0.4123 (0.2402)	0.9115 (0.0551)	1,311.9 (1,035.4)

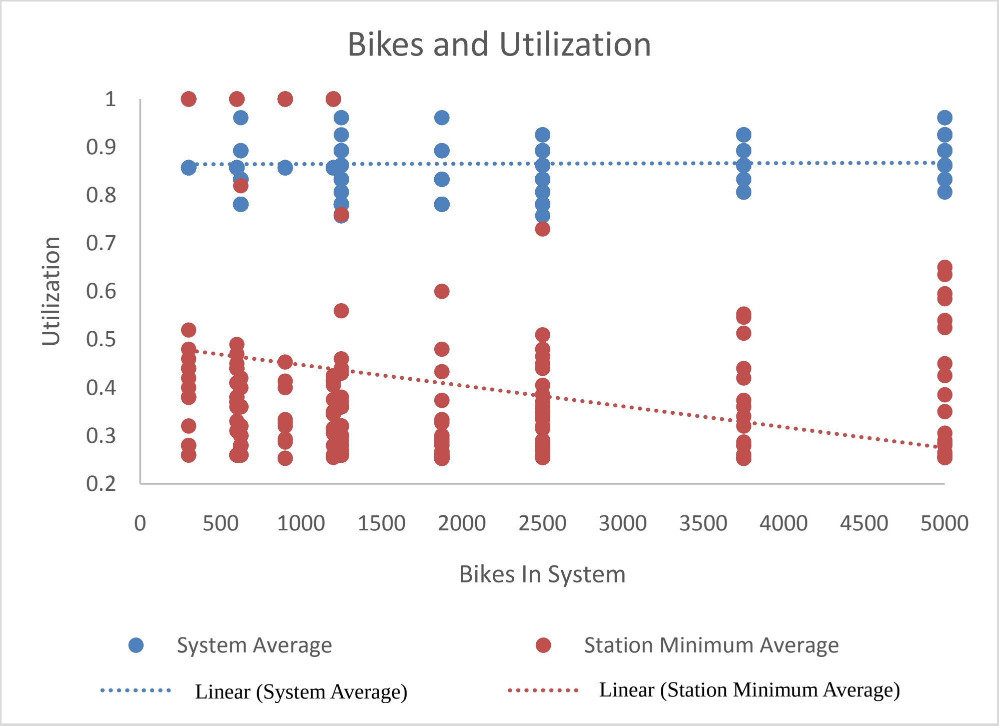

Figure 2. Relationship between Bikes in System and Utilization

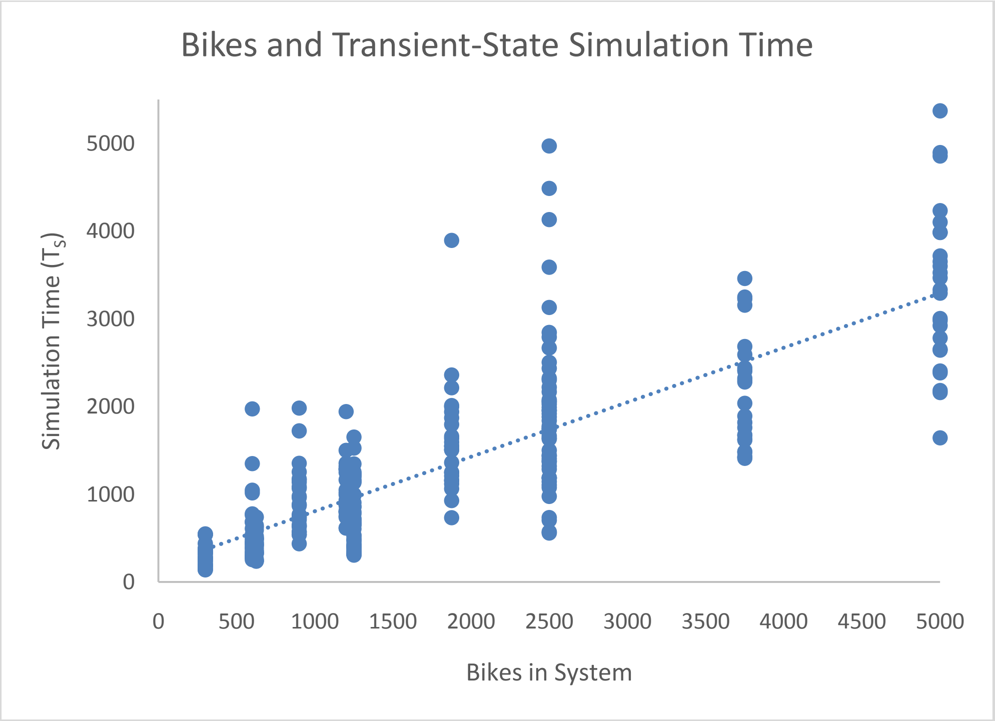

Figure 3. Relationship between Bikes in System and Transient-State Simulation Time

The overall success rate during the transient phase of the simulation is 91.15%, with a standard deviation of 5.51%. The overall success rate during the steady-state simulation phase is 59.74%, with a standard deviation of 4.79%. The success rate decreases during the steady-state phase of the simulation because there are fewer stations to accommodate the movement of bikes. If a higher success rate is desired during the steady-state simulation phase, the utilization thresholds should be lowered and/or the initial system capacity should be increased. This issue is clearly an opportunity for subsequent investigation.

The size of the problem at hand deserves consideration as well. Here, the “size” of the problem is quantified by the number of bikes in the system, and it is important to consider any potential relationships between the number of bikes in the system and important simulation outputs. Figure 2 illustrates the relationship between the number of bikes in the system and utilization – both overall system utilization and station minimum utilization.

Here we see no relationship between overall system utilization and number of bikes in the system (a flat slope), but we do see an inverse relationship between the number of bikes in the system and minimum station utilization – more bikes in the system result in the minimum station utilization being close to the minimum utilization threshold (V_{min}).

Figure 3 shows us that there is a positive relationship between the number of bikes in the system and the time required to complete the transient-state of the simulation.

The regression statistics associated with Figures 2 and 3 are detailed in Table 3.

Table 3. Regression Results

Independent Variable	Dependent Variable	Slope t-statistic	Slope p-value	R^2
Bikes in System	System Utilization	0.23	0.8167	0.0002
Bikes in System	Min. Station Utilization	-4.33	<0.0001	0.0591
Bikes in System	Simulation Time (T_S)	23.97	<0.0001	0.6584

Here, we see that there is no relationship between problem size (bikes in system) and average system utilization, but there is a meaningful relationship between the minimum station utilization and the number of the bikes in the system – a negative relationship. We also see that the relationship between the number of bikes in the system and the transient-state simulation time is meaningful – a positive relationship.

5. Concluding Comments

This effort was concerned with exploiting the Markovian behavior of bikes transition from station to station in an urban environment. The intent was to design a bike-sharing system that fairly balances the tradeoff between supply of bike space and demand for bike space. The general strategy was to start with a system having excess supply, and scaling back supply as the transient-state simulation time continues. The overall system utilization is considered reasonable – a good balance of the system's ability to supply the demand for service.

Of course, with any research effort, there are opportunities for subsequent work, and this effort is no exception. Trying to design a system with a strategy other than one that starts with excess supply is an opportunity – perhaps considering user-satisfaction as an objective function value is an opportunity for study [12]. Trying stopping thresholds other than the ones used here is another opportunity for subsequent research. Adjusting the transition matrix to account for structural changes is something that also should receive consideration [13]. Non-negligible transition times for bikes transitioning from one station to another is also another research opportunity.

REFERENCES

[1] Larsen, J. (2013) Bike Sharing Programs Hit the Streets in Over 500 Cities Worldwide, Earth Policy Institute. http://www.earth-policy.org/plan_b_updates/2013/update112 .

[2] Shaheen, S.A., Guzman, S. and Zhang, H. (2010) Bike Sharing in Europe, the Americas, and Asia: Past, Present, and Future. Transportation Research Record Journal of the Transportation Research Board, 2143, 159-167. https://doi.org/10.3141/2143-20.

[3] Fishman, E., Washington, S. and Haworth, N. (2014) Bike Share: A Synthesis of the Literature. *Urban Transport of China*, 33, 148-165.

[4] Larabee, M. (2009) Portland to Experiment with Rental Bike System. *The Oregonian*.

[5] Zhang, L.H., et al. (2015) Sustainable Bike-Sharing Systems: Characteristics and Commonalities across Cases in Urban China. Journal of Cleaner Production, 97, 124-133. https://doi.org/10.1016/j.jclepro.2014.04.006.

[6] Almanna, M., Elhenawy, M. and Rakha, H. (2019) Identifying Optimum Bike Station Initial Conditions using Markov Chains Modeling. Transport Findings. https://doi.org/10.32866.6801.

[7] Zhao, Y., Wang, L., Zhong, R. and Tan, Y. (2018). Mathematical Problems in Engineering, Article ID 8028714, https://doi.org/10.1155/2018/8028714.

[8] Chiariotti, F., Pielle, C., Zanella, A., and Zorzi, M. (2018) A Dynamic Approach to Rebalancing Bike-Sharing Systems. Sensors. Vol. 18. https://dio.10.3390/s19020512.

[9] Winston, W. (2004) Operations Research: Algorithms and Applications, 4th edition. Cengage Publishing.

[10] Shahsavaripour, S. (2015) A New Approach for Developing Bike Sharing, Network System. *Journal of Civil Engineering Research*, 5, 28-32. https://doi:10.5923/j.jce.20150502.02.

[11] Wilensky, U. and Rand, W. (2015) An Introduction to Agent-Based Modeling. MIT Press.

[12] Chi, Y.H. (2018) Self-Organized Bike Redistribution in Urban City. American Journal of Operations Research, 8, 386-394. https://doi.org/10.4236/ajor.2018.85022.

[13] De Chardon, C.M., Caruso, G. and Thomas, I. (2016) Bike-Share Rebalancing Strategies, Patterns, and Purpose. Journal of Transport Geography, 55, 22-39. https://doi.org/10.1016/j.jtrangeo.2016.07.003.