

Article

Optimization Strategies for Dockless Bike Sharing Systems via two Algorithms of Closed Queuing Networks

Rui-Na Fan ^{1,*} , Fan-Qi Ma ¹  and Quan-Lin Li ²

¹ School of Economics and Management, Yanshan University, Qinhuangdao 066004, China; mafanqi@stumail.ysu.edu.cn

² School of Economics and Management, Beijing University of Technology, Beijing 100124, China; liquanlin@tsinghua.edu.cn

* Correspondence: fanruina@stumail.ysu.edu.cn

Received: 19 February 2020; Accepted: 12 March 2020; Published: 18 March 2020



Abstract: The dockless bike sharing system (DBSS) has been globally adopted as a sustainable transportation system. Due to the robustness and tractability of the closed queuing network (CQN), it is a well-behaved method to model DBSSs. In this paper, we view DBSSs as CQNs and use the mean value analysis (MVA) algorithm to calculate a small size DBSS and the flow equivalent server (FES) algorithm to calculate the larger size DBSS. This is the first time that the FES algorithm is used to study the DBSS, by which the CQN can be divided into different subnetworks. A parking region and its downlink roads are viewed as a subnetwork, so the computation of CQN is reduced greatly. Based on the computation results of the two algorithms, we propose two optimization functions for determining the optimal fleet size and repositioning flow, respectively. At last, we provide numerical experiments to verify the two algorithms and illustrate the optimal fleet size and repositioning flow. This computation framework can also be used to analyze other on-demand transportation networks.

Keywords: sustainable transportation; dockless bike sharing system; mean value analysis; closed queuing network; flow equivalent server algorithm; optimization strategy

1. Introduction

Bike sharing systems (BSSs) are a type of smart transportation. Nowadays, there are primarily two types of BSSs. The first one, the station-based bike sharing system (SBBSS), consists of stations distributed across the city. The capacity of a station is limited. An arriving user rents a bike from a station and returns it to a destination station if there is a vacant dock. Otherwise, the user must find another station to return the bike. SBBSSs have been carried out in over 2900 cities around the world ([1]). Another type of BSS, with no fixed stations for parking bikes, is the dockless bike sharing system (DBSS). Bikes can be parked everywhere allowable without capacity limit (refer to Shi et al. [2]). The DBSSs first arose in 2014 and now they have been deployed in over 250 cities in the world. Since the difference of capacity and parking positions, the modeling and computation of the SBBSS and DBSS are their distinguishing features. In this paper, we provide research for the DBSS.

Since the BSS is large scale and dynamic, it is vital to determine an optimal fleet size. The optimal fleet size is a tradeoff between the earnings from users and maintaining cost of the BSS. Bike availability is defined to be the probability that the parking region is not empty. It is an essential task to understand the relationship between the bike availability of each parking region and the fleet size. If the fleet is in surplus, it is a waste of resource and many bikes will pile up at the roadside and cause traffic congestion. It is a severe problem in China, notably for Mobike and ofo (two popular dockless

bike sharing companies), who launched many bikes to compete for market share, which results in a tremendous waste of resources ([3]). Thus, in this paper we provide an optimization function to determine the optimal fleet size.

Bikes are scattered in the system asymmetrically, so repositioning is a necessary measure to preserve the sustainability of DBSS. There are two classes of repositioning strategies: static repositioning and dynamic repositioning. Static repositioning does not take into consideration the movements of bikes during the repositioning period, while dynamic repositioning considers the fluctuation of bikes during the repositioning process. This paper proposes a repositioning flow to rebalance the DBSS, which is a novel dynamic repositioning process.

Among the model technologies, the closed queuing network (CQN) is an appropriate method to study DBSSs, which can give sufficient consideration to the stochasticity of the system. CQN is well known as a suitable tool for studying complex systems. However there is a profound difficulty in theory to compute a large scale system because of the normalization constant. The mean value analysis (MVA) algorithm is a common algorithm to calculate the CQN without the normalization constant. Since there are plenty of road nodes, the number of nodes in the DBSS is huge. Based on the special construction of DBSS, we use the flow equivalent server (FES) algorithm to partition a parking region and its downlink road node into a subnetwork, which lessens the number of nodes significantly.

The contributions found in this paper are threefold: The first one is to use the CQN to describe a DBSS and propose an optimization function to find the optimal fleet size. The second one is to propose an optimization function to determine the repositioning flow. The third one is to use the FES algorithm to study the DBSS, which is the first time this algorithm is applied to analyze the DBSS by dividing a parking region and its downlink roads into a subnetwork to reduce the computation greatly.

The remainder of this paper is organized as follows. Section 2 provides the literature review. Section 3 sets system parameters and model description of the DBSS. Section 4 establishes a CQN to express the DBSS and gives the product form solution. Section 5 provides the MVA algorithm to solve the CQN. Section 6 proposes the FES algorithm to simplify the computation of the DBSS. Section 7 provides optimization functions for determining the optimal fleet size and repositioning flow. Section 8 gives numerical experiments. The conclusion is given in Section 9.

2. Literature Review

The research of fleet sizing is a vital topic of BSS. Some work has provided different ideas to determine the optimal fleet size. Kóchel et al. [4] provided a joint simulation optimization idea of fleet size and bike repositioning strategy and proposed an iterative means to obtain the fleet size. Fanti et al. [5] viewed an electric car sharing system as a CQN to determine the optimal fleet size and developed an optimization problem to maximize the system revenue aiming to find the optimal fleet size. Zhai et al. [6] proposed a sparse matrix construction solution method for determining the fleet size of the DBSS. Bazan et al. [7] studied mobility-on-demand networks, which are similar to BSS. They presented a simulation model combined with optimization function and queuing network and computed the optimal fleet size. These papers provide different methods to find the optimal fleet size. However, in order to consider the stochasticity fully, we choose the CQN to model the DBSS. We determine the optimal fleet size based on an optimization function, in which the parameters are obtained by computing the CQN.

Another important measure of BSS is repositioning management. For the static repositioning, Liu et al. [8] studied a DBSS with multiple heterogeneous vehicles, depots, and visits. Their conclusion was that the enhanced chemical reaction optimization (CRO) had better solutions than the preliminary CRO. Schuijbroek et al. [9] proposed a mixed-integer programming based on decomposing the multi-vehicle repositioning matters into single-vehicle problems. They also provided a heuristic of cluster-first route-second to mitigate its running time. For other static repositioning research, readers can refer to Szeto et al. [10], Bulhões et al. [11], Jia et al. [12], and Tang et al. [13]. Static repositioning typically can not adequately capture the surges in user demand. Thus dynamic repositioning needs to

be studied sufficiently. For the dynamic repositioning, Shu et al. [14] found that repositioning execution can get an additional 15–20% trips. Caggiani et al. [15] carried out dynamic bike repositioning at a constant gap time. The aim is to get a high user satisfaction and a low repositioning cost. Legros et al. [16] developed an implementation decision-support tool to decide the number of bikes and which station to carry out the repositioning. They also proved that there is an optimal inventory level. Mellou and Jaillet [17] proposed a novel mixed-integer programming formulation to solve the dynamic repositioning problem and provided a linear programming model to capture the bike flows from all trips. The authors of these studies computed the repositioning flows requiring a huge number of trucks, but in this paper we relax this requirement by setting the availability of parking regions to a certain level. We also provide a probabilistic guarantee for meeting any given service level.

CQN is an appropriate method to model BSS. George et al. [18] modeled a vehicle rental system with fixed stations as a CQN. They analyzed the asymptotic behavior of vehicle availability and provided an optimization function for determining the optimal fleet size. Li et al. [19] analyzed an SBBSS as a virtual CQN, in which parking regions and roads were considered as “virtual nodes”. This was the first time that road nodes were analyzed in detail in the CQN. Li et al. [20] extended the CQN model of Li et al. [19] from three points: (i) from strong connections between stations to an irreducible graph of stations; (ii) from Poisson arrival of users to Markovian arrival processes; (iii) from exponential distributions of riding-bike times on the roads to phase-type distributions. Furthermore, Li et al. [21] developed a fluid and diffusion approximation of a multiclass CQN to analyze the SBBSS. In this paper, we extend the computation of CQN by applying the MVA and FES algorithms. Further, some other important examples include Zhang et al. [22], Fricker et al. [23], Mizuno et al. [24], Calafiore et al. [25], Samet et al. [26], Iglesias et al. [27], and Vishkai et al. [28].

3. Model Description

In this section, we give a detailed model description for the DBSS.

(1) Parking regions: In the DBSS, parking regions are viewed as nodes. Since the geographical locations and surrounding traffic environment may be different, we determine that the parking regions are different. We assume that the parking space of each parking region is sufficient so that bikes can always be parked upon arrival at a parking region immediately, that is, having no blocking effects. We assume that there are N parking regions distributed in the DBSS and denote the set of parking region nodes by $S = \{1, 2, \dots, N\}$.

(2) Roads: Parking regions are connected by roads. We express the set of roads as I . Since there are origin-direction links of parking regions, we view the roads between parking regions are directed. Road $i \rightarrow j$ and Road $j \rightarrow i$ are different roads for $1 \leq i, j \leq N$. We view Road $i \rightarrow j$ as a node with finite servers and assume that the travel time from Parking region i to Parking region j is exponential with mean random variable $1/\mu_{ij}$.

We denote the set of downlink parking regions of Parking region i by Θ_i . Similarly, we denote the set of uplink parking regions of Parking region i by Δ_i .

To express the set of parking regions starting from Parking region i for $1 \leq i \leq N$, we write

$$R(i) = \{\text{Road } i \rightarrow j : j \in \Theta_i\}.$$

The number of directed roads in the set $R(i)$ at most is $N - 1$. The number of elements in the set $R(i)$ is $|R(i)|$. We let $M = \sum_{i=1}^N |R(i)|$, thus $M \leq N(N - 1)$.

We assume that there is an irreducible path in the DBSS. The irreducibility is guaranteed by a suitable road construction with $R(i)$ for $1 \leq i \leq N$. The set of road nodes is denoted by $I = \{N + 1, N + 2, \dots, M\}$. For the detailed expression of irreducible path, readers can refer to Li et al. [20].

(3) User arrival processes: We assume that the arrivals of users at Parking region i is a Poisson process and set the mean arrival rate as $\lambda_i > 0$ for $1 \leq i \leq N$.

(4) Bike using processes: When a user arrives at Parking region i with at least one bike, he rents a bike and rides it on the Road $i \rightarrow j$ with probability p_{ij} for $j \in \Theta_i$ and $\sum_{j \in \Theta_i} p_{ij} = 1$ for each $i = 1, 2, \dots, N$. The service times on the Road $i \rightarrow j$ are i.i.d. and exponential with mean rate $\mu_{ij} > 0$.

(5) The departure discipline: The users will leave the system with two cases: (i) when a user arrives at a parking region that has no bikes, he will leave the system directly. (b) When a user finishes his trip and returns the bike to a parking region successfully, he leaves the system immediately.

All the random variables above are independent of each other.

At each parking region, bikes form a queue while waiting for users to rent. Thus, from the perspective of bikes, we consider the parking regions as single-server (SS) nodes and the roads as finite-server (FS) nodes (the number of the servers on the road is less than or equal to the total number of bikes in the DBSS). Parking region i can be viewed as a server with service rate λ_i and follows First Come First Served (FCFS) scheduling for $1 \leq i \leq N$. The service time of a parking region is the interval time of customer arrival. A bike leaves an SS node and moves to an FS node. Parking region nodes are viewed as M/M/1 queuing systems, while road nodes are modeled as M/M/n queuing systems. Figure 1 illustrates a DBSS with three parking regions.

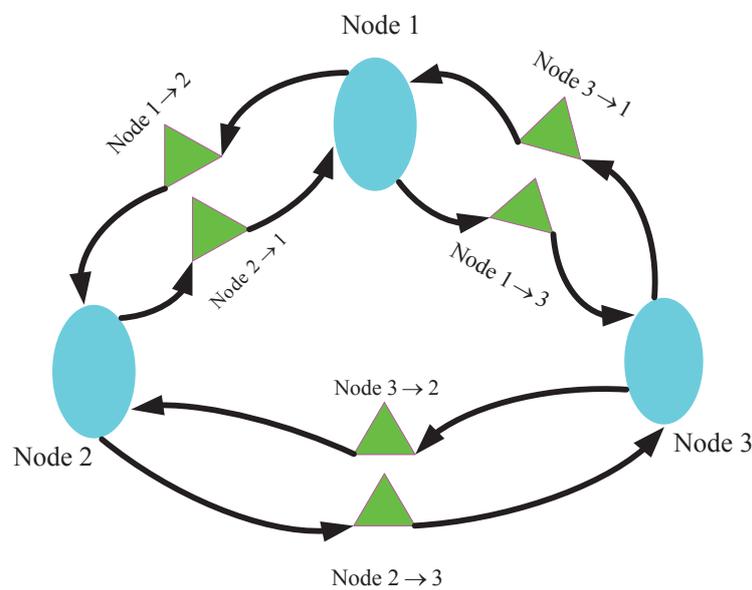


Figure 1. The structure of a dockless bike sharing system (DBSS) with three parking regions.

4. Closed Queuing Network

All the bikes are just circulating among parking regions and roads in the DBSS, and the number of bikes in a DBSS is fixed. From bikes’ perspective, we model the DBSS as a closed queuing network.

First, we let $n_i(t)$ denote the number of bikes in Node i at time t and $\mathbf{Q}(t) = (n_1(t), n_2(t), \dots, n_{N+M}(t))$. Thus $\{\mathbf{Q}(t) : t \geq 0\}$ is a Markov process. When a user rents a bike at Parking region i , he selects a destination of Parking region j with probability p_{ij} , where $p_{ij} > 0$, $p_{ii} = 0$, and $\sum_{j \in \Theta_i} p_{ij} = 1$. For each $i \in I$, we define $u(i)$ and $d(i)$ to be the uplink and downlink nodes of i , which denote the origin and destination nodes of i , respectively. The elements of routing matrix P are written as

$$f_{ij} = \begin{cases} p_{ij}, & i \in S, j \in I, i = u(j), \\ 1, & i \in I, j \in S, i = d(j), \\ 0, & \text{otherwise.} \end{cases}$$

The service time of each node is exponential. For $i \in S$, $\mu_i = \lambda_i$. For Node $i \rightarrow j$, $\mu_{ij} = n_{ij}\mu_{ij}$ where n_{ij} is the number of bikes on Node $i \rightarrow j$.

We set n_i as the number of bikes in Node i . To study the CQN, the service rates, the routing matrix and relative arrival rates are needed to be determined first. For a given fleet size K , the state space of the CQN is $\Omega = \left\{ \mathbf{n} : \sum_{i=1}^{N+M} n_i(t) = K, 0 \leq n_i \leq K \right\}$, where $\mathbf{n} = (n_1, n_2, \dots, n_N, n_{N+1}, n_{N+2}, \dots, n_{N+M})$. The relative arrival rate e_i is written as

$$e_j = \sum_{i=1}^{N+M} e_i f_{ij}. \quad (1)$$

Our network consists of finite-server nodes and single-server nodes with exponential service time, so this model is a BCMP network, which has a product form solution (See Baskett et al. [29]). The product form solution is given by

$$\mathfrak{f}(\mathbf{n}) = \frac{1}{G} \prod_{i=1}^{N+M} \mathfrak{f}_i(n_i), \quad (2)$$

in which

$$\mathfrak{f}_i(n_i) = \begin{cases} \left(\frac{e_i}{\lambda_i} \right)^{n_i}, & i \in S, \\ \left(\frac{e_i}{\mu_i} \right)^{n_i} \frac{1}{n_i!}, & i \in I. \end{cases}$$

Note that $\mathbf{n} \in \Omega$ and

$$G = \sum_{\mathbf{n} \in \Omega} \prod_{i=1}^{N+M} \mathfrak{f}_i(n_i).$$

For the detailed proof of the product form solution, readers can refer to Baskett et al. [29]. For the DBSS, there are two vital performance measures:

(1) Expected revenue E charged by the bikes on roads

$$E = K - \sum_{i \in S} n_i.$$

(2) Service level A_i measured by the mean bike availability

$$A_i = 1 - p(n_i = 0), i \in S.$$

Expected revenue E is the utmost concern of operators. The value A_i is an acknowledged measure of the quality of service. Incentive measures can be implemented based the results of A_i . For example, measures include guiding customers to rent bikes at parking regions with high relative utilization and return bikes to parking regions with lower utilization.

5. MVA Algorithm of the Closed Queuing Network

Since the computation of normalization constant G is expensive, we apply the MVA algorithm (see Reiser and Lavenberg [30]) to compute the CQN by avoiding the calculating of the normalization constant. Through the MVA algorithm, we can calculate the mean waiting times $\bar{T}_i(k)$ and the mean queue lengths $\bar{K}_i(k)$ at each Node i of a CQN, where $k = 1, \dots, K$.

Based on the the MVA algorithm, we can get the mean response time at the i th node

$$\bar{T}_i(k) = \begin{cases} \frac{1}{\mu_i} [1 + \bar{K}_i(k-1)], & i \in S, \\ \frac{1}{\mu_i}, & i \in I. \end{cases} \quad (3)$$

Based on $\bar{T}_i(k)$, the overall throughput is given by

$$\lambda(k) = \frac{k}{\sum_{i=1}^{N+M} e_i \bar{T}_i(k)}, \quad (4)$$

and the throughput of each node can be written as

$$\lambda_i(k) = \lambda(k) e_i, \text{ for } i = 1, 2, \dots, N + M, \quad (5)$$

in which e_i is determined by Equation (1).

Based on $\bar{T}_i(k)$ and $\lambda_i(k)$, the mean number of bikes at the i th node is written as

$$\bar{K}_i(k) = \lambda_i(k) \bar{T}_i(k). \quad (6)$$

We provide the detailed computation by MVA in Algorithm 1.

Algorithm 1 Mean value analysis (MVA) algorithm for closed queuing network.

Step 1. Initialization

$\bar{K}_i(k) = 0$, for $i = 1, 2, \dots, N + M$. $G(0) = 1$.

Step 2. Computation for the relative arrival rates

Let $e_1 = 1$. Compute relative arrival rates e_j for $j = 2, 3, \dots, N + M$, based on equation $e_j = \sum_{i=1}^{N+M} e_i p_{ij}$ for $i = 1, 2, \dots, N + M$.

Step 3. Iteration over the number of bikes in the network starting with $k = 1$:

Step 3.1 Compute mean response time based on equation $\bar{T}_i(1) = \frac{1}{\mu_i} [1 + \bar{K}_i(0)]$ for $i = 1, 2, \dots, N$, and $\bar{T}_i(1) = \frac{1}{\mu_i}$ for $i = N + 1, \dots, N + M$.

Step 3.2 Compute the throughput based on equation $\lambda(1) = \frac{1}{\sum_{i=1}^{N+M} e_i \bar{T}_i(1)}$.

Get the normalization constant based on equation $G(1) = \frac{G(0)}{\lambda(1)}$.

Step 3.3 Compute the mean number of bikes at each node based on equation $\bar{K}_i(1) = e_i \lambda(1) \bar{T}_i(1)$.

Step 4. Iteration over the number of bikes in the network for $k = 2, \dots, K$.

Step 4.1 Compute mean response time based on equation $\bar{T}_i(k) = \frac{1}{\mu_i} [1 + \bar{K}_i(k-1)]$ for $i = 1, 2, \dots, N + M$, and $\bar{T}_i(k) = \frac{1}{\mu_i}$ for $i = N + 1, \dots, N + M$.

Step 4.2 Compute the throughput based on equation $\lambda(k) = \frac{k}{\sum_{i=1}^{N+M} e_i \bar{T}_i(k)}$.

Get the normalization constant based on equation $G(k) = \frac{G(k-1)}{\lambda(k)}$.

Step 4.3 Compute the mean number of bikes at each node based on equation $\bar{K}_i(k) = e_i \lambda(k) \bar{T}_i(k)$.

Step 4.4 Get $\bar{T}_i(K)$ and $\bar{K}_i(K)$ for $i = 1, 2, \dots, N + M$.

Step 5. Computation for performance measures E_i and A_i in the closed queuing network.

We are interested in the availability of bikes at each parking region (i.e., the probability that there is at least one bike at the parking region). Set $\bar{T}_i(0) = \bar{K}_i(0) = 0$. By iteration, we can get

$$E_i = \bar{K}_i(K), i = 1, 2, \dots, N,$$

and

$$A_i = \frac{E_i}{1 + E_i}, i = 1, 2, \dots, N.$$

For small networks, this algorithm provides a method to obtain the optimal fleet size straightly. However, for a larger network, the computation is too expensive. Thus, we need to find another approach to reduce the computation.

6. FES Algorithm of the Closed Queuing Network

MVA algorithm can be computed directly without calculating the normalizing constant, but it needs high storage requirement and expensive computation. So we try to find another approach to reduce the computation. We use the FES algorithm to compute the DBSS.

In the closed queuing network of DBSS, since bikes are served in parking regions as well as on roads, we divide the nodes into SS nodes and FS nodes. However, when we consider the two performance measures: expected revenue E_i and available rate A_i , we only consider the results of SS nodes. Since the difference between parking region nodes and road nodes, we try to simplify the computation by partitioning them.

FES method is based on Norton's theorem, refer to Chandy et al. [31] and Akyildiz et al. [32]. Based on the FES method, we partition the closed queuing network into N disjoint subnetworks, in which Node i and node $i \rightarrow j$ are a subnetwork for $i = 1, \dots, N, j \in \Theta_i$. We write the subnetwork as SN_1, SN_2, \dots, SN_N . Analyzing one of the subnetworks, we need to short-circuit all the other nodes that do not belong to the subnetwork.

First, we need to determine the throughput $\lambda_{SN_j}(k)$ and the normalization constant $G_j(k)$ for $1 \leq j \leq N$ and $1 \leq k \leq K$. The arrival rates of nodes in the subnetwork are the same as the original network. Consider each subnetwork as an FES node and construct an equivalent reduced network. The dependent service rates μ_{ej} of the FES Node j are identical with the throughputs in the corresponding j th subnetwork, which is written as

$$\mu_{ej}(k) = \lambda_{SN_j}(k), \text{ for } j = 1, \dots, N \text{ and } k = 1, \dots, K.$$

We provide the detailed FES in Algorithm 2. In Algorithm 2, the time and storage requirement are reduced considerably.

Algorithm 2 Flow equivalent server (FES) algorithm for closed queuing network.

Step 1. Partition the network into N subnetworks.

Partition the network into subnetworks SN_1, SN_2, \dots, SN_N , in which Node i and Node $i \rightarrow j$ constitute SN_i subnetwork for $i = 1, \dots, N, j \in \Theta_i$.

Step 2. Analyze each subnetwork and construct the reduced network

For subnetwork 1. Short-circuit all the other subnetworks. Use the MVA algorithm to compute throughputs and the normalization constants of the subnetwork. Then we get $\lambda_{SN_1}(1), \lambda_{SN_1}(2), \dots, \lambda_{SN_1}(K)$ and $G_1(1), G_1(2), \dots, G_1(K)$.

Analogously, analyze subnetworks 2 to N , and obtain $\lambda_{SN_j}(1), \lambda_{SN_j}(2), \dots, \lambda_{SN_j}(K)$ and $G_j(1), G_j(2), \dots, G_j(K)$, for $j = 2, 3, \dots, N$. We get $\mu_{ei}(k) = \lambda_{SN_i}(k)$ for $k = 1, 2, \dots, K$, and $i = 2, 3, \dots, M$.

Step 3. Compute the normalization constants.

Compute the normalization constant of the reduced network by convoluting the N normalizing vectors, based on the equation $G = G_1 \otimes G_2 \otimes \dots \otimes G_M$.

Step 4. Computation for performance measures in the CQN.

Compute the mean number of bikes in Node i by $\bar{K}_i(K) = \sum_{k=1}^K \left(\frac{e_i}{\mu_i}\right)^k \frac{G(K-k)}{G(K)}$ for $i = 1, 2, \dots, N$.

The utilization of Node i by $\rho_i(K) = \frac{e_i}{\mu_i} \frac{G(K-1)}{G(K)}$ for $i = 1, 2, \dots, N$.

The mean response time of Node i by $\bar{T}_i(K) = \sum_{k=1}^K \left(\frac{e_i}{\mu_i}\right)^k \frac{G(K-k)}{e_i G(K-1)}$ for $i = 1, 2, \dots, N$.

7. Performance Analysis

In this section, we provide the results of a performance analysis of DBSSs. Based on the MVA and FES algorithms, we get the results of the expected number of bikes and the availability for each parking region. Then, we propose two optimization functions to determine the optimal fleet size and the repositioning flow based on the results.

7.1. Optimal Fleet Size

Since the optimal fleet size is a tradeoff between the earnings got from users and maintaining cost of the fleet, we present an optimization function to determine the optimal fleet size. Let r_i be the revenue obtained from a bike being in use at road nodes per-unit-time. The BSS will cause an unavailability cost c_j whenever bikes park at parking regions idly. Then the objective is to maximize the profit

$$F(K) = \max_{\{X_i\}} \sum_{i \in I} E_i(K) r_i - \sum_{j \in S} c_j \lambda_j (1 - A_j(K)), \quad (7)$$

in which E_i is the expected queue length at Node i for $i \in I$ and A_j is bike availability at Node j for $j \in S$ in steady state.

Based on Shanthikumar and Yao [33] and George et al. [18], we get that for a single-class CQN with FCFS exponential servers, both the throughput at each parking region and the system throughput are non-decreasing concave functions of fleet size. The values $\{A_j(K), j \in S\}$ and $\{E_i(K), i \in I\}$ are all non-decreasing concave functions, therefore $F(K)$ is concave.

To solve Equation (7), we can calculate the MVA algorithm iteratively for each $K \in \{1, 2, \dots\}$ and compute the associated profit. Since the concavity of $F(K)$, the method will terminate once profit decreases from K to $K + 1$.

7.2. Repositioning Flow

Since bikes will inevitably accumulate at one or more parking regions in the DBSS. To ease this problem, we introduce a set of repositioning flow to rebalance the system, i.e., moving bikes to parking regions that lack bikes. We use y_{ij} to denote the rate of bikes repositioning flow from Parking region i to Parking region j . We discuss how to achieve a target service level with different bike resources.

We assume that $Y_i = \sum_{j \in \Theta_i} y_{ij}$. The availability of Parking region i is given by

$$A_i = \frac{e_i}{\lambda_i + Y_i}.$$

If we set the availability A_i of Parking region i equal to 1, it needs a huge number of bikes idly waiting at each parking region to meet the availability (refer to Li et al. [34]). Thus, we set the objective as a service level $\min\{A_i\} \geq \sigma \in (0, 1)$. Therefore, we can realize the certain level of availability and release stressful demand of the fleet size. Then the repositioning cost minimization problem can be written by

$$\begin{aligned} \min_{\{X_i\}} \sum_{i \in S} b_i Y_i A_i \\ \text{s.t. } A_i \geq \sigma, i \in S, \end{aligned}$$

where b_j is the cost of maintaining bikes and $b_j = \sum_{i \in \Theta_j} b_{ij} p_{ij}$.

We set $\xi_i = e_i / \lambda_i$, $\bar{\xi}_i = \max\{e_i / \lambda_i\}$, $\gamma_i = e_i / (\lambda_i + X_i)$, and $\bar{\gamma}_i = \max\{e_i / (\lambda_i + X_i)\}$ for $i \in S$. Since $A_i = \gamma_i / \bar{\gamma}_i$, the optimal equation can be rewritten as

$$\min_{\{X_i\}} \sum_{i \in S} b_i Y_i \frac{e_i}{\bar{\gamma}_i (\lambda_i + Y_i)}$$

Table 1. The service rates and the relative arrival rates.

μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9
10	8	6	1	3	2	4	2	6
e_1	e_2	e_3	e_4	e_5	e_5	e_7	e_8	e_9
1	0.8889	1.2222	0.4000	0.2667	0.6000	0.7333	0.6222	0.4889

Using the FES algorithm, we divided the nine nodes into three subnetworks. i.e., Node 1, Node 4 and Node 6 are subnetwork 1; Node 2, Node 5, and Node 8 are subnetwork 2; Node 3, Node 7, and Node 9 are subnetwork 3. We computed the DBSS by Algorithm 1 and 2 respectively and verified that the results are consistent. We provide the results in Table 2.

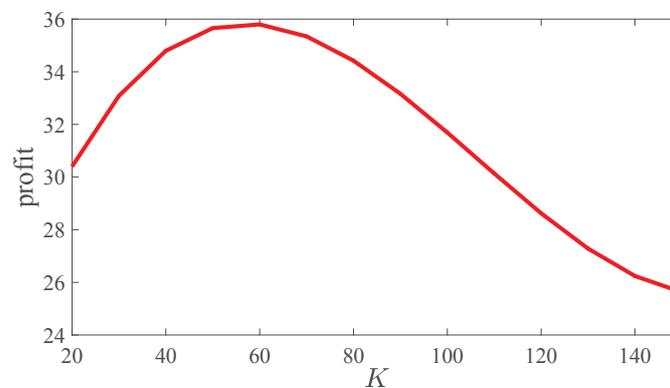
Table 2. The expected number obtained by MVA and FES algorithms.

\bar{K}_1	\bar{K}_2	\bar{K}_3	\bar{K}_4	\bar{K}_5	\bar{K}_6	\bar{K}_7	\bar{K}_8	\bar{K}_9
8.3333	15.3846	12.0376	1.3625	0.2857	2.9981	0.8461	3.4963	0.2558

8.2. Optimal Fleet Size and Repositioning Flow

(1) The three parking region DBSS

Let the hourly rental fee r_i for $i \in I$ be \$2 and the maintenance cost c_i per-bike per-hour for $i \in S$ be \$0.20. By Algorithm 2 and Equation (7), we compute the three parking region DBSS depicted in Figure 2. The variation of profits corresponding to fleet size is shown in Figure 3. We can find that the optimal fleet size is 60. This method considers the differences between parking regions and loose structures of the routing matrix. We can find that the profit first increases and then decreases as the fleet size K increases. This result offers operators quantitative basis of the fleet size design.

**Figure 3.** The profit vs. fleet size for a three parking region DBSS.

When setting $\min_{i \in S} \{A_i\} \geq 0.9$, we get the optimal repositioning strategy to add rebalancing flows $1 \rightarrow 4$ with a rate of 0.0641, $1 \rightarrow 6$ with a rate of 0.0547, and $3 \rightarrow 8$ with a rate of 0.1431. The $\min_{i \in S} \{A_i\} \geq 0.9$. The computation of repositioning flow considers the dynamics of DBSS, which is close to the actual operation.

(2) A 50 parking region DBSS

By Algorithm 2, we computed a DBSS with 50 parking regions. We partitioned the DBSS into 50 subnetworks and depict the optimal fleet size in Figure 4. The optimal repositioning flows are listed in Table 3, in which $y_{i,j}$ is the flow from Parking region i to Road $i \rightarrow j$. Algorithm 2 extends the computation of CQN of DBSS greatly. Since there are actual service processes on the road in the DBSS, when we analyze the system we must study the service rates on the road nodes. It is well known that the number of road nodes is enormous in a DBSS, which will cause the state space explosion and lead to the computation of normalization constant too expensive. By partitioning the system into subnetworks,

the number of nodes is reduced significantly, which realizes the performance computation of larger size DBSS. Thus the result provides operators with valuable quantitative reference.

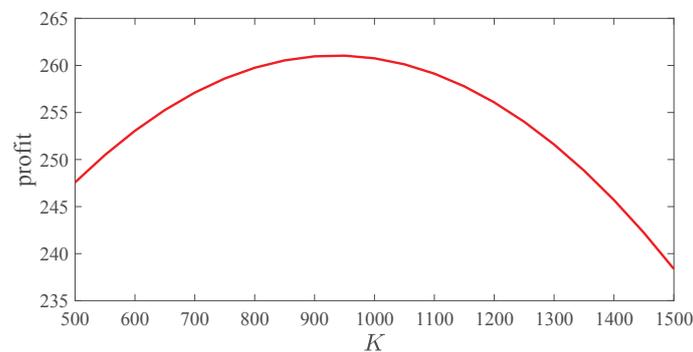


Figure 4. The profit vs. fleet size for a 50 parking region DBSS.

Table 3. The expected number obtained by MVA and FES algorithms.

$y_{1,2}$	$y_{4,5}$	$y_{8,7}$	$y_{10,9}$	$y_{11,10}$	$y_{13,14}$	$y_{17,18}$	$y_{24,23}$	$y_{25,26}$	$y_{28,27}$
1.3513	2.3468	2.0276	0.3215	1.8527	1.9812	0.4621	2.9613	0.0358	1.7852
$y_{29,30}$	$y_{31,32}$	$y_{35,36}$	$y_{37,36}$	$y_{39,40}$	$y_{41,40}$	$y_{43,42}$	$y_{44,45}$	$y_{47,48}$	$y_{50,1}$
1.4562	0.1383	1.8246	1.0037	1.6125	0.2471	2.8132	0.8160	3.2613	1.8421

9. Conclusions

In this paper, we provide a closed queuing network of dockless bike sharing systems. To avoid the expensive computation of the normalization constant, we use the MVA algorithm and the FES algorithm to analyze the small size and larger size DBSS, respectively. The FES algorithm is proposed to compute the DBSS for the first time. We let a parking region and its downlink roads as a subnetwork and partition the DBSS into different subnetworks. The FES algorithm reduces the number of nodes greatly, and thus the computation of the CQN is lessened significantly. This paper opens a new avenue to study the CQN of DBSS. Based on the computation results of the CQN, we determine the optimal fleet size and repositioning flow by two optimization functions. The computation of repositioning flow adequately considers the dynamics of bikes during the repositioning process, which is close to the actual DBSS. According to our analysis, the method can be used as guidelines for designing and controlling the fleet and arrange the repositioning flow to achieve a high level of service.

For future work, the CQN model presented in this paper can be used to analyze other on-demand transportation networks. Another upcoming issue is that we can consider the unusable bikes in the DBSS. Since the failure rates of bikes are high, unusable bikes account for a large proportion in the DBSS. Therefore when determining the optimal fleet and the repositioning flow, the failure rate of bikes can be introduced in.

Author Contributions: Conceptualization, R.-N.F.; formal analysis, R.-N.F.; investigation, F.-Q.M.; methodology, R.-N.F. and F.-Q.M.; resources, R.-N.F.; supervision, F.-Q.M. and Q.-L.L.; writing—original draft, R.-N.F.; writing—review and editing, R.-N.F. and Q.-L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fund of Innovation for Postgraduate of Hebei Province (No. CXZZBS2017055).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BBS	bike sharing system
DBSS	dockless bike sharing system
SBBSS	station-based bike sharing system
MVA	mean field analysis
FES	flow equivalent server
FCFS	first-come-first-service
SS	single-server
FS	finite-server
CRO	chemical reaction optimization

References

1. The Bike-Sharing World Map. Available online: https://www.google.com/maps/d/viewer?mid=1UxYw9YrwT_R3SGsktJU3D-2GpMU&ll=0%2C0&z=1 (accessed on 1 February 2020).
2. Shi, J.G.; Si, H.; Wu, G.; Su, Y.; Lan, J. Critical factors to achieve dockless bike-sharing sustainability in China: A stakeholder-oriented network perspective. *Sustainability* **2018**, *10*, 2090. [CrossRef]
3. The News about Bike Sharing in China. Available online: <https://baijiahao.baidu.com/s?id=1627399223714557603&wfr=spider&for=pc> (accessed on 1 February 2020).
4. Kóchel, P.; Kunze, S.; Nieländer, U. Optimal control of a distributed service system with moving resources: Application to the fleet sizing and allocation problem. *Int. J. Prod. Econ.* **2019**, *81*, 443–459. [CrossRef]
5. Fantì, M.P.; Mangini, A.M.; Pedroncelli, G.; Ukovich, W. Fleet sizing for electric car sharing system via closed queuing networks. In Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), San Diego, CA, USA, 5–8 October 2014; pp.1324–1329.
6. Zhai, Y.; Liu, J.; Du, J.; Chen, J. Solution to fleet size of dockless bike sharing station based on matrix analysis. In Proceedings of the ISPRS Annals of Photogrammetry, Remote Sensing Spatial Information Sciences, Volume IV-4, 2018 ISPRS TC IV Mid-term Symposium “3D Spatial Information Science – The Engine of Change”, Delft, The Netherlands, 1–5 October 2018; pp. 255–262.
7. Bazan, P.; Djanatliev, A.; Pruckner, M.; German, R.; Lauer, C. Rebalancing and fleet sizing of mobility-on-demand networks with combined simulation, optimization and queuing network analysis. In Proceedings of the 2018 Winter Simulation Conference (WSC), Gothenburg, Sweden, 9–12 December 2018; pp. 1527–1538.
8. Liu, Y.; Szeto, W.Y.; Ho, S.C. A static free-floating bike repositioning problem with multiple heterogeneous vehicles, multiple depots, and multiple visits. *Transport. Res. C-Emer.* **2018**, *92*, 208–242. [CrossRef]
9. Schuijbroek, J.; Hampshire, R.C.; Van Hoes, W.J. Inventory rebalancing and vehicle routing in bike sharing systems. *Eur. J. Oper. Res.* **2017**, *257*, 992–1004. [CrossRef]
10. Szeto, W.Y.; Shui, C.S. Exact loading and unloading strategies for the static multi-vehicle bike repositioning problem. *Transport. Res. B-Meth.* **2018**, *109*, 176–211. [CrossRef]
11. Bulhões, T.; Subramanian, A.; Erdoğan, G.; Laporte, G. The static bike relocation problem with multiple vehicles and visits. *Eur. J. Oper. Res.* **2018**, *264*, 508–523. [CrossRef]
12. Jia, H.; Miao, H.; Tian, G.; Zhou, M.; Feng, Y.; Li, Z.; Li, J. Multiobjective bike repositioning in bike-sharing systems via a modified artificial bee colony algorithm. *IEEE Trans. Autom. Sci. Eng.* **2019**, 1–12. [CrossRef]
13. Tang, Q.; Fu, Z.; Qiu, M. A bilevel programming model and algorithm for the static bike repositioning problem. *J. Adv. Transport.* **2019**, in press. [CrossRef]
14. Shu, J.; Chou, M.C.; Liu, Q.; Teo, C.P.; Wang, I.L. Models for effective deployment and redistribution of bicycles within public bicycle-sharing systems. *Oper. Res.* **2013**, *61*, 1346–1359. [CrossRef]
15. Caggiani, L.; Camporeale, R.; Ottomanelli, M.; Szeto, W.Y. A modeling framework for the dynamic management of free-floating bike-sharing systems. *Transport. Res. C-Emer.* **2018**, *87*, 159–182. [CrossRef]
16. Legros, B. Dynamic repositioning strategy in a bike-sharing system; how to prioritize and how to rebalance a bike station. *Eur. J. Oper. Res.* **2019**, *272*, 740–753. [CrossRef]

17. Mellou, K.; Jaillet, P. Dynamic resource redistribution and demand estimation: An application to bike sharing systems. *SSRN* **2019**. [[CrossRef](#)]
18. George, D.K.; Xia, C.H. Fleet-sizing and service availability for a bike rental system via closed queuing networks. *Eur. J. Oper. Res.* **2011**, *211*, 198–207. [[CrossRef](#)]
19. Li, Q.L.; Fan, R.N.; Ma, J.Y. A unified framework for analyzing closed queuing networks in bike sharing systems. In *Information Technologies and Mathematical Modelling - Queuing Theory and Applications*; Dudin, A., Gortsev, A., Nazarov, A., Yakupov, R., Eds.; Communications in Computer and Information Science, Volume 638; Springer: Cham, Switzerland, 2016; pp. 177–191.
20. Li, Q.L.; Fan, R.N.; Qian, Z.Y. A nonlinear solution to closed queuing networks for bike sharing systems with Markovian arrival processes and under an irreducible path graph. In *Queuing Theory and Network Applications*; Yue, W., Li, Q.L., Jin, S., Ma, Z., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2017; Volume 10591, pp. 118–140.
21. Li, Q.L.; Qian, Z.Y.; Fan, R.N. Fluid and diffusion limits for bike sharing systems. In *Queuing Theory and Network Applications*; Yue, W., Li, Q.L., Jin, S., Ma, Z., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2017; Volume 10591, pp. 217–245.
22. Zhang, R.; Rossi, F.; Pavone, M. Analysis, control, and evaluation of mobility-on-demand systems: A queuing-theoretical approach. *IEEE Trans. Control Netw.* **2015**, *6*, 115–126. [[CrossRef](#)]
23. Fricker, C.; Serval, N. Two-choice regulation in heterogeneous closed networks. *Queuing Syst.* **2016**, *82*, 173–197. [[CrossRef](#)]
24. Mizuno, S.; Iwamoto, S.; Seki, M.; Yamaki, N. Proposal for optimal placement platform of bikes using queuing networks. *SpringerPlus* **2016**, *5*, 2071. [[CrossRef](#)]
25. Calafiore, G.C.; Portigliotti, F.; Rizzo, A. A network model for an urban bike sharing system. *IFAC-PapersOnLine* **2017**, *50*, 15633–15638. [[CrossRef](#)]
26. Samet, B.; Couffin, F.; Zolghadri, M.; Barkallah, M.; Haddar, M. Performance analysis and improvement of the bike sharing system using closed queuing networks with blocking mechanism. *Sustainability* **2018**, *10*, 4663. [[CrossRef](#)]
27. Iglesias, R.; Rossi, F.; Zhang, R.; Pavone, M. A BCMP network approach to modeling and controlling autonomous mobility-on-demand systems. *Ind. Robot.* **2019**, *38*, 357–374. [[CrossRef](#)]
28. Vishkaei, B.M.; Mahdavi, I.; Mahdavi-Amiri, N.; Khorram, E. Balancing public bicycle sharing system using inventory critical levels in queuing network. *Comput. Ind. Eng.* **2020**, 106277. [[CrossRef](#)]
29. Baskett, F.; Chandy, K.M.; Muntz, R.R.; Palacios, F.G. Open, closed, and mixed networks of queues with different classes of users. *J. ACM* **1975**, *22*, 248–260. [[CrossRef](#)]
30. Reiser, M.; Lavenberg, S.S. Mean-value analysis of closed multichain queuing networks. *J. ACM* **1980**, *27*, 313–322. [[CrossRef](#)]
31. Chandy, K.M.; Herzog, U.; Woo, L. Parametric analysis of queuing networks. *IBM J. Res. Dev.* **1975**, *19*, 36–42. [[CrossRef](#)]
32. Akyildiz, I.F.; Bolch, G.; Paterok, M. Die erweiterte parametrische Analyse für geschlossene Warteschlangennetze. In *Messung, Modellierung und Bewertung von Rechensystemen*; Beilner, H., Ed.; Informatik—Fachberichte; Springer: Berlin/Heidelberg, Germany, 1985; Volume 110, pp. 170–185.
33. Shanthikumar, J.G.; Yao, D.D. Second-order properties of the throughput of a closed queuing network. *Math. Oper. Res.* **1988**, *13*, 524–534. [[CrossRef](#)]
34. Li, S.; Luo, Q.; Hampshire, R. Optimizing large on-demand transportation systems. *SSRN* **2019**. [[CrossRef](#)]

