Dynamic repositioning strategy in a bike-sharing system; how to prioritize and how to rebalance a bike station

Benjamin Legros

Ecole de Management de Normandie, Laboratoire Métis, 64 Rue du Ranelagh, 75016 Paris, France

benjamin.legros@centraliens.net

Abstract

Bike-sharing systems are becoming increasingly popular in large cities. The natural imbalance and the stochasticity of bike's arrivals and departures lead operators to develop redistribution strategies in order to ensure a sufficiently high quality of service for users. Using a Markov decision process approach, we develop an implementable decision-support tool which may help the operator to decide at any point of time (i) which station should be prioritized, and (ii) which number of bikes should be added or removed at each station. Our objective is to minimize the rate of arrival of unsatisfied users who find their station empty or full. The existence of an optimal inventory level at each station is proven. It may vary over time but does not depend on the capacity of the truck which operates the repositioning. Next, we compute the relative value function of the system, together with the average cost and the optimal state. These results are used to derive a policy for station's prioritization using a one-step policy improvement method. We evaluate our policy in comparison with the optimal one and with other intuitive ones in an extended version of our model. From our numerical experiments, we show that only a little intervention of the operator can significantly enhance the quality of service, and that the rule of thumb for bike repositioning is to prioritize the closer, the more active, the closer to be full or empty, and the more imbalanced stations if no reversing in the imbalance is anticipated.

Keywords: Markov decision process; bike-sharing system; repositioning strategy; one-step policy improvement; relative value function.

1 Introduction

Bike-sharing systems (BSS) are an increasingly part of today's cities transportation. As of June 2014, this transportation system was available in 50 countries, including 712 cities, and operating approximately 806,200 bicycles at 37,500 stations (Shaheen et al., 2014). Bike-sharing systems operate as an ecological alternative/complement to existing transportation modes. It has the ability to reduce the congestion in other transportation modes. Yet, the implementation of a Bike-sharing system does not have equivalent effects on each other transportation modes. For instance, bus users are more likely to use bikes than car users (Bachand-Marleau et al., 2012; Ma et al., 2015). It is thus difficult beforehand to determine the effect of implementing a BSS; its eventual popularity and its effect on other transportation modes.

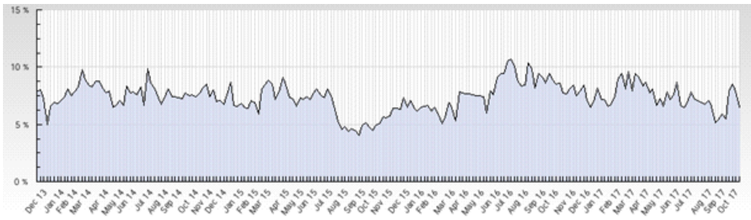
The objective for the BSS operator is to propose a reliable service where users can easily find an available bike or an available dock to return a bike. For this purpose, the design of the BSS should be well conceived. This is a long-term issue for which the sizes of the stations, their location, or the number of available bikes have to be determined. An important research literature tackles this issue (Lin and Yang, 2011; Martinez et al., 2012; Lin et al., 2013; Nair and Miller-Hooks, 2014; Frade and Ribeiro, 2015). However, even with a well-designed BSS, the risk of having full or empty stations cannot be completely avoided due to the stochasticity and the imbalance of bike arrivals and departures at a station. For instance, during peak morning hours, user flows are almost always one directional from residential areas to business areas. Another example is the flow from higher altitude stations to lower altitude stations. Increasing the size and the number of stations could enhance the BSS performance. It is however a costly solution which often cannot be considered in the short-term due to economic constraints.

One way to improve the quality of service without modifying the architecture of the BSS is to redistribute bikes among stations in order to provide a sufficient number of available bikes and empty docks at each station. For instance in Paris in 2017, the redistribution is done 24/7 by 23 trucks each with a capacity of 20 bikes and 2 buses each with a capacity of 62 bikes. The team in each truck is responsible of a given sector whereas the buses travel longer distances between large stations in front of train stations or universities. The buses follow predetermined routes which can be readjusted by the operator whereas the trucks' teams have more responsibilities in the choices made to redistribute the bikes in one sector. In both cases, the idea for the operator is to move bikes from maintenance inventory zones in the suburbs to a succession of bike stations in order to better balance the number of bikes per station.

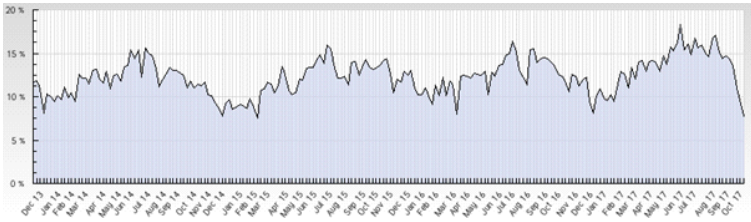
The definition of a good redistribution policy is difficult to make for the agents and the operator. One of the reasons of this difficulty is the forecast of users' future movements. The forecasts are based on historical data and on external factors like the weather or the specific users' behavior due to a particular event. Yet, the stochastic aspect of the demand leads to important gaps between the forecasts and the realizations. Another difficulty lies in the limits of the interventions of the operator. In Paris, only 3000 bikes in average are moved per day by the operator over a total of 110 000 rented bikes per day in average. Hence, the ability of the operator to redistribute bikes is limited and priority choices have to be made. For example, when the stations are spaced it is more annoying to arrive at a full station. The last difficulty is to evaluate the number of unsatisfied users. For this purpose, one need to estimate the number of renters who could not rent or return a bike at the wanted station. However, the system only keeps track of the users' realized actions, not their primary choices. Further investigations based on data analysis are needed to estimate the number of unsatisfied users.

In Figure 1, we show the evolution of the average proportion of empty and full stations per month in

Paris between December 2013 and October 2017. Although the majority of stations are neither full nor empty in average, the proportion of full and empty stations is significant. In addition, some sectors are more affected by this problem than others. In Figure 2, we show the example of an altitude station in Paris (Station 20036) for which the demand for bikes is much higher than the demand for empty docks (departure dominant station). For this station, the operator had to add bikes 10 times during a week but still the station was empty most of the time. One way to improve the quality of service would be to increase the



(a) Proportion of full stations



(b) Proportion of empty stations

Figure 1: Performance of the BSS in Paris between December 2013 and October 2017

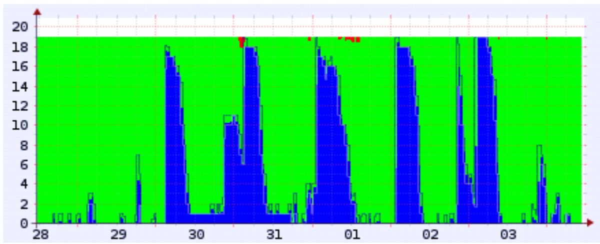


Figure 2: Evolution of the number of bikes at Station 20036 in Paris from the 28 of October to the 04 of November (green: empty docks, blue: available bikes, red: broken bikes)

number of trucks which participate in the bike redistribution. Yet, this is costly solution which may also reduce the ecological benefit of the BSS. Instead, we propose to develop an implementable decision-support tool which may help the operator to decide at any point of time (i) which station should be prioritized, and (ii) which number of bikes should be added or removed at each station. Our objective is to minimize the rate of arrival of unsatisfied users.

An optimal decision-support tool should take into account the state of each station, the distances between the truck and the stations, the priority for spaced stations, the truck capacity, the uncertainties related to the traffic, the stochasticity of the bike's demand and the bike's arrival, the demand forecasts, and the specificity of each sector of the city. The complexity of this high dimensional problem did not allow researchers to determine an optimal policy which could be implemented in any BSS. Instead, good heuristics, asymptotic regimes, or stylized models were considered to enhance the so-called "dynamic bike repositioning" strategy (see references below).

Alternatively to the existing studies, we propose to use a *Markov decision process* approach to derive a dynamic bike repositioning strategy. The value of Markov decision process approaches is that they can lead to exact optimal policies in the long-run in a stochastic context (Pandelis, 2010; Zhuang and Li, 2012; Fianu and Davis, 2018). In this sense, it seems to be the best tool to tackle the dynamic redistribution problem. Yet, in the existing studies, these approaches have been avoided. Instead, in the literature, a method is often built on restrictive assumptions (no stochasticity, no time-dependency, or deterministic times to reach a station) for a short-term objective (correct an imbalanced situation). The proposed methods are next evaluated using simulations in a more realistic context. The main reason why Markov decision process approaches have been avoided is the high dimensionality of the problem which does not allow us in general to compute the optimal policy in a reasonable time. This problem has already been described in Bellman (1961) and is referred as "the curse of dimensionality". Another reason is the time-dependency of the arrival and departure processes.

We propose in this article to tackle the limitations of the Markov decision process approach when it is applied to a BSS in order to provide a long-run repositioning policy. Let us start with the time-dependency of the processes. Borgnat et al. (2009) and Vogel (2016) have shown that the arrival and departure processes have a cyclic form. We hence propose to approximate a time cycle by a succession of intervals with random durations. On each interval, arrivals and departures are assumed to be random processes with constant parameters. The number of intervals per cycle can be controlled in order to get as close as wanted to the real cycle. Due to the cyclic effect, this approximation leads to a finite number of phases and solves the problem of the time-dependency.

Next, we consider a policy for a single bike station with regular operator's intervention. Using a value iteration step, we prove that there exists a unique (local and global) minimum for the value function irrespective of the truck capacity under this policy. This proves that for each station the intervention of the operator consists in adding or removing bikes in order to reset the station to its optimal state or to get as close as possible to this optimal state. For this policy, we derive the *relative value function* of the system together with the average cost, and the optimal state as a function of the system parameters. Closed-form expressions are obtained with constant arrival and departure parameters.

The computation of the relative value function is next used to obtain an improved policy for the prioritization of one station. Starting from a Bernoulli policy where the relative value function for each station can be derived, we apply a one-step policy improvement method to determine which station should be prioritized for repositioning when a truck has to choose between several stations. The value of the so-called Bernoulli policy is that this policy takes decisions randomly among a finite set of actions independently of the system state based on fixed probabilities (Boucherie and van Dijk, 2017). Hence, the relative value function of the single station can be used to derive the multi-station policy. This method reduces the complexity of the computation from exponential to linear.

We evaluate our policy for bike repositioning in settings where the limited number of stations allows us to make the comparison with the optimal policy. Our illustrations show that our approach yields *nearly optimal* policies in the sense that the optimal policy and our policy do not differ much in their decision actions and in their proportions of unsatisfied users. We further investigate the impact of our repositioning policy on the performance in a sector of a BSS. In particular, we show that only a little operator's intervention can significantly enhance the quality of service. This improvement is significant in the most imbalanced stations. We next identify the key drivers for prioritizing a station. These are the distances to reach a station, the intensity of the users' activity, and the future imbalance between arrivals and departures. The priority for one station increases as the state of one station gets closer to its boundaries. Finally, we extend our model definition to closer to reality settings with non-exponential times for the operator to move from one station to the other, non-independent arrival processes at each station, and a limitation in the truck capacity. Since the optimal policy is unknown in these settings, we choose to evaluate the quality of our policy in comparison with other policies which follow intuitive priority rules like shortest distance first, longest idle/full station first, and a more complex prioritization rule combining the key drivers mentioned above. The results indicate that a significant improvement can be obtained by policy enhancement for a given frequency of bike repositioning.

Structure of the article. The remainder of this paper is structured as follows. We conclude this section with a literature survey. Section 2 formulates the optimization problem and explains our assumptions. Section 3 determines the optimal policy in one station at a truck intervention. Section 4 computes the relative value function under the optimal policy. Section 5 proposes an approach to obtain an improved policy for the station prioritization. Section 6 compares our improved policy with others in a closer to reality setting. Section 7 gives concluding remarks and opens on future research directions.

Literature review. The popularity of BSS gives rise to a growing research activity. For a background on the historical development of BSS, we refer the reader to DeMaio (2009). A recent overview on BSS research can be found in Vogel (2016). One important long-run issue is the design of BSS (Lin and Yang, 2011; Martinez et al., 2012; Lin et al., 2013; Nair and Miller-Hooks, 2014). The purpose of these studies is to determine the location of the stations and their size together with the number of available bikes considering economic constraints and service level objectives. Some papers combine the problem of the BSS design with the relocation problem. For instance, Frade and Ribeiro (2015) propose a solution which integrates strategic decisions for locating bike-sharing stations, the definition of the dimension of the system, and operational decisions (relocating bicycles). A fraction of recent work has been dedicating efforts on developing good repositioning policies. In the considered studies, either the bikes are moved by the operator during the night (static repositioning) or during the day (dynamic repositioning).

Let us start with the dynamic repositioning studies. These studies are the most closely related to our article. Kek et al. (2009) use a mixed-integer program to generate redistribution plans and allocate operator staff to redistribution and maintenance activities. Their model uses a time-expanded network with a static and known demand instead of a stochastic one. Nair and Miller-Hooks (2011) develop a stochastic joint chance constraint model to study vehicle redistribution. As in our model, they assume that the demand for vehicles and empty docks at a station follows a known probability distribution. Their model aims to correct effectively short-term demand asymmetry. Considering this short-term objective, they assume that the demand for vehicles and empty docks does not vary over time. This is a difference with our model and objective. We consider a long-term objective with time-dependent probability distributions for the demands. Shu et al. (2013) identify the appropriate operational environments in which periodic redistribution of bicycles will be most effective for improving the system performance. Using a linear programming approach, they evaluate the appropriate inventory level of each station for redistribution and the appropriate number of bicycles to deploy in the network in relation with the utilization rate. In addition, they show that a good redistribution policy has the potential to reduce the number of docks needed at each station. They assume that the demand for the travel from one station to the other follows a time-dependent Poisson distribution. We use the same distribution for the demand of bikes and empty docks at a station. Compared to their article, we do not consider the information on users' future travel and we use a dynamic programming approach instead of a linear programming one.

Schuijbroek et al. (2017) propose a new cluster-first route-second heuristic, in which the polynomial-size Clustering Problem simultaneously considers the service level feasibility constraints and approximate routing costs. As in our article, they show that it is better to reset a close station even if this station is neither full nor empty. Their results are built on the transient analysis of an M/M/1/K queue to model the bike station. This approach allows them to obtain upper and lower bounds for the optimal inventory level at each station. The model assumptions of our article differ from this article. We assume that the parameters are time-dependent and that the time to reach a station is stochastic. In the context of car rentals, Ghosh et al. (2015) propose to dynamically redeploy idle vehicles using carriers so as to minimize lost demand or alternatively maximize revenue for the vehicle sharing company. This revenue maximization approach differs from the BSS where the quality of service is the objective and the redistribution represents a cost constraint.

Considering static repositioning, Raviv and Kolka (2013) propose an inventory model to study the management of bicycle stations based on the information of a single station. Also, Forma et al. (2015) develop a 3-step heuristic for the static repositioning problem. Bulhões et al. (2018) consider the static bike relocation problem with multiple vehicles and visits. They develop an integer programming formulation with an iterated local search metaheuristic that employs efficient move evaluation procedures. They include in their model the features of vehicles capacities and service time limits. Erdoğan et al. (2014) study the static bike relocation problem as a variant of the one commodity pickup and delivery traveling salesman problem. Datner et al. (2017) tackle the repositioning problem by including the dependency between the demands of different stations and the possibility of users' abandonment. Their objective is to determine optimal inventory levels for each station.

Other studies focus on performance evaluation and on alternative incentives to improve the quality of service of the BSS. Fricker and Gast (2014) use a mean-field approximation to obtain the asymptotic behavior of a BSS model as the system size becomes large in order to derive qualitative and quantitative results and further investigate on incentives and redistribution strategies. An interesting approach consists in letting users redistribute the bikes among the stations through pricing incentives (Haider et al., 2014; Fricker and Servel, 2016). This possibility is implemented in Paris where 15 free extra minutes are given if a bike is returned to an altitude station. In the context of car rentals, Kaspi et al. (2016) develop the possibility of vehicle reservation. Reservation could have a high potential Legros (2017) but is however unlikely to be implemented in a BSS.

Another way to help the operator in the redistribution policy is to develop forecast models. Borgnat et al. (2009) develop a linear regression model to predict the system-wide number of bikes hired per day and fluctuation per hour from historical data for Lyon's bike-sharing program Vélo'V. In particular, they show the existence of a cyclic effect in the bike demand. Nair et al. (2013) use a stochastic characterization of demand to study the nature of the imbalance. The objective is to better identify stations with capacity bottlenecks. Li et al. (2015) propose a hierarchical prediction model to predict the number of bikes that will be rent from/returned to each station in a future period so that reallocation can be executed in advance.

2 Problem formulation

The BSS Model. We consider a geographical area in a BSS modeled as a set of s independent stations each with capacity c_i, for $1 \leq i \leq s$. At Station i, a bike departure is due to the arrival of a user in need of a bike. We assume that the arrival process of these users is a non-homogeneous Poisson process with rate $\mu_{i,t}$, for $1 \leq i \leq s$ at a given time t, $t > 0$. A bike arrival is due to a user in need of returning a bike at a station. Again, we assume that the arrival process of these users at Station i is a non-homogeneous Poisson process with rate $\lambda_{i,t}$, for $1 \leq i \leq s$ at a given time t, $t > 0$.

The operator's interventions in the area consist in sending a truck at a given station to add or remove bikes. The operator's ability to operate an intervention in a given Station i starting from another Station

j is an important aspect of the repositioning problem. The time between two interventions is determined by the time spent at a given station, the driving time between these stations, the eventual necessity to go back to a maintenance inventory zone in the suburbs, and the intensity of the operator' activity in the intervention area. The time spent at a station is influenced by the number of bikes to add or remove and by unanticipated problems like bikes' breakdown which may force the truck to stay longer than expected at Station i to do on-site repairs. The driving time depends on the distance between the stations and on the traffic. The necessity to go back to the maintenance zone is determined by the working periods of the agents and by the need to add or remove bikes from the truck. The intensity of the operator' activity is conditioned by the limited fleet of trucks to operate repositioning in the intervention area. Hence, even if an intervention is desirable at Station j, the operator may decide to postpone the intervention and send the truck to a more urgent intervention area. The time to reset Station j starting from Station i is hence very complex to estimate. For the tractability of the analysis, we assume that this duration is exponentially distributed with rate $\gamma_{i,j}$. We further assume that the truck has a sufficiently large capacity which does not alter its ability to modify a station state.

Objective. The objective for the system operator is to minimize the long-run overall rate of arrival of unsatisfied users. This can be formulated as minimizing the following weighted cost summation,

$$\min \left(\lim_{t \to \infty} \left(\sum_{i=1}^{s} r_{1,i} \cdot T_{1,i,t} + r_{2,i} \cdot T_{2,i,t} \right) \right), \tag{1}$$

where $T_{1,i,t}$ and $T_{2,i,t}$ are the expected rate of arrival of users who cannot find an available bike and who cannot find an empty dock at Station i up to time t, respectively ($1 \leq i \leq s$, $t > 0$). A cost of $r_{1,i}$ per unsatisfied user due to the absence of bikes and a cost of $r_{2,i}$ per unsatisfied user due to the absence of empty docks are counted at Station i. These cost parameters represent the importance, given by the operator, to unsatisfied users. In case $r_{1,i} = r_{2,i}$, the operator does not consider the reason of the dissatisfaction at Station i. If $r_{2,i} > r_{1,i}$, a higher importance is given to the absence of empty docks. The absence of empty docks forces the user to cycle to another station. The time spent on the bike may then exceed the due time for returning the bike. This can lead to higher renting costs and may affect more the user than finding an empty station. Yet, finding an empty station may also result in unwanted costs for the user if an alternative transportation has to be chosen (car, subway, ...). The case $r_{1,i} > r_{2,i}$ may then also be justified. The cost parameters can differ from one station to the other. As mentioned in the introduction, if stations are spaced, then it is more urgent to reset them. This can be translated by high values for $r_{1,i}$ and $r_{2,i}$ in such stations.

Actions. In order to answer the optimization problem, regular interventions of the operator are decided. These interventions consist in sending a truck to a station in order to add or to remove a given number of bikes. As explained in the introduction for the BSS of Paris, we assume that a truck is responsible of a given sector of the city. Hence, two trucks do not compete to reset a station. This assumption is also consistent with the limited resources of the operator encountered in most BSS. We further assume that the next station to reset is chosen just at the end of an intervention in another station. Therefore, we do not consider a predefined route. Instead, we let the decision be function of the system state and the time evolution of the system parameters at a given time. This flexibility is implemented for small trucks in Paris. Even for large buses, the predefined route is often readjusted during the day function of the system state. In summary, after an intervention in a station, the operator has to choose (i) which station should be prioritized to be the next one to be rebalanced and (ii) for the chosen station, what should be its optimal inventory level.

The Markov decision process approach. We propose to formulate the routing problem as a Markov decision process (MDP). The first difficulty is related to the time-dependency of the parameters. If $\lambda_{i,t}$ and $\mu_{i,t}$ were completely unknown function of time, then taking a rational decision at a given time t would be an impossible task. Fortunately, statistical studies have shown that $\lambda_{i,t}$ and $\mu_{i,t}$ have a cyclic form (Borgnat et al., 2009; Vogel, 2016). The duration of the cycle during the week days is of 24 hours. Yet, irregularities can be observed during the week-ends. We can then also choose cycle length of a week to better capture the arrival parameters.

We propose to approximate the deterministic cycle length, τ, by an Erlang random variable with N phases and rate θ per phase. We choose N and θ such that $\frac{N}{\theta} \triangleq \tau$. This ensures that as N and θ go to infinity, the considered Erlang random variable converges in distribution to the deterministic cycle length, τ. This approximation allows us to obtain a finite state space defined by the line vector $\overline{x} = (x_1, x_2, \cdots, x_s, j, y)$, where y is the cycle phase of the process, x_i is the number of available bikes at Station i, and j is the station at which the truck is positioned at phase y, with $0 \leq x_i \leq c_i$, $1 \leq i, j \leq s$, and $1 \leq y \leq N$. For a given cycle phase y ($1 \leq y \leq N$), the arrival parameters are then approximated by constant parameters, $\lambda_{i,y}$, and $\mu_{i,y}$ chosen such that $\frac{y}{\theta} = t$. Hence, as y and θ tend to infinity $\lambda_{i,y}$ and $\mu_{i,y}$ tend to $\lambda_{i,t}$ and $\mu_{i,t}$. The parameter θ should be carefully chosen; it should be sufficiently high to well approximate the continuous time model. However, too high values of θ may induce computational difficulties due to the high dimensionality of the state space.

We choose to discretize our continuous-time model. This is possible because it is uniformizable (Section 11.5.2. in Puterman (1994)) since the event rates out of each state are bounded by $\theta + \max_{1 \leq y \leq N} \left(\sum_{i=1}^{s} (\lambda_{i,y} + \mu_{i,y}) \right) + \max_{1 \leq i,j \leq s} (\gamma_{i,j})$ that we assume equal to 1 for simplicity. We define the dynamic programming value function $V_n(\overline{x})$ over $n \geq 0$ steps, depending on the state of the system as follows. At a bike arrival (respectively, at a bike departure), the number of bikes at Station i is incurred by 1 (respectively, reduced by 1) if there is at least one available dock (respectively, one available bike) at the station. Otherwise, a cost of $r_{2,i}$ (respectively, $r_{1,i}$) is counted and the state of the station is not modified. After a θ-transition, the age of the cycle is incurred by 1 if the state of the cycle is strictly below N. Otherwise, a new cycle starts at Phase 1. After a truck intervention, the operator has to decide how many bikes should be added or removed in the best chosen station. For a given station, the minimizing action is chosen between all possible changes of the state of the station. For this operation, one could limit the number of possible changes due to the capacity limitations of the truck. This possibility is not considered here but it will be later investigated as an extension of the model. Next, the prioritized station is the one which minimizes the value function. Finally, a last transition from a state to itself is artificially added in order to have the sum of the probabilities out of each state equal to one.

We introduce the line vector $\overline{e_i} = (e_{i,1}, e_{i,2}, \cdots, e_{i,s}, e_{i,s+1}, e_{i,s+2})$, such that $e_{i,i} = 1$, and $e_{i,k} = 0$ if $i \neq k$, for $1 \leq i, k \leq s+2$. In summary, we express $V_{n+1}(\overline{x})$ in $V_n(\overline{x})$ in the following way:

$$
\begin{aligned}
V_{n+1}(\overline{x}) = & \sum_{i=1}^{s} \lambda_{i,y} \cdot \left(\mathbb{1}_{x_i<c_i} V_n(\overline{x}+\overline{e_i}) + \mathbb{1}_{x_i=c_i}(V_n(\overline{x}) + r_{2,i})\right) \\
& + \sum_{i=1}^{s} \mu_{i,y} \cdot \left(\mathbb{1}_{x_i>0} V_n(\overline{x}-\overline{e_i}) + \mathbb{1}_{x_i=0}(V_n(\overline{x}) + r_{1,i})\right) \\
& + \theta \cdot \left(\mathbb{1}_{1\leq y<N} V_n(\overline{x}+\overline{e_{s+2}}) + \mathbb{1}_{y=N} V_n(\overline{x}-(N-1)\cdot\overline{e_{s+2}})\right) \\
& + \gamma_{j, \arg\min\limits_{1\leq i\leq s}\left(\min\limits_{0\leq m\leq c_i}(V_n(\overline{x}+(m-x_i)\overline{e_i}+(i-j)\overline{e_{s+1}}))\right)} \cdot \min_{1\leq i\leq s}\left(\min_{0\leq m\leq c_i}(V_n(\overline{x}+(m-x_i)\overline{e_i}+(i-j)\overline{e_{s+1}}))\right) \\
& + \left(1 - \sum_{i=1}^{s}\lambda_{i,y} - \sum_{i=1}^{s}\mu_{i,y} - \theta - \gamma_{j, \arg\min\limits_{1\leq i\leq s}\left(\min\limits_{0\leq m\leq c_i}(V_n(\overline{x}+(m-x_i)\overline{e_i}+(i-j)\overline{e_{s+1}}))\right)}\right) V_n(\overline{x}),
\end{aligned}
\tag{2}
$$

for $n \geq 0$, where $\mathbb{1}_{(z\in A)}$ is the indicator function of a given subset A and we choose $V_0(\overline{x}) = 0$. The first minimizing operator in line of Equation (2) determines the optimal state of a given Station i. It is followed by a second minimizing operator which identifies which station should be reset in priority at its optimal state. Ideally, one way of obtaining the long-run average optimal actions is to use the value iteration technique introduced by Bellman (1957) and Howard (1960), by recursively evaluating V_n using Equation (2), for $n \geq 0$. However, the dimension of the state space prevents to do so; as the number of stations increases the dimensionality of the system state explodes.

The policy improvement approach. We instead propose the following approach to obtain a policy for the bike repositioning problem:

- **Policy π_γ.** We assume that the operator visits regularly a given station irrespective of the system state and the time evolution of the system parameters. The time between two visits of the operator is assumed to be exponentially distributed with constant rate γ. In this context, the parameter γ represents the ability of the operator to reset the station. We denote this policy by π_γ.

- **Policy π^*_γ.** Under Policy π_γ, we prove in Section 3 that there exists a time-dependent optimal inventory level for each station which depends on the arrival parameters and the frequency of visit of the station given by the parameter γ. This allows us to optimize Policy π_γ. The optimized Policy π_γ is denoted by π^*_γ. In Section 4, we derive the optimal inventory in one station, the relative value function, and the average optimal cost under Policy π^*_γ.

- **Policy** $\pi^*_{p_1,p_2,\cdots,p_s}$**.** In Section 5, we consider the general multistation setting. First, we consider a policy for which the operator returns to Station j after each intervention. The operator decides which station should be the next one to be reset from this station. This allows us to only consider the parameters which represent the time to reset Station i from Station j; $\gamma_{j,i}$. The other parameters $\gamma k, i$ do not need to be known at this step. We propose a Bernoulli policy, such that Station i is chosen from Station j with probability p_i for $1 \leq i \leq s$, with $p_s = 1 - p_1 - p_2 - \cdots - p_{s-1}$. We denote this policy by $\pi_{p_1,p_2,\cdots,p_s}$. This policy decouples the stations such that Station i behaves as a single independent station with transition rates $\lambda_{i,y}$, $\mu_{i,y}$ and $p_i \cdot \gamma_{j,i}$ for $1 \leq i \leq s$, and $1 \leq y \leq N$. Hence, using the result of Section 4, the value function, the optimal state, and the optimal average cost can be computed at each station. The average cost and the value function for the whole system is given by the sum of the individual average costs and the sum of the individual value functions of the different stations. The parameters p_1, p_2, $\cdots$, and p_s are chosen in order to minimize the average cost of the system. This leads to the optimal Bernoulli policy, denoted by $\pi^*_{p_1,p_2,\cdots,p_s}$. At this step the optimal inventory level at each station is determined.

- **Policy** π^*_I**.** Next, we improve this optimal Bernoulli policy from the minimizing action between sending the truck to Station 1, Station 2, ... or Station s from Station j using the value function computed in Section 4. This improved policy is denoted by π_{I_j}. The set of policies $\{\pi_{I_1}, \pi_{I_2}, \cdots, \pi_{I_s}\}$ determines the improved policy for the dynamic bike repositioning problem, Policy π^*_I. In practice, the optimal policy is recomputed after each truck intervention. Hence, it suffices to derive policy π_{I_j} to know which station should be reset at which state at the next intervention.

Our approach for policy improvement is depicted in Figure 3.

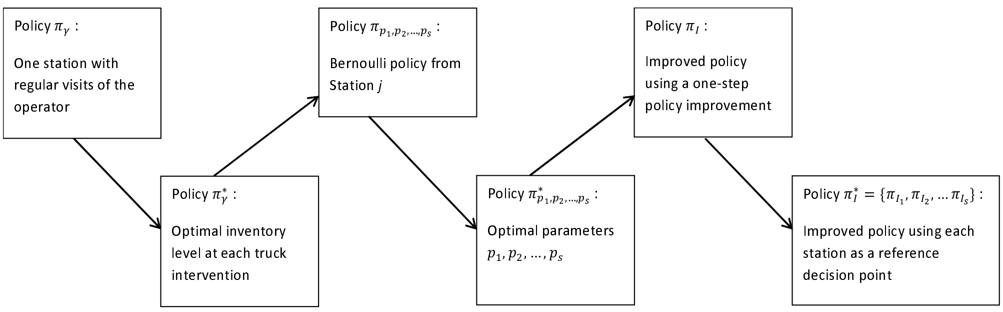


Figure 3: Approach to derive the dynamic repositioning policy π^*_I

3 Optimal intervention at a bike station

In this section, we want to determine the optimal policy at each truck arrival for a single station under Policy π_γ. More precisely, based on the station state, we want to optimize the number of bikes to add or to remove from the station. The main result of this section is that there exists an optimal station state that the operator should reach at each truck intervention. This result defines Policy π^*_γ. As an extension, we show that the optimal state in a station is independent of the truck size.

Optimal policy at each truck intervention We rewrite the value function defined in Equation (2) for a single station with $s = 1$ and $\gamma_{1,1} = \gamma$. Since only one station is considered here, we simplify the notations and denote by c, the station capacity, by λ_y and μ_y, the arrival parameters at phase y, $1 \leq y \leq N$, and by r_1 and r_2, the costs per unsatisfied users who cannot find a bike or who cannot find an empty dock. The state space reduces to the couple (x, y), where x is the number of bikes at the station and y is the phase for $0 \leq x \leq c$, and $1 \leq y \leq N$. Equation (2) can be rewritten as

$$V_{n+1}(x,y) = \lambda_y \left(\mathbb{1}_{x<c} V_n(x+1,y) + \mathbb{1}_{x=c}(V_n(x,y)+r_2)\right) + \mu_y \left(\mathbb{1}_{x>0} V_n(x-1,y) + \mathbb{1}_{x=0}(V_n(x,y)+r_1)\right) \tag{3}$$

$$+ \theta \left(\mathbb{1}_{y<N} V_n(x,y+1) + \mathbb{1}_{y=N} V_n(x,1)\right) + \gamma \min(V_n(0,y), V_n(1,y), \cdots, V_n(c,y))$$
$$+ (1 - \lambda_y - \mu_y - \theta - \gamma) V_n(x,y),$$

for $n \geq 0$, with $V_0(x) = 0$.

In Theorem 1, we prove by induction on n that the value function is convex in x for each n. The convexity property of the value function can be expressed by the relation $V_n(x+2,y) - V_n(x+1,y) \geq V_n(x+1,y) - V_n(x,y)$, for $0 \leq x \leq c-2$ and $1 \leq y \leq N$. Hence, if for a given $x \in [0, c-2]$ we have $V_n(x+1,y) - V_n(x,y) \geq 0$, then $V_n(x+2,y) - V_n(x+1,y) \geq 0$. This means that if it is optimal to remove a bike in state $(x+1,y)$, then it is also optimal to remove a bike in state $(x+2,y)$. After removing a bike in state $(x+2,y)$, the system gets into state $(x+1,y)$ where it is also optimal to remove a bike. So, it is optimal to remove two bikes at least in state $(x+2,y)$ and by extension, it is optimal to remove at least k bikes in state $(x+k,y)$, where $0 \leq k \leq c-x$ if $V_n(x+1,y) - V_n(x,y) \geq 0$. In the symmetric case, if for a given $x \in [0, c-2]$ we have $V_n(x+2,y) - V_n(x+1,y) \leq 0$ then $V_n(x+1,y) - V_n(x,y) \leq 0$. This means that if it is optimal to add a bike in state $(x+1,y)$, then it is also optimal to add a bike in state (x,y). With the same explanation as above, it is optimal to add at least $k+2$ bikes in state $(x-k,y)$, where $0 \leq k \leq x$ if $V_n(x+2,y) - V_n(x+1,y) \leq 0$.

This proves that for each n and each phase y, there exists a unique state $x^*_{n,y}$ such that the optimal action is to remove $x - x^*_{n,y}$ bikes if $x \geq x^*_{n,y}$ and the optimal action is to add $x^*_{n,y} - x$ bikes if $x \leq x^*_{n,y}$.

As n tends to infinity, the optimal policy converges to the unique average optimal policy. The convergence is due to the aperiodic irreducible finite-state Markov chains considered here. Then, Theorem 8.5.3 part c of Puterman (1994) guarantees the existence of an optimal deterministic stationary policy. The proof of Theorem 1 is given in Section 1 of the online supplement.

Theorem 1 *In the long-run, for a given phase y ($1 \leq y \leq N$), there exists an optimal state x_y^* ($0 \leq x_y^* \leq c$) such that:*

- *If $x \leq x_y^*$, the optimal action is to add $x_y^* - x$ bikes to the station.*
- *If $x \geq x_y^*$, the optimal action is to remove $x - x_y^*$ bikes from the station.*

This minimum is the unique local minimum of the value function.

Effect of the truck size. We assume so far that the operator has the ability to change the state of the station as wanted at each truck arrival. This may not be possible in practice if the truck doesn't have a sufficient quantity of bikes to add at the station or if it does not have enough capacity to remove a sufficient quantity of bikes. We prove in this section that the truck limitation does not impact the optimal state of a station. If the truck does not have the ability to change the station state as wanted at each truck arrival due to a capacity limitation then the minimizing operator in Equation (3) should be redefined. Assume that the truck can at most remove k_1 bikes and at most add k_2 bikes at each visit of the station. The minimizing operator is then redefined as $U_n(x,y) = \min_{z \in S_x} V_n(z,y)$, where $S_x = [\max(x-k_1,0), \min(x+k_2,c)]$. In Proposition 1, we prove that U_n has the same monotonicity properties as V_n. The details of the proof are given in Section 2 of the online supplement.

Proposition 1 *If $V_n \in \mathcal{F}$, then $U_n \in \mathcal{F}$.*

The consequence of Proposition 1 is that the optimal policy for the operator at each truck intervention is to get as close as possible to the optimal state. This result is important in the sense that the truck limitation may negatively affect the efficiency of the operator's intervention but it does not affect its policy.

4 Relative value function under Policy π_γ^*

We now derive the relative value function together with the optimal state and the long-run average costs under Policy π_γ^*. In the case where the rates are constant, these elements can be obtained explicitly (Section 4.1). Otherwise, a numerical method is proposed (Section 4.2).

4.1 Explicit results with constant rates

We denote by λ and μ the constant arrival and departure rates. Under the optimal policy defined in Theorem 1, the dynamic programming optimality equations which define the relative value function $V(x)$ are given by

$$(\lambda+\mu+\gamma)V(0)+g=\lambda V(1)+\mu V(0)+\gamma V(x^*)+\mu r_1,$$
$$(\lambda+\mu+\gamma)V(x)+g=\lambda V(x+1)+\mu V(x-1)+\gamma V(x^*),\ \text{for } 0<x<c,$$
$$(\lambda+\mu+\gamma)V(c)+g=\lambda V(c)+\mu V(c-1)+\gamma V(x^*)+\lambda r_2.$$

The state x^* is the optimal state for the bike station. This state minimizes the value function. We choose to define this state as the reference state and assume $V(x^*)=0$. We denote by g the average long-run cost of the system.

Remark. As in the previous section we could have divided the optimality equations by $\lambda+\mu+\gamma$ and assume that $\lambda+\mu+\gamma=1$. We choose to keep $\lambda+\mu+\gamma$ in the equations in order to better explain the impact of the system parameters in Section 4.3.

In Theorem 2, we give the closed-form expressions of the relative value function with the related average cost and the optimal state of the system. The idea of the proof is to introduce the difference $\Delta(x)=V(x+1)-V(x)$, for $0\le x\le c-1$. Subtracting the expression of $V(x)$ to the expression of $V(x+1)$ using the optimality equations allows us to derive a linear relation for $\Delta(x)$. Using this relation, we obtain an expression of $\Delta(x)$ as a function of g. Finally, from the boundary optimality equations for $x=0$ and $x=c$, we obtain the unique expression of g which in turn leads to the expression $V(x)$ via $V(x)=\sum_{k\le x-1}\Delta(k)$. Note that due to linearity, the value function could be decomposed into $V(x)=V_1(x)+V_2(x)$, which are due to the costs per unsatisfied user either due to the absence of bike or the absence of empty dock. In the same way we can decompose the average cost into $g=r_1T_1+r_2T_2$. The proof is given in Section 3 of the online supplement.

Theorem 2 *We have for* $0\le x\le c$,

$$V(x)=\frac{(\alpha_1^{x^*}-\alpha_1^x)(\alpha_1 r_2+\frac{\mu}{\lambda}r_1\alpha_2^c)}{(1-\alpha_1)(\alpha_1^{c+1}-\alpha_2^{c+1})}-\frac{(\alpha_2^{x^*}-\alpha_2^x)(\alpha_2 r_2+\frac{\mu}{\lambda}r_1\alpha_1^c)}{(1-\alpha_2)(\alpha_1^{c+1}-\alpha_2^{c+1})},$$

Moreover $g=r_1T_1+r_2T_2$, *where*

$$T_1=\mu\cdot\frac{\alpha_1^c\alpha_2^{x^*}(\alpha_1-1)-\alpha_2^c\alpha_1^{x^*}(\alpha_2-1)}{\alpha_1^{c+1}-\alpha_2^{c+1}},\ \textit{and}\ T_2=\lambda\cdot\frac{\alpha_1^{x^*}(\alpha_1-\mu/\lambda)-\alpha_2^{x^*}(\alpha_2-\mu/\lambda)}{\alpha_1^{c+1}-\alpha_2^{c+1}},$$

with

$$x^*=\frac{\ln\left(\frac{\ln(\alpha_2)(1-\alpha_1)(r_2\alpha_2+\frac{\mu}{\lambda}r_1\alpha_1^c)}{\ln(\alpha_1)(1-\alpha_2)(r_2\alpha_1+\frac{\mu}{\lambda}r_1\alpha_2^c)}\right)}{\ln(\alpha_1/\alpha_2)},$$

$\alpha_1=\frac{\lambda+\mu+\gamma+\sqrt{(\lambda+\mu+\gamma)^2-4\lambda\mu}}{2\lambda}$ *and* $\alpha_2=\frac{\lambda+\mu+\gamma-\sqrt{(\lambda+\mu+\gamma)^2-4\lambda\mu}}{2\lambda}$.

Remarks.

- The derived value for x^* using the closed form expression in Theorem 2 is a real. Since $V(x)$ is convex in x, the value which minimizes $V(x)$ is either $\lfloor x^* \rfloor$ or $\lceil x^* \rceil$, where the notations $\lfloor x \rfloor$ and $\lceil x \rceil$ mean the first integer below and the first integer above a given real x.

- One way to get closer to the real threshold x^* is to randomize between $\lfloor x^* \rfloor$ or $\lceil x^* \rceil$ (i.e., the operator chooses $\lfloor x^* \rfloor$ with a given probability or $\lceil x^* \rceil$ otherwise). This may however be difficult to implement in practice.

- In extreme imbalanced cases between the arrival of bikes and the arrival of users, one can find situations where the computed value of x^* is either below 0 or above c. In these cases, the value which minimizes $V(x)$ is either 0 or c.

In Figure 4, we present the evolution of the value which minimizes $V(x)$ (integer threshold) and the one of the real threshold as a function of λ for a fixed value for γ. As expected, as the arrival of bikes increases compared to the demand for bikes, the optimal number of bikes at the station decreases.

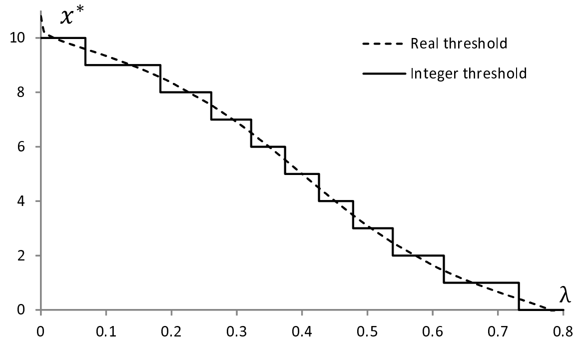


Figure 4: Obtained optimal state as a function of λ ($c = 10$, $\gamma = 0.2$, $\mu = 0.8 - \lambda$, $r_1 = r_2 = 0.1$)

4.2 Numerical analysis with non-constant rates

With non-constant rates, under the optimal policy defined in Theorem 1, the dynamic programming optimality equations which define the relative value function $V(x, y)$ are given by

$$(\lambda_y + \mu_y + \gamma + \theta)V(0, y) + g = \lambda_y V(1, y) + \mu_y V(0, y) + \gamma V(x_y^*, y) \tag{4}$$
$$+ \theta(\mathbb{1}_{y<N} V(0, y + 1) + \mathbb{1}_{y=N} V(0, 1)) + \mu_y r_1,$$
$$(\lambda_y + \mu_y + \gamma + \theta)V(x, y) + g = \lambda_y V(x + 1, y) + \mu_y V(x - 1, y) + \gamma V(x_y^*, y) \tag{5}$$
$$+ \theta(\mathbb{1}_{y<N} V(x, y + 1) + \mathbb{1}_{y=N} V(x, 1)), \text{ for } 0 < x < c,$$
$$(\lambda_y + \mu_y + \gamma + \theta)V(c, y) + g = \lambda_y V(c, y) + \mu_y V(c - 1, y) + \gamma V(x_y^*, y) \tag{6}$$
$$+ \theta(\mathbb{1}_{y<N} V(c, y + 1) + \mathbb{1}_{y=N} V(c, 1)) + \lambda_y r_2,$$

for $1 \leq y \leq N$. This corresponds to a finite set of equations. Unfortunately, $V(x, y)$ cannot be obtained explicitly. Instead, we propose a numerical method to derive $V(x, y)$ for $0 \leq x \leq c$, and $1 \leq y \leq N$. The approach is similar to the one developed in Section 4.1. Yet, the 2-dimensional problem considered here involves matrix and does not lead to explicit expressions.

As in Section 4.1, we introduce the difference $\Delta(x, y) = V(x + 1, y) - V(x, y)$ for $0 \leq x \leq c - 1$ and $1 \leq y \leq N$. By subtracting $V(x, y)$ to $V(x + 1, y)$ using the optimality equations, we get

$$(\lambda_y + \mu_y + \gamma + \theta)\Delta(0, y) = \lambda_y \Delta(1, y) + \theta(\mathbb{1}_{y<N}\Delta(0, y + 1) + \mathbb{1}_{y=N}\Delta(0, 1)) - \mu_y r_1, \tag{7}$$
$$(\lambda_y + \mu_y + \gamma + \theta)\Delta(x, y) = \lambda_y \Delta(x + 1, y) + \mu_y \Delta(x - 1, y) + \theta(\mathbb{1}_{y<N}\Delta(x, y + 1) + \mathbb{1}_{y=N}\Delta(x, 1)), \tag{8}$$
$$\text{for } 0 < x < c - 1,$$
$$(\lambda_y + \mu_y + \gamma + \theta)\Delta(c - 1, y) = \mu_y \Delta(c - 2) + \theta(\mathbb{1}_{y<N}\Delta(c - 1, y + 1) + \mathbb{1}_{y=N}\Delta(c - 1, 1)) + \lambda_y r_2. \tag{9}$$

Let us denote by U_x the column vector of dimension N, $(\Delta(x, 1), \Delta(x, 2), \cdots, \Delta(x, N))$, for $0 \leq x \leq c - 1$. Assuming that $\lambda_y > 0$ for $1 \leq y \leq N$, Equations (7) and (8) can be rewritten as

$$U_1 = A \cdot U_0 + r_1 B \cdot \mathbf{1}, \text{and,} \tag{10}$$
$$U_{x+1} = A \cdot U_x - B \cdot U_{x-1}, \tag{11}$$

for $0 < x < c-1$, where $\mathbf{1}$ is the column vector of dimension N with all coefficients equal to 1,

$$A = \begin{pmatrix} \frac{\lambda_1+\mu_1+\gamma+\theta}{\lambda_1} & -\frac{\theta}{\lambda_1} & 0 & \cdots & 0 \\ 0 & \frac{\lambda_2+\mu_2+\gamma+\theta}{\lambda_2} & -\frac{\theta}{\lambda_2} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\lambda_{N-1}+\mu_{N-1}+\gamma+\theta}{\lambda_{N-1}} & -\frac{\theta}{\lambda_{N-1}} \\ -\frac{\theta}{\lambda_N} & 0 & \cdots & 0 & \frac{\lambda_N+\mu_N+\gamma+\theta}{\lambda_N} \end{pmatrix} \text{ and, } B = \begin{pmatrix} \frac{\mu_1}{\lambda_1} & 0 & \cdots & 0 \\ 0 & \frac{\mu_2}{\lambda_2} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \frac{\mu_N}{\lambda_N} \end{pmatrix}.$$

The difficulty to derive an explicit expression of U_x as a function of U_0 is that the matrix A and B do not commute. On way to avoid this difficulty is to define the matrix C of dimension $2N \times 2N$ by

$$C = \begin{pmatrix} A & -B \\ I & O \end{pmatrix},$$

where I is the identify matrix of dimension $N \times N$ and O is the matrix of dimension $N \times N$ with all coefficients equal to 0. Hence, Equation (11) can be rewritten as $\begin{pmatrix} U_{x+1} \\ U_x \end{pmatrix} = C \cdot \begin{pmatrix} U_x \\ U_{x-1} \end{pmatrix}$, for $0 < x < c-1$. Combining this relation with equation (10) leads to

$$\begin{pmatrix} U_{x+1} \\ U_x \end{pmatrix} = C^x \cdot \begin{pmatrix} A \cdot U_0 + r_1 B \cdot \mathbf{1} \\ U_0 \end{pmatrix},$$

for $0 \leq x < c-1$. Equation (9) can be rewritten as $r_2\mathbf{1} = AU_{c-1} - BU_{c-2}$. Hence, Equation (11) can be extended to $x = c-1$ by defining $U_c = r_2\mathbf{1}$. Thus, we may write

$$\begin{pmatrix} r_2\mathbf{1} \\ U_{c-1} \end{pmatrix} = C^{c-1} \cdot \begin{pmatrix} A \cdot U_0 + r_1 B \cdot \mathbf{1} \\ U_0 \end{pmatrix}.$$

Let us write

$$C^x = \begin{pmatrix} C_{1,x} & C_{2,x} \\ C_{3,x} & C_{4,x} \end{pmatrix},$$

for $x \in \mathbb{R}$ where $C_{i,x}$ is a matrix of dimension $N \times N$. We hence obtain $U_0 = C_{1,c}^{-1} \cdot [r_2 I - r_1 C_{1,c-1} B] \cdot \mathbf{1}$, and $U_x = C_{3,x+1} \cdot U_0 + r_1 C_{3,x} B \cdot \mathbf{1}$, for $0 \leq x \leq c-1$. The vector U_x can then be computed.

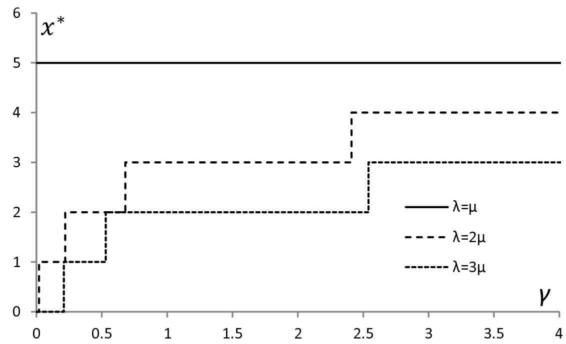
We now derive $V(x,y)$. We denote by W_x the column vector $(V(x,1), V(x,2), \cdots, V(x,N))$ for $0 \leq x \leq c$ and by X^* the column vector $(V(x_1^*,1), V(x_2^*,2), \cdots, V(x_N^*,N))$. We have $W_1 = W_0 + U_0$. From Equation (4), we get $W_0 = (-U_0 + g \cdot I + \gamma X^*) \cdot (I - A + B)^{-1}$. Next, with $W_x = W_0 + U_0 + U_1 + \cdots + U_{x-1}$, one can express W_x as a function of U_0, g and X^*. Hence, for $1 \leq y \leq N$, $V(x,y)$ can be expressed as a function of $\Delta(0,y)$, g, and $V(x_y^*,y)$. The cost g is obtained using $V(x_y^*,y) = 0$ for the value of y which minimizes $V(x_y^*,y)$ (reference state). Next, by replacing g in the expression of $V(x,y)$, we get the numerical value of $V(x,y)$.

4.3 Impact of the repositioning frequency in one station

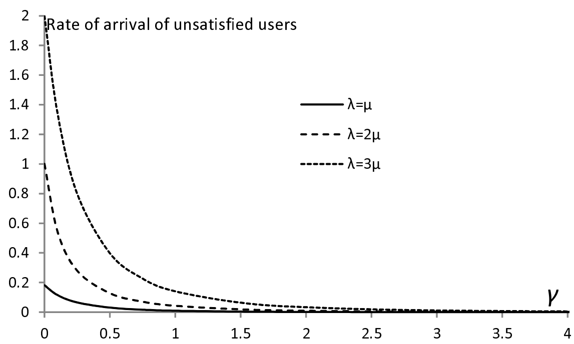
We investigate the impact of the repositioning frequency in one station in a context with constant arrival and departure rates. Although Policy π_γ^* is not optimal, the frequency of visit of a station can be controlled in this policy and hence the effect of this frequency can be measured. The objective is to gain insights on how much the repositioning strategy should be involved to sufficiently improve the user's satisfaction. Another interesting aspect is the expected number of bikes in a station. In Corollary 1, we give the expression of the expected number of bikes at the station, $E(N)$. The proof follows from the expressions of T_1 and T_2 given in Theorem 2. The details are given in Section 4 of the online supplement.

Corollary 1 *We have* $E(N) = x^* + \frac{\lambda}{\gamma} \cdot \frac{\alpha_1^{x^*}(\alpha_1-\mu/\lambda)-\alpha_2^{x^*}(\alpha_2-\mu/\lambda)}{\alpha_1^{c+1}-\alpha_2^{c+1}} - \frac{\mu}{\gamma} \cdot \frac{\alpha_1^{c}\alpha_2^{x^*}(\alpha_1-1)-\alpha_2^{c}\alpha_1^{x^*}(\alpha_2-1)}{\alpha_1^{c+1}-\alpha_2^{c+1}}$.

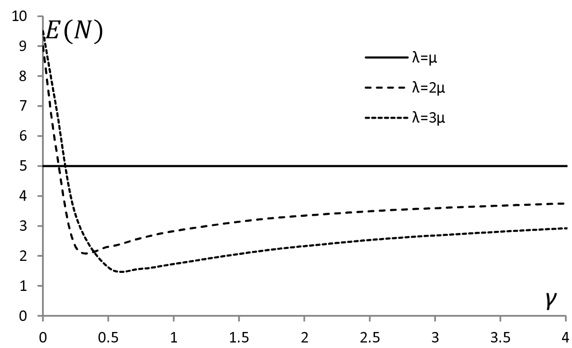
In Figure 5, we illustrate the optimal state, the related rate of arrival of unsatisfied users and the expected number of bikes at the station as a function of γ. As the imbalance between λ and μ increases, x^* decreases



(a) Optimal state x^*



(b) Rate of arrival of unsatisfied users



(c) $E(N)$

Figure 5: Impact of γ ($c = 10$, $\mu = 1$, $r_1 = r_2 = 1$)

(it would instead increase if $\mu > \lambda$). This result is intuitive; the optimal state x^* is chosen to reduce the risk of observing the system in state $x = 0$ or in state $x = c$. Moreover, as γ increases, x^* and $E(N)$ increase and converge to $c/2$. For high values of γ the system is able to quickly correct a dangerous situation for future users. Choosing a low value for x^* would lead to a risk of observing an unintended succession of bike demand which could lead to $x = 0$. It is therefore better with a high corrective capacity to choose to have a half-full station. On the contrary, if the operator does not have the ability to often reset a station, its action consists in compensating the natural imbalance of the arrival and demand by adding or removing a high number of bikes. By extension, in a multi-station setting, it means that distant stations from the truck location should be repositioned in an extreme state which compensates the imbalance whereas close stations should be repositioned in a state far from its boundaries.

Finally, as γ increases, the proportion of unsatisfied users decreases. This result is straightforward but it is interesting to observe the convexity of the curves in Figure 5(b). The system performance is very sensitive to the parameter γ for low values of γ. For instance in the case $\lambda = \mu = 1$, with only $\gamma = 0.1$ we obtain a reduction of the rate of arrival of unsatisfied users of 33%. This means that if the operator resets the station every 20 users' movements (arrivals or departure), the rate of arrival of unsatisfied users can be significantly reduced. This also means that the operator should adjust the number of visit in one station to the intensity of users' movements. Our observation meets the one of Shu et al. (2013) who show that the effectiveness of the redistribution strategy is strongly impacted by the parameters of bike arrivals and departures. The significant observed improvements mean that there is a strong incentive to efficiently develop a repositioning strategy. In Section 5 of the online supplement, we study the relation between the repositioning frequency and the truck size. The conclusion of this analysis is that the limitation of the truck capacity has a strong impact when the frequency of truck visit is high.

5 How to choose the next Station?

We now explain how to derive Policy π_I^*. In Section 5.1, we present the algorithm to obtain Policy π_I^*. In Section 5.2, we evaluate this policy in comparison with the optimal one.

5.1 Algorithm to determine Policy π_I^*

As explained in Section 2, Equation (2) cannot be used to obtain the optimal policy due to the high dimension of the state space. Therefore, we propose an approximation method to determine the station to prioritize. We propose to use the one-step policy improvement introduced by Norman (1972) and Ott and Krishnan (1992) and further developed by Hwang et al. (2000) and Bhulai and Koole (2003). The one-step policy improvement works for models for which the value function of a certain policy can be obtained. One-step policy improvement consists of doing the improvement step on the basis of this initial policy. The motivation for considering this method is that policy improvement method gives the biggest improvement during the first step.

We assume that the truck has just finished to reset Station j ($1 \leq j \leq s$). As a starting point, we use the so-called Bernoulli policy with the optimal parameters p_1, p_2, $\cdots$, p_{s-1}, and $p_s = 1 - p_1 - p_2 - \cdots p_{s-1}$ such that the average system cost is minimized. This policy is derived using the results of Section 4 by replacing the parameter γ by $p_i \cdot \gamma_{j,i}$ for Station i ($1 \leq i \leq s$). This determines Policy $\pi^*_{p_1,p_2,\cdots,p_s}$ with the optimal inventory levels at each station. More precisely, to determine the optimal Bernoulli policy, we need to find p_1, p_2, $\cdots$, and p_s such that $g(p_1, p_2, \cdots , p_s) = g_1(p_1) + g_2(p_2) + \cdots + g_s(p_s)$ is minimized under the constraints $p_1 + p_2 + \cdots + p_s = 1$, and $0 \leq p_i \leq 1$, for $1 \leq i \leq s$, where $g_i(p_i)$ is the average cost at Station i obtained from the results of Section 4. Given that the function g is decreasing and convex in the variables $p_1, p_2, \cdots , p_s$ and that the constraints of the problem define a close bounded subset of $[0, 1]^s$, we propose an approximation algorithm based on the strongest slope principle controlled by a parameter ϵ (see Algorithm 1 below).

Algorithm 1: Policy $\pi^*_{p_1,p_2,\cdots,p_s}$.

1. Initialization. Set $\epsilon > 0$ and $\vec{p}_0 = (p_{1,0}, p_{2,0}, \cdots , p_{s,0})$ with $p_{1,0} = 1$, and $p_{i,0} = 0$, for $1 < i \leq s$. Go to line 2.
2. Iterations. For $k > 0$, compute $\vec{p}_k = \underset{1\leq i\leq s}{\text{argmin}} \left(\mathbb{1}_{\vec{p}_{k-1}+\epsilon\vec{e_i}\in[0,1]^s} g(\vec{p}_{k-1} + \epsilon\vec{e_i}) + \mathbb{1}_{\vec{p}_{k-1}+\epsilon\vec{e_i}\notin[0,1]^s} g(\vec{p}_{k-1})\right)$ where $\vec{e_i}$ is a line vector with all coefficients equal to $-\frac{1}{s-1}$ except the ith coefficient which is chosen equal to 1. If $g(\vec{p}_k) < g(\vec{p}_{k-1})$, then go back to line 2. Otherwise go to line 3.
3. The vector $\vec{p}_k$ approximates the parameters of the optimal Bernoulli policy.

Next, we improve this optimal Bernoulli policy from the minimizing action between sending the truck to Station 1, Station 2, ..., or Station s at phase y. Assume that Station i is in state x_i at phase y, for $1 \leq i \leq s$, $0 \leq x_i \leq c_i$ and $1 \leq y \leq N$. We denote by $V_i(x_i, y)$ and x^*_i the value function and the optimal state for Station i under Policy $\pi^*_{p_1,p_2,\cdots,p_s}$. The value function of the set of the s stations is $V(x_1, x_2, \cdots , x_s, y) = V_1(x_1, y) + V_2(x_2, y) + \cdots + V_s(x_s, y)$. The action of the truck consists in changing the state of one station to its optimal state. If Station i is reset to its optimal state then the value function becomes $V_1(x_1, y) + V_2(x_2, y) + \cdots + V_{i-1}(x_{i-1}, y) + V_i(x^*_i, y) + V_{i+1}(x_{i+1}, y) + \cdots + V_s(x_s, y) = V(x_1, x_2, \cdots , x_s, y) + V_i(x^*_i, y) - V_i(x_i, y)$. We want to minimize this expression by determining the best station to reset. Hence, if $m = \underset{1\leq i\leq s}{\text{argmin}}\ V(x_1, x_2, \cdots , x_s, y) + V_i(x^*_i, y) - V_i(x_i, y)$, then it is optimal to send the truck to Station m. In case strictly more than one station minimize $V(x_1, x_2, \cdots , x_s, y) + V_i(x^*_i, y) - V_i(x_i, y)$, one should apply an equiprobably routing among the stations which minimize this function. This step of improvement determines the improved policy from Station j, π_{I_j}. Algorithm 2 summarizes the procedure to compute Policy π_{I_j} from a given Station j ($1 \leq j \leq s$). Next, by considering each other station as the reference location, one can determine π_{I_1}, π_{I_2},..., and π_{I_s}. The improved policy for the dynamic bike repositioning problem, π^*_I can then be fully characterized.

Algorithm 2: Policy π_{I_j}.

1. Assume that the truck is at Station j at phase y ($1 \leq y \leq N$). Estimate the system parameters: $\lambda_{i,1}$, $\lambda_{i,2}$, ..., $\lambda_{i,N}$, $\mu_{i,1}$, $\mu_{i,2}$, ..., $\mu_{i,N}$, c_i, $\gamma_{j,i}$, θ and N, for $1 \leq i \leq s$. Go to line 2.

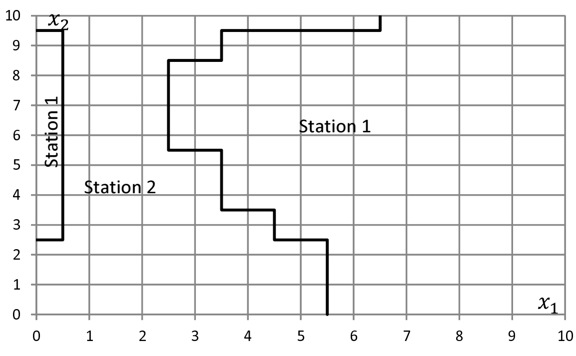
2. Determine the optimal Bernoulli policy, $\pi^*_{p_1,p_2,\cdots,p_s}$, with the optimal inventory levels at each station, x_i^*, and the related value function at Station i, $V_i(x_i, y)$, where $0 \leq x_i \leq c_i$ using the results of Section 4.2 and Algorithm 1. Go to line 3.

3. Choose the station to prioritize by improving the Bernoulli policy, $m = \underset{1\leq i\leq s}{\text{argmin}}\ V(x_1, x_2, \cdots, x_s, y) + V_i(x_i^*, y) - V_i(x_i, y)$.

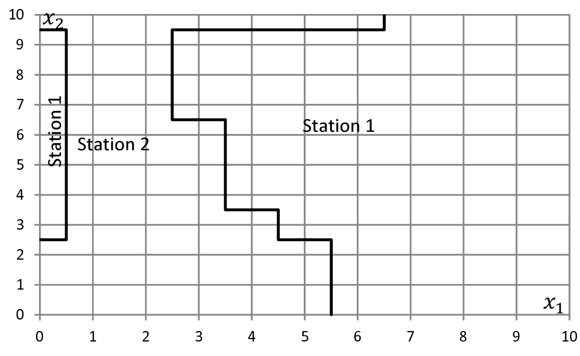
Computational complexity. Computing policy π_I^* is faster than using policy iteration for deriving the optimal policy using Equation (2). Consider a situation with s stations each with capacity c. To obtain the optimal policy using Equation (2), one has to derive the value function on a state space of dimension $s \times N \times (c+1)^s$ at each iteration of the value function. The computational complexity of this problem is exponential. This in turn renders the optimal policy difficult to obtain as the number of stations increases. To obtain policy π_I^*, each station is taken in isolation. Hence, only $N \times (c+1)$ values per station have to be computed to obtain a given Bernoulli policy. These values correspond to the values of the value function defined in Section 4.2 under a given Bernoulli policy. With s stations, this leads to $s \times N \times (c+1)$ values to compute for determining a given Bernoulli policy. Hence, the computational complexity becomes linear with the one-step policy improvement method.

5.2 Comparison with the optimal Policy

We first conduct a graphical comparison between the optimal policy and Policy π_I^* for the routing problem of the truck to two parallel bike stations called Station 1 and Station 2 from a third station called Station 3. With only 3 stations, the state space is sufficiently small to obtain the optimal policy using Equation (2). In most cases, the optimal policy and Policy π_I^* are identical. In Figure 6, we provide an example where these two policies slightly differ. Figure 6 reveals that the optimal policy and Policy π_I^* follow a "closest to



(a) Optimal policy obtained with Equation (2)



(b) Improved policy obtained via the one-step policy improvement method

Figure 6: Comparison between the optimal and the improved policy ($\lambda_{1,1} = 0.4$, $\lambda_{2,1} = 0.1$, $\mu_{1,1} = 0.1$, $\mu_{2,1} = 0.2$, $\gamma_{3,1} = 0.25$, $\gamma_{3,2} = 0.2$, $\gamma_{3,3} = 1000$, $c_1 = c_2 = 10$, $r_{1,1} = r_{1,2} = r_{2,1} = r_{2,2} = 1$, $N = 1$)

boundary first" principle. Further numerical investigations on Policy π_I^* are given in Section 6 of the online supplement to determine the key drivers of the station's prioritization. The rule of thumb developed using Policy π_I^* is to prioritize the closer, the more active, the closer to be full or empty, and the more imbalanced stations if no reversing in the imbalance can be anticipated. In addition to this observation, we are also interested in comparing the performance of these two policies. This can be done using simulations. In what follows, we describe the setting of the numerical experiments. This setting will be extended in Section 6 to closer to reality assumptions.

Setting of the numerical experiments. In Table 1, we give the setting of the numerical experiments for a repositioning zone with s stations. For notational simplicity, we assume that Station i and Station $i+s$ are the same station ($i \geq 1$). For the arrival and departure processes in the zone, we consider a cyclic evolution over a period of 24 hours. With the definition of the departure process, there is a peak hour around 8 am and an off-peak hour around 8 pm. This may correspond to what can be observed in residential areas (Vogel, 2016). We propose different cases for the arrival process. Case 1 corresponds to a constant arrival parameter. Case 2 illustrates a situation where the cyclic evolution of the arrival process follows the one of the demand for bikes. In this case the ratio $\lambda(t)/\mu(t)$ is maintained constant. This case illustrates the variation in the activity of the BSS without changes in the imbalance between arrivals and departures. Case 3 represents an arrival process evolving in opposition with the demand for bikes. This situation is typical of a residential area where a pic of demand is encountered in the morning and a pic of arrivals is encountered in the afternoon. Once a bike arrives (respectively will be rented) in the repositioning zone, the renter chooses Station i with probability q_{a_i} (respectively, with probability q_{d_i}) for $1 \leq i \leq s$. Again, we consider 3 cases defined as a function of the parameter q chosen sufficiently high. Case a corresponds to an equiprobable choice. Case b translates a preference for Station 1 at arrivals and departures. Case c illustrates a preference for Station 1 at arrivals and preference for Station s at departures. We call Case $i.j$, the combination of Case i for the arrival process in the zone ($i = 1, 2, 3$) and Case j for the users' preference among stations ($j = a, b, c$). After discritization on each interval of expected length 30 minutes, we assume that the arrival and departure parameters at each station are constant and equal to their average value. The definition of the parameters $\gamma_{i,j}$ translates a circular geographical location of the stations.

Table 2 tabulates the performance of the optimal policy and the performance of Policy π_I^*. Instead of giving the rate of arrival of unsatisfied users, we provide the proportion of unsatisfied users (i.e., it is ratio between the long run rate of arrival of unsatisfied users and the long run overall rate of arrival of users) which is easier to analyze. As a reference, we also provide the performance in a situation without intervention. The performance of the optimal policy is obtained using Equation (2). As n tends to infinity, $V_{n+1}(\overline{x}) - V_n(\overline{x})$ as given in Equation (2) tends to the long-run average rate of arrival of unsatisfied users. Hence, the performance of the optimal policy can be obtained without simulations. Policy π_I^* is evaluated using simulation. After each intervention, the simulation is interrupted and Policy π_I^* is computed using

Table 1: Setting of the numerical experiments (t is the number of hours from the beginning of the simulation)

Departure process in the zone	$\mu(t) = \mu(1 + \sin(\frac{2\pi}{24}(t-2)))$
Arrival process in the zone	Case 1: $\lambda(t) = \lambda$
	Case 2: $\lambda(t) = \lambda(1 + \sin(\frac{2\pi}{24}(t-2)))$
	Case 3: $\lambda(t) = \lambda(1 + \sin(\frac{2\pi}{24}(t-2) + \pi))$
Users' preference for Station i ($1 \leq i \leq s$)	Case a: $q_{a_i} = q_{d_i} = \frac{1}{s}$
	Case b: $q_{a_1} = q_{d_1} = q$, and $q_{a_i} = q_{d_i} = \frac{1-q}{s-1}$, for $2 \leq i \leq s$
	Case c: $q_{a_1} = q$, $q_{d_s} = q$, $q_{a_i} = \frac{1-q}{s-1}$, for $2 \leq i \leq s$ and $q_{d_i} = \frac{1-q}{s-1}$, for $1 \leq i \leq s-1$
Capacity of Station i	$c_i = 20$, for $1 \leq i \leq s$
Parameters of time discretization	$N = 48$ phases and $1/\theta = 30$ minutes
Parameters $\gamma_{i,j}$ ($1 \leq i,j \leq s$)	$\gamma_{i,i+j} = \gamma_{i,i-j} = \frac{\sqrt{j}}{\gamma}$, for $1 \leq i \leq s$, and $0 \leq j \leq \lfloor \frac{s}{2} \rfloor$
Objective: Minimize the overall rate of arrival of unsatisfied users	$r_{1,i} = r_{2,i} = 1$, for $1 \leq i \leq s$

Algorithm 2 to decide which station should be reset.

Table 2: Performance evaluation ($s = 3$ stations, $c_1 = c_2 = c_3 = 20$, $\lambda = 1/10$ min^{-1}, $\mu = 3/10$ min^{-1}, $\theta = 1/30$ min^{-1}, $\gamma = 1/360$ min^{-1}, $q = 50\%$)

Cases	Proportion of unsatisfied users		
	No-intervention	Optimal Policy	Policy π_I^*
Case 1.a	52.056%	39.042%	39.136%
Case 1.b	52.174%	39.652%	39.732%
Case 1.c	54.248%	40.144%	40.192%
Case 2.a	60.673%	49.145%	49.210%
Case 2.b	60.751%	49.816%	49.965%
Case 2.c	61.784%	49.427%	49.497%
Case 3.a	68.601%	42.533%	42.575%
Case 3.b	68.716%	46.727%	46.867%
Case 3.c	71.358%	46.811%	46.886%

The truck intervention significantly improves the system performance. In average on the different cases, the proportion of unsatisfied users is reduced by 26% between the no-intervention policy and the optimal policy. The performance of Policy π_I^* is in average 0.19% above the performance of the optimal one. These illustrations show that the one-step policy improvement method yields *nearly optimal* policies. These observations meet the ones of Hwang et al. (2000) and Bhulai and Koole (2003) who have also shown the close proximity between the improved and the optimal policy in other queueing contexts.

In addition, we observe that the time variations of the arrival and departure parameters have a strong influence (Comparison of Cases 1,2,3) compared to the users' preference for a given station (Comparison of Cases a,b,c). In Case 1, with a constant arrival parameter, the stations alternate between arrival dominant and departure dominant periods without reaching highly imbalanced periods. This may create beneficial compensation effects. In Case 2, the imbalance between arrivals and departures is maintained constant at any point of time. This cancels the compensation effect that the time variations could provide and leads to worse performance than in Case 1. In Case 3, with an opposite evolution between arrivals and departures, the compensation effect is reduced by the highly imbalanced periods encountered. For instance at 8 am, one departure occurs every 5 minutes without any arrival. The effect of the operators' intervention varies from an improvement of 19% in average in Case 2 to 34% in average in Case 3. It seems that the operator can beneficially use the alternations between arrival dominant and departure dominant periods to reset the stations.

The users' preference for one station has a small effect on the overall system performance. Yet, an area in the BSS performs better when the arrivals and departures are equally spread over the different stations. This can be intuitively understood from the case with constant arrival and departure rates. The rate of arrival T_1 (respectively T_2) as given in Theorem 2 is increasing and convex in μ (respectively in λ). Applying the Jensen's inequality for s stations leads to $\frac{1}{s}\sum_{i=1}^{s} T_1(\mu_i) \geq T_1(\sum_{i=1}^{s} \frac{\mu_i}{s})$. In the inequality, the left hand side corresponds to the average rate of arrival of users in need of a bike for an arbitrary situation, i.e., with arbitrary values of μ_i's. As for the right hand side, it gives the overall rate of arrival of users in need of a bike for a symmetric situation, i.e., all the μ_i's are identical. The same inequality applies for T_2. This shows that the symmetric case is preferred. However, since T_1 is almost a linear function of μ, the difference between the symmetric case and any other one is not significant.

6 Validation of Policy π_I^* in a more realistic setting

Policy π_I^* is derived from an idealized mathematical framework. Some of the assumptions made to derive this policy can be questionable. We propose here to extend the setting of Table 1 to a context closer to reality and evaluate π_I^* in comparison with other routing policies for a larger number of stations. First, we assume that the arrival processes at each station are independent. In practice, if a user does not find an empty dock she/he will cycle to another station. Using online applications, a cyclist can know in real time which station is full or not. We assume that cyclists choose a "closest first policy" for the choice of the first available station. A bike renter is forced to return a bike, hence all stations are checked to return the bike whereas a user in need of a bike can choose another transportation if there is no available bike in the neighborhood. Hence, if at a bike arrival the wanted Station i is full then the renter checks if Station $i+1$ or Station $i-1$ is full. If both stations are not full then the user chooses Station $i+1$ with probability 50%. If only one station is not full, then this station is chosen. Finally, if the two stations are full then the user checks Station $i+2$ or $i-2$ and so on until all stations are checked. If all stations are full then the renter moves to another zone. When a user wants to rent a bike the same "closest first policy" is applied. However, we choose to limit the choice to the two adjacent stations. For the performance evaluation, we count a penalty of 1 if the initial wanted station is not available. In this way, we evaluate the rate of arrival of unsatisfied users as defined in Equation (1) with $r_{1,i} = r_{2,i} = 1$, for $1 \leq i \leq s$. This means that the operator is only interested in allowing users to get their primary choice. With this definition, a difficulty to determine Policy π_I^* is the estimation of the arrival processes at each station. These ones may be influenced by the arrival process in the other stations and by the operator's interventions. We choose to ignore this difficulty by only considering the arrival and departure processes that would happen if users could all find an empty dock or an available bike at each station. In this sense, the operator is willing to adjust its policy to the users' initial preference and not to their actual behavior.

Second, we assume that the truck's movements are exponential. Although the traffic and the time to operate at a station may create some variability, this variability is certainly lower than the one of the exponential distribution. We propose here to replace the exponential distribution by an Erlang one with 5 phases and the same expected value as the one defined in Table 1. Finally, the limited capacity of the truck does not allow any station to be reset at its optimal state at any time. We assume that the truck capacity is of 20 bikes (as in the BSS of Paris). After an intervention in a station, the truck decides which station should be the next one to be reset according to a given policy. Once this station is chosen, either the truck has the capacity in empty spaces and available bikes to improve the situation in the chosen station (i.e., to get closer to its optimal state) and the chosen station is reset as close as possible to its optimal state. Otherwise, the station is excluded from the choice of the stations to reset. If all stations in the repositioning zone cannot be improved by the truck intervention, then the truck is directly replaced by either a completely full or a completely empty truck starting from the same station according to what is the most profitable for the repositioning zone. This assumption is motivated by the possibility to go back to the maintenance zone to load or unload the truck.

In this context the optimal policy cannot be found. We instead propose to compare Policy π_I^* to other simple policies defined in Table 3. Note that an equiprobable choice is implemented if more than one station the quantities mentioned in Table 3. The choice between two stations for Policy π_1 allows the truck to go

Table 3: Policy definitions

Policy	Prioritization rule from Station i
Policy π_1 ("closest first policy")	Station which maximizes $\lvert x_j - x_j^*\rvert$ for $j = i+1$ or $j = i-1$.
Policy π_2 ("longest empty/full first policy")	Station which is for the longest time full or empty. If none of the stations are full or empty, Policy π_1 is applied.
Policy π_3 ("highest risk first policy")	Station which maximizes the quantity $\lvert\overline{\lambda_j} - \overline{\mu_j}\rvert \cdot \lvert x_j - x_j^*\rvert \cdot 1/\gamma_{i,j}$, where $\overline{\lambda_j}$ and $\overline{\mu_j}$ are the average arrival and departure rates at Station j in the next period of expected duration $1/\theta$.

back and forth to a problematic station (i.e., a very imbalanced or active one). The simple rule proposed for Policy π_3 allows us to consider the future imbalance, the actual state of each station and the time to reset this station which are the key drivers for the station's prioritization as shown in Section 6 of the online supplement.

Table 4 tabulates the performance of each policy. For the comparison, we also give the performance of Policy π_I^* with exponential times to reset Station i from Station j (instead of the Erlang distribution), referred to as Policy $\pi_{I\text{exp}}^*$ and with a truck of infinite capacity in bikes and empty docks, referred to as Policy $\pi_{I\infty}^*$. After each truck intervention, the simulation is interrupted and Policy π_I^* is derived by replacing the above assumptions by the ones of Section 5.2. For the other policies, we assume that the optimal state in each station is identical to the one that could be obtained with Policy π_I^*. This is consistent with the idea that the optimal state does not depend on the truck capacity as shown in Section 3. The system parameters are adjusted such that we consider the same imbalance per station and the same intensity of the repositioning activity as in the repositioning zone considered in Section 5.2.

Table 4: Performance evaluation ($c_i = 20$, for $1 \leq i \leq s$, $\lambda = s/30$ min^{-1}, $\mu = s/10$ min^{-1}, $\theta = 1/30$ min^{-1}, $\gamma = s/(3 \times 360)$ min^{-1}, $q = \frac{1.5}{s}$)

		Proportion of unsatisfied users						
	Cases	No-intervention	Policy π_I^*	Policy π_1	Policy π_2	Policy π_3	Policy $\pi_{I\text{exp}}^*$	Policy $\pi_{I\infty}^*$
$s = 10$	Case 1.a	63.91%	48.66%	51.10%	50.07%	50.95%	48.71%	43.31%
	Case 1.b	63.95%	48.75%	52.17%	53.79%	50.95%	48.81%	42.90%
	Case 1.c	65.02%	49.69%	53.42%	55.75%	51.87%	49.77%	45.22%
	Case 2.a	69.63%	55.99%	59.18%	57.88%	58.34%	56.07%	49.50%
	Case 2.b	70.11%	56.93%	61.09%	63.16%	59.15%	57.01%	51.92%
	Case 2.c	71.42%	58.87%	63.35%	65.60%	61.06%	58.93%	53.81%
	Case 3.a	75.69%	55.25%	58.31%	56.15%	57.65%	55.35%	50.00%
	Case 3.b	76.14%	55.78%	60.07%	61.95%	58.01%	55.88%	50.65%
	Case 3.c	78.89%	57.98%	62.56%	64.62%	60.16%	58.08%	52.70%
$s = 30$	Case 1.a	65.24%	52.90%	56.71%	57.39%	55.39%	53.00%	48.35%
	Case 1.b	65.68%	53.47%	58.61%	60.48%	55.75%	53.55%	49.41%
	Case 1.c	66.78%	54.48%	59.41%	61.54%	56.74%	54.57%	49.90%
	Case 2.a	71.18%	59.25%	63.58%	64.72%	61.81%	59.36%	55.16%
	Case 2.b	71.21%	59.43%	65.25%	67.47%	62.70%	59.53%	55.98%
	Case 2.c	71.96%	61.34%	67.17%	69.25%	64.55%	61.41%	55.21%
	Case 3.a	76.12%	59.48%	64.06%	65.06%	62.05%	59.55%	52.94%
	Case 3.b	76.69%	60.07%	66.19%	68.05%	63.35%	60.14%	53.16%
	Case 3.c	80.06%	63.64%	70.26%	72.29%	66.89%	63.76%	55.75%

Similar observations as the ones of Table 2 can be made here for the effect of the time variations, the asymmetry in the preference for one station and the improvement made by Policy π_I^* compared to the no-intervention policy (- 21% of unsatisfied users in average compared to the no-intervention Policy). As expected, Policy π_I^* outperforms the other proposed policy. The value of Policy π_1 is to minimize the driving times. This allows the operator to maximize the intensity of the activity in the repositioning zone. Yet, Policy π_1 ignores distant stations even if these are in need of repositioning. This leads to a deteriorated performance compared to Policy π_I^* (+7.89% of unsatisfied users in average) in particular when there is an important asymmetry as in Cases b and c.

Policy π_2 is worse than Policy π_1 (except in Case a for $s = 10$). This policy does not perform well for two reasons. First, the stations which are the longest empty or full are often the less active. If the time between two arrivals is long then a station may stay empty a long time. Hence, this policy may prioritize the less active stations where the lowest quantity of unsatisfied users will be observed. Second, Policy π_2 does not anticipate the future evolution of the arrival and departure parameters. This may lead to reset a station when the future arrivals and departures could rebalance effectively the station without truck intervention.

Remark. In practice, the principle of Policy π_2 is often applied. The reason is that the operator has a more complex optimization objective in mind than only minimizing the overall proportion of unsatisfied users. The operator also cares about the *fairness* between stations. Ignoring a given station can be optimal for Problem (1). However, it results in neglecting the neighborhood around the station. As a public service, the operator may then force repositioning in less active stations to provide a fair service among stations.

In most cases, Policy π_3 outperforms Policies π_1 and π_2. The value of Policy π_3 is that it follows the key drivers for the station's prioritization. Although the performance of this policy is relatively close to policy π_I^* (+ 4.1% of unsatisfied users in average compared to Policy π_I^*), it can be outperformed by Policy π_2 as in line 1. Moreover, if the arrival and departure rates were identical and constant, Policy π_3 would apply an equiprobable choice among stations. This would correspond to the optimal Bernoulli Policy which is far from being optimal. In general, the literature on the dynamic repositoning problem has shown that simple policies like Policy π_3 fail in answering the diversity of situations encountered in a BSS.

The possibility for users to choose another station as a secondary choice deteriorates the performance in the repositioning zone. This can be seen by comparing the results for $s = 3$ in Table 2, and $s = 10$ and $s = 30$ in Table 4. The reason is the increasing of the congestion in the system. In the model described in Section 2, users are rejected from the system if they do not find an empty dock or an available bike. Here, they choose another station which in turn further increases the imbalance between arrivals and departures in the repositioning zone. The larger this zone is, the higher chance of finding an available station is. Hence, by considering a large repositioning zone for the performance evaluation, less users decide to leave the zone which deteriorates the measured performance. This questions the dimension of the area to measure the system performance. Our observation indicates that with the same intensity of activity per station, it is easier to achieve a good performance over a small repositioning zone. Despite the decrease in performance as the number of stations is increased in the zone, we note that the degradation is decreasing in s, suggesting an eventual saturation. We observe that changing the exponential assumption to an Erlang one for the time to reset a station improves the performance. The reason of the improvement is the reduction of the variability. However, the improvement is limited (only -0.15% of unsatisfied users in average compared to the exponential case). The expected value of truck's movements has a predominant influence compared to its variability. Finally, the limitation of the truck capacity strongly deteriorates the performance (+10.5% of unsatisfied users in average compared to the infinite capacity truck). This is particularly the case since we consider a strongly imbalanced zone. After a given number of interventions, the truck is not able to significantly improve the system state. In order to improve the performance, it is then necessary to load or unload the truck at the maintenance zone. We observe that the number of visits of the maintenance zone per day in our simulations varies from 2.04 in Case 2.c with $s = 10$ to 1.72 in Case 1.a with $s = 30$. This means that in cases of highly imbalanced repositioning zones with a small number of stations like in the suburbs, it is necessary to often reset the truck to ensure a sufficiently high quality of service.

7 Concluding remarks

We modeled a single bike station with time-dependent arrival and departure processes. We derived its relative value function under the proven optimal policy for bike repositioning. We further developed a one-step policy improvement to determine which station should be prioritized. Although the one-step policy improvement did not allow us to obtain an optimal policy, this method solved the problem referred as "the curse of dimensionality". The close proximity with the optimal policy was shown in systems with a small number of stations. Moreover in more general settings, we have shown that our policy outperforms policies built on simpler prioritization rules.

Several interesting areas of future research arise. Although our method is computationally efficient, it is also built on a simplification of the real system. We have shown the efficiency of our policy in a more general setting. Yet, it would be interesting to include these features inside the Markov decision process approach. For instance, the time to reach a station for the truck is assumed to be exponentially distributed. This may be justified by the randomness of the traffic but the exponential distribution may overestimate the variability of the time spent on the road when there is no traffic. It would be interesting to know which distribution fits the best to the driving times so as to approximate it with an appropriate phase-time distribution. Finally, our study allows obtaining truck routing solutions when the expected arrival and departure rates can be anticipated. This may not always be the case or at least the forecast may not always be accurate. It could then be interesting to develop other methods which could be less depending on the forecasts.

Acknowledgement. This work was supported by research grants from the European Union under the PERFAD project via the ERDF program from the Normandie region. The author wants to express his gratitude to the reviews' team for their useful comments, that significantly improved this paper.

References

Bachand-Marleau, J., Lee, B., and El-Geneidy, A. (2012). Better understanding of factors influencing likelihood of using shared bicycle systems and frequency of use. *Transportation Research Record: Journal of the Transportation Research Board*, (2314):66–71.

Bellman, R. (1957). *Dynamic programming.* Princeton University Press, Princeton.

Bellman, R. (1961). *Adaptive control processes: a guided tour.* Princeton University Press, Princeton.

Bhulai, S. and Koole, G. (2003). On the structure of value functions for threshold policies in queueing models. *Journal of Applied Probability*, 40(3):613–622.

Borgnat, P., Abry, P., Flandrin, P., and Rouquier, J. (2009). Studying Lyon's Vélo'v: a statistical cyclic model. In *ECCS'09*. Complex System Society.

Boucherie, R. and van Dijk, N. (2017). Markov decision processes in practice.

Bulhões, T., Subramanian, A., Erdoğan, G., and Laporte, G. (2018). The static bike relocation problem with multiple vehicles and visits. *European Journal of Operational Research*, 264(2):508–523.

Datner, S., Raviv, T., Tzur, M., and Chemla, D. (2017). Setting inventory levels in a bike sharing network. *Transportation Science*.

DeMaio, P. (2009). Bike-sharing: History, impacts, models of provision, and future. *Journal of Public Transportation*, 12(4):3.

Erdoğan, G., Laporte, G., and Calvo, R. (2014). The static bicycle relocation problem with demand intervals. *European Journal of Operational Research*, 238(2):451–457.

Fianu, S. and Davis, L. (2018). A Markov decision process model for equitable distribution of supplies under uncertainty. *European Journal of Operational Research*, 264(3):1101–1115.

Forma, I., Raviv, T., and Tzur, M. (2015). A 3-step math heuristic for the static repositioning problem in bike-sharing systems. *Transportation research part B: methodological*, 71:230–247.

Frade, I. and Ribeiro, A. (2015). Bike-sharing stations: A maximal covering location approach. *Transportation Research Part A: Policy and Practice*, 82:216–227.

Fricker, C. and Gast, N. (2014). Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity. *EURO Journal on Transportation and Logistics*, pages 1–31.

Fricker, C. and Servel, N. (2016). Two-choice regulation in heterogeneous closed networks. *Queueing Systems*, 82(1-2):173–197.

Ghosh, S., Varakantham, P., Adulyasak, Y., and Jaillet, P. (2015). Dynamic redeployment to counter congestion or starvation in vehicle sharing systems. In *Eighth Annual Symposium on Combinatorial Search*.

Haider, Z., Nikolaev, A., Kang, J., and Kwon, C. (2014). *Inventory rebalancing through pricing in public bike sharing systems*. PhD thesis, Citeseer.

Howard, R. (1960). Dynamic programming and markov processes.

Hwang, R., Kurose, J., and Towsley, D. (2000). MDP routing for multi-rate loss networks. *Computer Networks*, 34(2):241–261.

Kaspi, M., Raviv, T., Tzur, M., and Galili, H. (2016). Regulating vehicle sharing systems through parking reservation policies: Analysis and performance bounds. *European Journal of Operational Research*, 251(3):969–987.

Kek, A., Cheu, R., Meng, Q., and Fung, C. (2009). A decision support system for vehicle relocation operations in carsharing systems. *Transportation Research Part E: Logistics and Transportation Review*, 45(1):149–158.

Legros, B. (2017). Reservation, a tool to reduce the balking effect and the probability of delay. *Operations Research Letters*, 45(6):592–597.

Li, Y., Zheng, Y., Zhang, H., and Chen, L. (2015). Traffic prediction in a bike-sharing system. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 33. ACM.

Lin, J. and Yang, T. (2011). Strategic design of public bicycle sharing systems with service level constraints. *Transportation research part E: logistics and transportation review*, 47(2):284–294.

Lin, J., Yang, T., and Chang, Y. (2013). A hub location inventory model for bicycle sharing system design: Formulation and solution. *Computers & Industrial Engineering*, 65(1):77–86.

Ma, B., Liu, C., and Erdoğan, S. (2015). Bicycle sharing and transit: Does capital bike-share affect metrorail ridership in washington, dc? In *83rd Annual Meeting of the Transportation Research Board*.

Martinez, L., Caetano, L., Eiró, T., and Cruz, F. (2012). An optimisation algorithm to establish the location of stations of a mixed fleet biking system: an application to the city of Lisbon. *Procedia-Social and Behavioral Sciences*, 54:513–524.

Nair, R. and Miller-Hooks, E. (2011). Fleet management for vehicle sharing operations. *Transportation Science*, 45(4):524–540.

Nair, R. and Miller-Hooks, E. (2014). Equilibrium network design of shared-vehicle systems. *European Journal of Operational Research*, 235(1):47–61.

Nair, R., Miller-Hooks, E., Hampshire, R., and Bušić, A. (2013). Large-scale vehicle sharing systems: analysis of Vélib'. *International Journal of Sustainable Transportation*, 7(1):85–106.

Norman, J. (1972). *Heuristic procedures in dynamic programming.* Manchester University Press.

Ott, T. and Krishnan, K. (1992). Separable routing: A scheme for state-dependent routing of circuit switched telephone traffic. *Annals of operations research*, 35(1):43–68.

Pandelis, D. (2010). Markov decision processes with multidimensional action spaces. *European Journal of Operational Research*, 200(2):625–628.

Puterman, M. (1994). *Markov Decision Processes.* John Wiley and Sons.

Raviv, T. and Kolka, O. (2013). Optimal inventory management of a bike-sharing station. *IIE Transactions*, 45(10):1077–1093.

Schuijbroek, J., Hampshire, R., and Van Hoeve, W. (2017). Inventory rebalancing and vehicle routing in bike sharing systems. *European Journal of Operational Research*, 257(3):992–1004.

Shaheen, S., Martin, E., Cohen, A., Chan, N., and Pogodzinsk, M. (2014). Public bikesharing in North America during a period of rapid expansion: Understanding business models, industry trends & user impacts, mti report 12-29.

Shu, J., Chou, M., Liu, Q., Teo, C., and Wang, I. (2013). Models for effective deployment and redistribution of bicycles within public bicycle-sharing systems. *Operations Research*, 61(6):1346–1359.

Vogel, P. (2016). *Service Network Design of Bike Sharing Systems: Analysis and Optimization.* Springer.

Zhuang, W. and Li, M. (2012). Monotone optimal control for a class of Markov decision processes. *European Journal of Operational Research*, 217(2):342–350.