Bike Sharing System Simulator with User Incentive Methods

Rodrigo Plácido
rodrigo.placido@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

September 2021

Abstract

The interest in creating sustainable and green cities free of carbon emissions has grown in more recent decades, with non-pollutant means of transportation taking a fundamental role in this development process. By joining these criteria, and the concept of shared mobility, Bike Sharing Systems (BSS) emerged. They are present in hundreds of cities and are nowadays a competitive alternative to cars, metro or buses. Nevertheless, maintaining these systems in perfect conditions, so that final customers always have a good experience, is not an easy task. Bike flows tend to generate a lack or excess of bikes in some stations preventing users from picking up or dropping off bikes. There are two major alternatives to solve this issue and bring the network to a steady condition: either operators relocate bikes using trucks to transport them between stations, or users are engaged in the balancing process using incentives. This document is focused on the second alternative. A model, based on historical data from Lisbon BSS, Gira, was created in a simulation software (Anylogic). This is an agent-based model with a visual interface where bike trips and station states can be observed in real time, and some model parameters can be easily changed. Two user incentive (UI) methods were computed and tested in the simulator, both alone and in conjunction with a nightly vehicle repositioning. After several simulations, results have shown that if all users participate in repositioning tasks, it would be possible to keep the system balanced. Since this is not an easy task, it was concluded that, for the experimented methodologies, and to maintain station level requirements, user incentive methods and repositioning tasks have to be both used. The effect that UI methods have on the number of visited stations and transported bikes during repositioning, it is also shown. Although some valuable conclusions are presented in this work, a future research about the costs related to UI and its effect on repositioning tasks, also considering customer cooperation, must be taken.
Keywords: Bike-sharing system; User incentive; Static repositioning; Simulator

1. Introduction

In more recent years, Bike Sharing Systems (BSS) have been widely adopted in many major cities all over the world. The adoption of the bike as a mean of transportation is expected to continue growing, mostly due to governments' efforts to face climate changes, reduce traffic, and also due to increasing health awareness [1–3].

These systems consist of bikes spread around the city and where users can pick them up, travel to their intended destination and drop off the bikes "at specific locations or anywhere in the city depending on the type of (...) system, locking technology, and payment mechanisms" [2]. Two types of systems are recognised: free-floating, which allows people to place bikes anywhere; and station-based, which forces people to pick up and drop off bikes from stations. These stations may use docks or geo-fences to hold bikes [2]. This document will only address the second system type since the problem addresses Lisbon BSS, which is station-based. However, with either free-floating or station-based, trips dynamics may take too many bikes from or to determined stations emptying them out or overloading. There would be no problem if the difference between arrivals and departures was kept close to zero; however, this does not happen for every station: some are more propitious to have departures and other arrivals, leaving those stations in unbalanced states. Besides the possibility of not encountering a bike if a station is empty, a problem that can be solved by using other means of transport, stations might also be full. Actually, this represents a bigger problem since it does not allow people to dock their bikes and consequently forces them to travel to another station with empty docks [2].

The most common way to deal with this issue is by using trucks to move bikes between stations. BSSs surpass this difficulty by applying either static repositioning (system rebalancing when the system is closed for users, usually during the night shift) or dynamic repositioning (rebalancing operations

while the system is open for users) [2, 4, 5]. Another possible approach is having a direct influence on user trips, recommending them to grab bikes at stations highly occupied and/or drop off bikes at stations with low occupations. This alternative may be achieved by changing the prices of trips (most used technique), offering points, etc. [6, 7].
This work addresses Lisbon station-based BSS, Gira, and user incentive approaches to keep the system balanced. An interactive agent-based simulation of Gira is created in Anylogic, based on treated historic data, to estimate the impact of user-based approaches in service quality (percentage of customers satisfied) and repositioning tasks of a NR (Nightly Repositioning).

1.1. Types of repositioning

• Static repositioning consists of finding the best routes and inventory instructions for a fleet of vehicles, while the system is idle for customers to use, for example, during the night. The goal is to rebalance the system so that every station achieves a predetermined inventory level. This type of method is usually processed overnight when the movement is negligible.
• Static repositioning is a good option to reset a BSS to an ideal state or when stations have low fluctuations throughout the day but performs poorly when the Spatio-temporal demand patterns display high variance, easily resulting in full/empty stations. Dynamic repositioning tries to overcome these issues by executing repositioning tasks throughout the day and in real-time [2].
• Last type is characterised by the help of customers on stations' rebalancing operations, in exchange for an incentive. Price incentives are the most used approach, but there are other options like promotions, points, extra minutes, etc.

1.2. Gira BSS

Gira is Lisbon's BSS [8], belongs to EMEL - Empresa Municipal de Mobilidade e Estacionamento de Lisboa, and represents nowadays a green, healthy and proficient alternative mean of transportation inside the city.
Throughout this document, data provided by EMEL and from January to March of 2019, is analysed. At that time, the network had a total of 74 stations, each one is composed of several docks. There are also hundreds of bikes both in the depot and on the streets.
Rain, social events, holidays, among other factors can increase or decrease the number of trips during a day; however, some patterns reiterate. One of them is the considerable difference between workdays, when there is a lot of movement and traffic inside the city, reflected in more trips; and weekends with much less traffic, due to less movement. Another temporal pattern is the three well-identified peaks of traffic during workdays, one in the morning when people arrive at their jobs, one during lunchtime, and the last in the afternoon when people return home.
During the period of available data, Siemens was the company responsible for the maintenance of all Gira assets. To perform this maintenance, multiple teams of operators work day and night. As mean of transport, they have a fleet of five electric trucks (a contractual imposition) with each truck being able to transport up to 5 bikes, and the capability of attaching a trail to increase its capacity to 11, 12 or 15.
Repositioning operations are performed across all day; therefore, a dynamic methodology is executed. Unlike Gira's real type of operation, this work will, as already mentioned, implement a static repositioning methodology to rebalance the system during the night.
Reverting to repositioning tasks, in three months, around 5.000 visits were recorded, with an average of 56 visits per day, in a total of 2.678 work orders.

2. Related work

Static repositioning

Rodrigues [9] based its work on four MILP formulations of Dell'Amico et al. [10], that had as objective function the total travel cost and were solved with a branch-and-cut algorithm. All of them were evaluated by Rodrigues [9], for 74 stations of Gira. He confirmed the conclusion of Dell'Amico et al. [10] that the third formulation presented, generally, outperforms the remaining.
Schuijbroek, Hampshire, and van Hoeve [11] proposed a cluster first route second heuristic and compared it with the MIP (Mixed Integer Programming) and a constraint programming. The objective function consists of minimising the total travelled distance. Their heuristic handles multiple vehicles, which are allowed to visit the same station more than once. Schuijbroek et al. [4] is its last article update, where loading and unloading time factors were added. This dissertation implemented the method of Schuijbroek et al. [4] to get repositioning instructions.
Chemla, Meunier, and Calvo [12] combined a branch-and-cut with tabu search. The first solves a relaxation of the problem, and the second obtains an upper bound of the optimal solution. They focus on the single vehicle problem and define the minimisation of the total travelling distance as a goal.

User incentives

Fricker and Gast [13] propose a stochastic model of a homogeneous bike sharing system; in other words, all stations have the same capacity and demand rate. They analyse the influence of a two-choice model. Users pick up two possible final destinations, chosen at random, and are influenced to return the bike to the emptiest station. According to them, the state of the network was improved by an exponential factor, with these incentives.

Aeschbach, Zhang, Georghiou, and Lygeros [7] focused on customer's ability to balance the system, proposing four different control strategies, where the way they interact with customers is what distinguishes them. They intended to evaluate strategies' efficacy by treating customer cooperation (CC) as a variable parameter; this way, different percentages of cooperation give different service levels. They found that a CC of 50% with a neighbourhood radius (Rad) of 700 meters is enough to balance London's Barclays Cycle Hire (nowadays, Santander Cycle Hire). Part of their work is applied and experimented within this thesis.

Haider, Nikolaev, Kang, and Kwon [6] named its heuristic approach: Iterative Price Adjustment Scheme (IPAS). The goal is to incentivise people to take bikes from or park them at imbalanced stations to make them more imbalanced, therefore creating hub stations. This way, the number of stations visited by trucks to carry inventory repositioning are reduced. Results were obtained with data from Capital Bikeshare in Washington, D.C.

Simulators

Given that BSS simulators are a recent trend, there is not much literature on the topic, nevertheless, in the last few years, their popularity increased [14].

The most common objective presented in simulators' literature is to evaluate repositioning schemes and Caggiani and Ottomanelli [15] is one of those examples. They propose a method where it is assumed that an operating day is divided into discrete time intervals. At the beginning of each interval, the network is updated, and trips are generated "based on relative OD (Origin-Destination) attractiveness". People who cannot find bikes at the departure station are removed and those who arrive at a full station must wait until there is a free dock.

Fernández, Billhardt, Ossowski, and Sánchez [14] built a simulator called Bike3S, again, an agent-based simulator. Their tool is designed to test different station capacities, station distributions, and balancing strategies. This simulator presents a visual interface and allows several configurations to be altered according to preference.

3. Modelling

Trip generation and trip time

One of the main processes when building a BSS simulator is trip generation. Following the work of Pfrommer et al., days were split into 72 slices, so that a new number of trips are generated every 20 minutes for every start-end station relation. This number relies on a Poisson distribution of parameter $\lambda_w(s_{out}, s_{in}, t)$, where s_{out} is the initial station, s_{in} the end station, t the day time interval (e.g. 8:40 to 9am), and w the day type (workday/weekend). $\lambda_w(s_{out}, s_{in}, t)$ values are obtained and stored after taking the average number of trips of each (w, s_{out}, s_{in}, t) relation from real trip data. Since from 2 AM to 6 AM the system is not available for customers, the number of trips generated is zero within this interval. Each of these new trips has a unique trip time that is originated based on a Lognormal distribution of parameter $TripTime(s_{out}, s_{in})$.

Although this simulation creates trips every 20-minute interval, these are not launched in the precise moment they are created, instead, they start after a random uniformly distributed number of minutes between 0 and 20. This avoids caveats like the ones modelled by Pfrommer et al. [5] (all trips created for a determined interval of time, are launched at a precise moment; for example, all trips created for the interval 10 AM to 10:20 AM are launched at 10 AM), and approximates this implementation to reality.

Trip generation and trip time are the two variable inputs that bring dynamism to the system. Below, it is explained why Poisson and Lognormal were the distributions chosen, based on real and fit PMFs (Probability Mass Function).

- $\lambda_w(s_{out}, s_{in}, t)$ *probability distribution*

Data analysis has shown that the PMF of the number of trips generated at each (w, s_{out}, s_{in}, t) relation seems to fit a distribution.

To check which distribution fits the best between Normal and Poisson, the MSE (Mean Squared Error) was used as a measure of goodness of fit:

$$MSE = \frac{1}{n} \sum_{k=1}^{n} (\overline{y}_k - y_k)^2 \tag{1}$$

where n is the number of data points, $\overline{y}_k$ the value returned by the fit and y_k the actual value for data point k.

Table 1 shows that Poisson presents the lower values of AMSE, which, for this case, represents the average value of all $MSE(s_{out}, s_{in}, t)$. This result confirms the use of Poisson in other researches, e.g. Fricker and Gast [13] or Aeschbach et al. [7].

	Fit PMF	
	Normal	*Poisson*
Workday	0.2079	0.0149
Weekend	0.0928	0.0223

Table 1: AMSE between real and fit PMFs of the generated trips for all combinations of (s_{out}, s_{in}, t)

- $TripTime(s_{out}, s_{in})$ *probability distribution*

Just like the number of trips between stations at interval t, trip time also seems to fit a distribution. For this case, the only variables chosen were arrival and departure stations, without considering time variable t or weekday type w. To check which is the best fitting distribution, Normal, Poisson, Rayleigh, Gamma or Lognormal were taken into account. Table 2 shows that the Lognormal has the lowest AMSE (average value of all $MSE(s_{out}, s_{in})$).

Fit PMF

Normal	*Poisson*	*Rayleigh*	*Gamma*	*Lognormal*
0.0997	0.1088	0.1059	0.0895	0.0785

Table 2: AMSE between real and fit PMFs of the total trip time from s_{out} to s_{in} for all combinations of (s_{out}, s_{in})

User behaviour

The behaviour of users was modelled according to the work of Aeschbach et al. [7]. At pick up moments, when a person arrives at a station without available bikes (empty station), that person goes to the nearest station; if that station also does not have bicycles, the person leaves the system. At drop off moments, when a person arrives at a station without available docks (full station), the person travels to the nearest station that is not full.

User incentives (UI)

After trips are generated users have the opportunity of helping in the rebalancing of the system. Usually receiving an incentive for it (points, trips, etc.).
The basis of this thesis is to evaluate the effect of user incentives in Gira BSS. This way, two main methods were used and tested:

- M1 (Method 1) followed the work of Aeschbach et al. [7]. It consists of using App where people may inform the system of its start and end stations, subsequently receiving, as output, alternative stations - that could help to balance stations inventory - to begin and conclude its journey. To calculate these alternative stations, the model uses the concept of neighbours. Stations that are within a certain Rad from the inputted station are considered its neighbours. Therefore, the one from the group which has a higher/lower occupancy level is presented as the best departure/arrival station. This strategy is applied separately and only implemented if the departure station occupancy level is above $1 - WPC$ or the arrival station occupancy level is below WPC, where WPC (Weak Preemptive Control) is the station level control parameter, so that, when considering $WPC = 0$, the strategy is not applied, but if $WPC = 1$, it is always applied. It must be emphasised that when the best station from the group is the one introduced by the user, it will also be the one shown after the strategy is applied.
- M2 (Method 2) tries to reproduce the incentive methodology used by Gira. The concept of neighbour is also considered, however, the nearest station with occupancy level above $1 - WPC$ is presented for departure station or below WPC for arrival station, instead of the best from the group. Another difference to the first method is that when none of the neighbours fulfils the requirements (to be above $1-WPC$ if it is a departure or below WPC if it is an arrival), even though the input station has the worst inventory ratio, the output will be the same as the input. Since Gira applies 0.3 for arrival and 0.7 for departure as the ratios for which people receive points, this was the ratio used ($WPC = 0.7$).

To model people participation behaviour, a CC ratio was taken into account. Since some may not want to accept both arrival and departure alternative station presented by the app, it was also considered that from the users who want to cooperate, 1/3 would agree with the alternative arrival station, 1/3 with the departure station and the last 1/3 to both departure and arrival (see figure 1). These considerations were not taken into account by Aeschbach et al. [7]. In their paper, people accept both departure and arrival.

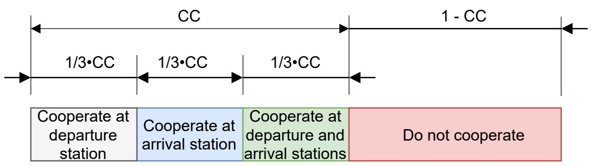

Figure 1: Stacked bar graph of types of cooperation

Repositioning

Though Gira operates with a dynamic repositioning and an algorithm, constantly running to obtain the best orders of repositioning, in this thesis, to drop significantly computational time and since the final objective is not to find the best repositioning method, but instead evaluate UI methods, it was decided to implement the static repositioning method of Schuijbroek et al. [4] which has the particularity of achieving good results in a few seconds or minutes.
Unlike many static methodologies in literature [10, 12], Schuijbroek et al. [4] use target intervals of the station level, instead of target values. To find maximum and minimum target values, for workdays the "time horizon period" from 6 AM to 9 AM was

used because the first peak occurs during this period. The "time horizon period" chosen for weekend days was from 6 AM to 1 PM. This period is larger due to stations' demand curves on weekends, more regular during the morning and with a peak at 4 PM.

4. Anylogic simulator

To implement the model explained in previous sections in Anylogic (simulation software), its major abilities were used: agents and flowcharts. The characteristics of the created agents are summarised below:

- **Agent Main** is the main agent of the program. It is where all the other agents are stored, as well as input files, functions, parameters and variables. Main also has the GIS (Geographic Information System) Map, which during the simulation will show the city map, stations, trucks, and bikes moving between stations, but also the controls where input values are inserted. Main stores the function that initialises the BSS, generate trips and obtain repositioning instructions.
- **Agent Person** defines every user that enters the system right after its trip is created in main and has not yet grabbed a bike. It follows the behaviour of a real user: by deciding whether to accept user incentives or not at the beginning of its trip (a decision based on a CC ratio); or by walking to another station if its intended initial station has no bikes available. All people follow a flowchart and take actions depending on their decisions and the stations' occupations. It is inside this agent that user incentive methods are used to calculate alternative stations and is where the person decides to take them or not.
- **Agent Bike** represents every bike in the system. It is activated when a person grabs a bike and takes a trip between stations. Just like type Person, its actions are taken by following a flowchart, with full stations taking major importance on the followed path.
- **Agent Station** is a static type agent that represents BSS stations. It is mainly characterised by its coordinates, name, and identification number, number of docks, occupation percentage, etc.
- **Agent Depot** is also a static agent that represents BSS depot. It is the place where trucks initiate and finish their tasks and where bikes can be stored if at the end of the repositioning trucks are not empty. It is only characterised by its location.
- **Agent Truck** represents the population of trucks responsible for repositioning tasks. After repositioning instructions (path and number of bikes to drop or collect) are calculated, in agent Main, for each truck, they are activated to move bikes between stations or to depot. To complete its tasks, follows a flowchart, just like Bike and Person. It is mainly characterised by instructions received and the number of bikes capable of carrying.

5. Evaluation

This chapter presents the results of experiments carried out in the simulator. These experiments took place in a computer equipped with Windows 10, an Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz, and 8.00GB of RAM. Initially, the model will be validated in all of its components, followed by an analysis to UI methods.

5.1. Parameters choice

Before validating the model and getting results from the application of different methods for different parameters values, some inputs were considered to be constant for every run:

- Truck speed = 50 km/h;
- Person speed = 1.4 m/s;
- Loading/Unloading task time = Number of bikes to (un)load x 1 minute;
- Number of trucks = 4;
- Truck capacity = 25.

5.2. Verification and validation

To validate this model and assess how close this model fits reality, filtered data from Gira was used, and the parameters presented in the previous subsection were applied. The simulator was validated by examining three major measures: number of trips generated, trip time, and number of users lost.

Datasets

To check the error between reality and the model, two $\lambda_w(i, j, t)$ datasets were created, to input parameters into the simulation. The first testing data, *dataset* 1, included all available trips after being filtered, corresponding to 3 months of trips. With the goal of analysing if the simulator is a good predictor, a second test dataset, *dataset* 2, with data from 2^{nd} of January to 15^{th} of March (excluding the Carnival week from 4^{th} of March to 8^{th}), was used.

Scenarios

Four scenarios, described below, were assessed:

(1) Since stations with restrictions have an impact on user initial preferences for departure and arrival stations, this first scenario used sink/source stations, which mean that station restrictions were not considered. The dataset used was *dataset* 1, and the comparison was made against the average values per day of the same 3 months of real data.

(2) Similar to scenario (1), this scenario attempts to evaluate the prediction capabilities of this simulator, using sink/source stations. It was trained with *dataset* 2, and it was tested and compared for two weeks of data (16 of March to 31^{st} of March).

(3) Scenario for which station and bike restrictions were considered, applying methodologies related to Person and Bike behaviours modelled in previous sections. For this case, UI methods and repositioning are turned off. *dataset* 1 was used as training and were simulated for 28 straight days. Then compared to the average of 3 months of real data.

(4) Similar to 3^{rd} scenario (dataset and simulated days), except for one difference: NR method was activated. The main difference for the third relies on a reset made every night, using a fleet of 4 trucks with a capacity to transport up to 25 bikes each.

To study the number of trips are generated and validate these values,

Trip generation
Trips were counted in four different formats, with one or two metrics per format.
Three metric types were applied for this validation: MAPE (Mean Average Percentage Error), MAE (Mean Average Error), and VAF (Variance Accounted For), which checks the proximity between departure and arrival signals. A VAF value of 1 means that both signals have the same shape.

$$MAPE = abs(\overline{D} - D)/D \tag{2}$$

$$MAE = mean(abs(\overline{D}_{i,j} - D_{i,j})) \tag{3}$$

$$VAF = 1 - var(\overline{D}_{i,j} - D_{i,j})/var(D_{i,j}) \ \ \forall\, i \in I \tag{4}$$

These formats and their respective metrics are presented below:

- MAPE where $\overline{D}$ is the average number of trips per day in the simulator and D the real demand;

- MAE where $\overline{D}_{s_{out},s_{in}}$ is a 2D counter matrix with the number of trips starting at s_{out} and ending at s_{in}, and $D_{s_{out},s_{in}}$ the OD matrix based on real demand;

- MAE where $\overline{D}_{s_{out},t}$ is a 2D counter matrix with the number of trips starting in s_{out} at time interval t, and $D_{s_{out},t}$ the respective real matrix;

- VAF where $\overline{D}_{s,t}$ is the counter vector with number of trips starting in $s \in S_{out}$ at time interval t and $D_{s,t}$ the respective real counter vector. Since this metric results in n metrics where n=number of stations, will be evaluated the mean of all stations' VAF, called MVAF (Mean Variance Accounted For);

- MAE where $\overline{D}_{s_{in},t}$ is a 2D counter matrix with the number of trips starting in s_{in} at time interval t, and $D_{s_{in},t}$ the respective real matrix;

- VAF where $\overline{D}_{s,t}$ is the counter vector with number of trips starting in $s \in S_{in}$ at time interval t and $D_{s,t}$ the respective real counter vector.

The average results of multiple days were considered. Stations (or pairs) will not be evaluated separately, therefore there might be cases with higher errors than the average of errors presented.
Table 3 summarises results of the metrics applied for the four scenarios, differentiating workdays (WD) from weekends (WE). Scenario (1) validates the generation of trips and scenario (3) the generation of trips with station restrictions. As expected, with the repositioning activated, scenario (4) achieved better values than (3). Even though scenario (2) has low MAEs, MVAF values are too low, therefore, it is not recommended to use this model as a trip generation predictor.

Repositioning (intervals and path)
The last factor to check is related to the static repositioning applied.
A simple truck route validation can be made by looking at the figure 2, that shows each truck route and the number of bikes picked up or delivered at each station.
Figure 3 also validates the repositioning, showing that all self-sufficient station (blue dot) occupancy is maintained while not self-sufficient stations (red dot) occupancy enter the required occupancy level (black line) after repositioning (green dot).

5.3. Results and discussion
After validating the model, this section will now test the efficacy of user incentive methodologies for different scenarios and parameters. Two main methods will be considered: the one applied by Aeschbach et al. [7], M1; and the one currently used by Gira, M2. Since M1 was tested for three different WPC values, four scenarios were considered:

- M1-05: M1 with WPC=0.5, where WPC is the control station level which says to the UI method if an alternative station should be found or not. Alternative stations are presented when the initial/final station has an oc-

	Scenario 1		Scenario 2		Scenario 3		Scenario 4	
	WD	*WE*	*WD*	*WE*	*WD*	*WE*	*WD*	*WE*
(Total Trips)/Day - MAPE	0.16	0.24	0.27	1.72	8.14	3.82	4.41	3.28
(s_{out}, s_{in}) **- MAE**	0.07	0.08	0.34	0.29	4.11	1.53	4.04	1.52
(s_{out}, t) **- MAE**	0.13	0.13	0.35	0.33	0.22	0.20	0.19	0.19
(s_{out}, t) **- MVAF**	0.93	0.76	0.61	0.08	0.80	0.37	0.85	0.44
(s_{in}, t) **- MAE**	0.08	0.09	0.38	0.33	0.18	0.17	0.16	0.17
(s_{in}, t) **- MVAF**	0.97	0.87	0.52	-0.03	0.88	0.55	0.89	0.55

Table 3: Model versus simulation error metrics

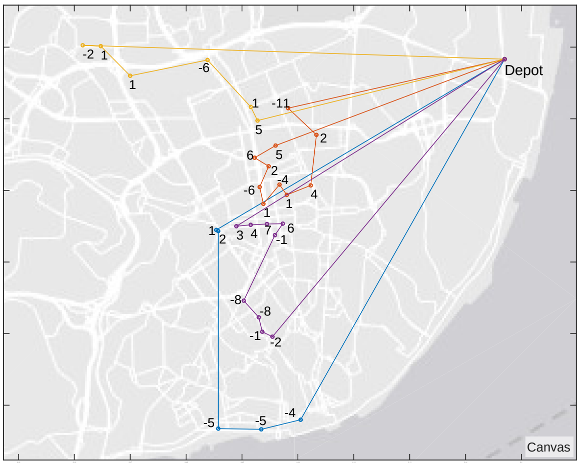

Figure 2: Example of repositioning tasks in the map with 4 trucks

cupation percentage lower/higher than 50/50 (%).

- M1-07: M1 considering WPC=0.7. This methodology was not considered by Aeschbach et al. [7] in his work and will be tested here due to its similarity to M2-07; Users are only requested to use the app if their initial/final station has an occupation percentage lower/higher than 30/70 (%).

- M1-10: M1 with preemptive control or WPC=1. Users are always requested to use the app with this method.

- M2-07: M2 presented in section 3. Unlike Aeschbach et al. [7] method, which presents as alternatives the best stations from o group (neighbourhood), this one shows nearest stations that fulfil the requirements: below/above 30/70 (%) for initial/end stations.

The above scenarios were tested applying the reasoning of Aeschbach et al. [7], varying two adjustable control parameters: CC which is the percentage of people willing to help on repositioning tasks; and the Rad used to define sets of stations. These tests were accomplished with and without static repositioning, with the last taking place during the night interval between 2 AM and 6 AM.
For each scenario, several experiments were carried out for different parameters. For the first experiments, *Rad* was varied in intervals of 0.1 km from 0.1 km to 1 km, maintaining a *CC* of 0.5 and without NR. Afterwards, *CC* levels were varied in intervals of 0.1 from 0.1 to 1, fixing Rad in 0.5 km, and also without NR. Finally, with NR activated and a *Rad* settled to 0.5 km, experiments with the following *CC* levels were carried: $CC \in \{0.25, 0.5, 0.75, 1\}$. Experiments were also taken, with UI and without repositioning for CC=1 and Rad=1, with the objective of exploring maximum ranges.
To compare results, three metrics were applied:

- Service level (SL) - A measure of service quality provided by the bike sharing system, also used in Pfrommer et al. [5];

$$SL = \frac{\#Customers - \#No\ Service}{\#Customers} \tag{5}$$

- Average extra effort (EE) (m) - Result of summing the extra meters travelled to alternative station;

$$EE = \frac{\sum_{(s_{out}, s_{in})}[d(s_{out}, \bar{s}_{out}) + d(s_{in}, \bar{s}_{in})]}{\#\ \text{Customers (with positive effort)}} \tag{6}$$

  where $d(s_{out}, \bar{s}_{out})$ is the distance between the starting station chosen by the customer (s_{out}) and the starting station chosen by the control strategy ($\bar{s}_{out}$); and $d(s_{in}, \bar{s}_{in})$ is the distance between the ending station chosen by the customer (s_{in}) and the ending station chosen by the control strategy ($\bar{s}_{in}$).

- Lost user percentage - A measure of major importance since there is no available data hold by Gira regarding people who leave the system due to empty stations.

The impact of user incentive methods on repositioning (4 trucks) was measured by:

- Visited stations per day

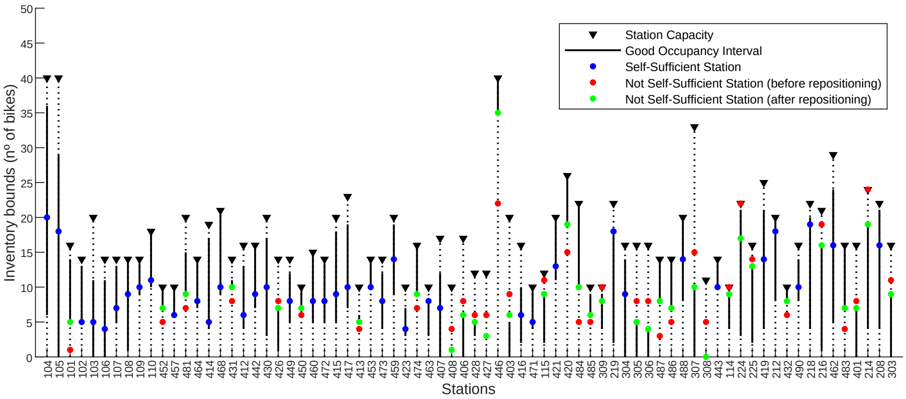

Figure 3: Example of stations' level before and after repositioning

- Bikes transported per day
- Total travelled distance per day (km)
- Sum of all operation times (minutes)
- CPU time per repositioning instructions calculus (seconds)

Experiments were performed for 28 straight days, and each case was experimented four times, therefore, the following results represent the average of four values.

Service level
BSS service level for different values of Rad is shown in figure 5, and the influence of varying CC is represented in figure 4. For both cases, experiments were carried without repositioning.
Without user incentives, a service level of 0.43 was achieved, the minimum service level found. The highest level was obtained by the M1-10 with CC=1.0 and Rad=1.0 which achieved a level of 0.99, showing that is possible to obtain a system totally balanced without repositioning, as long as all users accept to participate.
It is clear how an increase in the Rad and CC also increases service level. However, this behaviour is not the same for every scenario tested. M2-07 has the worst results, with a difference of only 0.2 when varying Rad from 0 to 1 km. M1 curves have a higher service level when compared to M2, being conspicuous that a higher WPC shows higher service level values.
Figure 6 shows the service level for different cooperation values with nightly repositioning active. As expected, service level saw a great increase for all methods, with the behaviours between them remaining unchanged. For example, M1-05 with CC=0.5 and Rad=0.5 went from 0.6 without repositioning to 0.86 with repositioning, getting a value of 0.71 when repositioning is activated and UI is inactive. These values represent a relevant increase relatively to experiments without repositioning, showing how important is rebalancing stations through trucks, and why this methodology should not be discarded, but instead cooperate with UI methods.

Average extra effort
Asking customers to change their journey implies an extra effort by cooperative users. This effort was measured by equation 6. An increase on average extra effort over Rad was identified for all scenarios, but not over CC. This was the expected outcome, given that only an increase in the size of the neighbourhood should result in higher efforts.
When it comes to the methods applied, M1 clearly implies a higher effort over M2. This is due to the fact that M2 shows as an alternative station the one closest to the intended; unlike M1 that indicates the one with the best level of occupancy. This way, unlike M2, M1 will have a higher tendency to spread bikes uniformly. Differences are only visible between M1 and M2, therefore, variations of WPC inside M1 do not seem to have an impact on this metric. The same goes for experiments with repositioning, for which values with and without repositioning appear to be the same, such as the example of (M1-05, CC=0.5, Rad=0.5, no repositioning) with a value of 0.53 km, and 0.52 km with repositioning but the same CC and Rad parameters.

Lost users
Inversely to service level, as shown by figure 8, the percentage of lost users decreases with CC and Rad. As displayed by figure 9, in this case, NR also has a huge impact, decreasing these percentages. Similarly to the case for previous metrics, M2 was the worst method, always showing higher percentages of lost users.

Repositioning results
Lastly, results related to repositioning are presented in table 4. A general analysis shows that, similar to previous metrics, M1-10 is the best, followed by M1-07, then M1-05 and finally M2-07.
The number of visited stations is reduced with the decrease of cooperation levels for all scenarios, and may be reduced to even less than 11 stations when

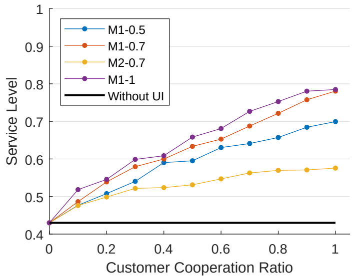

Figure 4: Service level (without NR; Rad=0.5 km)

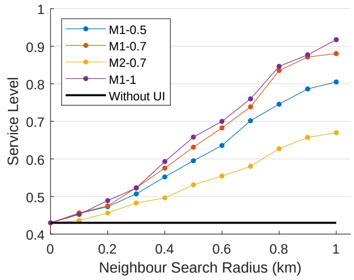

Figure 5: Service level (without NR; CC=0.5)

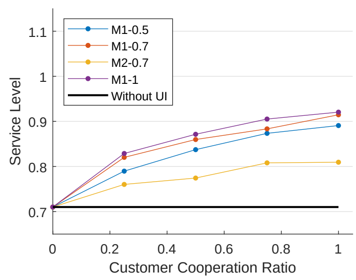

Figure 6: Service level (with NR; Rad=0.5 km)

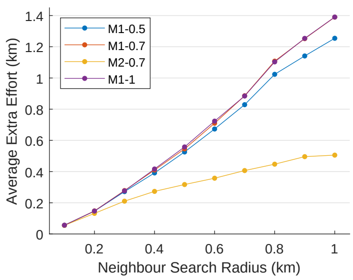

Figure 7: Average extra effort (without NR; CC=0.5)

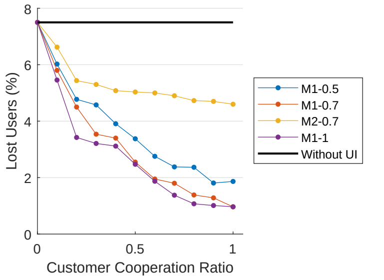

Figure 8: Percentage of lost users (without NR; Rad=0.5 km)

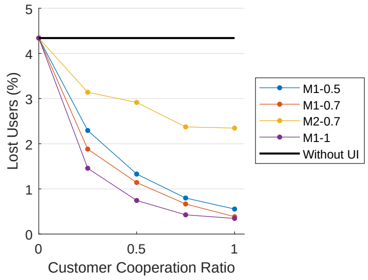

Figure 9: Percentage of lost users (with NR; Rad=0.5 km)

comparing no repositioning (42 stations) to M1-10 with CC=1 (31 stations).

Table 4 also shows that the number of bikes transported is on average 2 to 2.5 times the number of visited stations. Considering that almost half of the stations are receiving bikes and the other half losing bikes, on average, there are 4 to 5 bikes dropped or picked up per station visited.

When it comes to the total distance travelled and operation time, the behaviour is the same as previous measures, starting with 143 km for 445 minutes without UI, and dropping to values as low as 118 km travelled for 350 minutes (M1-05, CC=1), therefore, showing the influence of UIs on repositioning.

6. Conclusions

This dissertation consists of the development of a functional simulator of a Gira BSS in Anylogic. This simulation allows the study of repositioning schemes and user incentive methods, and comprises a visual interface where people, trucks and bikes can be seen travelling between stations all day. Besides observing trips on a map, the software user can change determined parameters like the number of trucks, truck's capacity, CC or Rad, as well as the methods applied.

In the simulator a nightly repositioning method (implementation of Schuijbroek et al. [4]'s work) and two user incentive methods are incorporated. The first method follows the work of Aeschbach et al. [7] and the second was originally modelled in the course of this project, in the pursue of reproducing the UI method currently applied to Gira's users.

Real data from Lisbon's BSS has validated the simulator components, such as the number of trips generated at each 20-minutes, trip time and repositioning tasks. The generation of trips was mainly validated by not considering the station's capacity or lack of bikes. Even so, when solely applying nightly repositioning and station constraints, low errors were still reached, with the number of lost

Scenario	***CC***	*Visited stations*	*Bikes transported*	*Distance traveled (km)*	*Operation time (minutes)*	*CPU time (seconds)*
No UI		42.43	106.79	143.16	445.04	44.58
M1-05	**0.25**	39.17	91.61	134.57	436.80	20.41
	0.5	37.79	85.30	133.93	422.54	31.48
	0.75	36.46	78.40	131.45	404.89	27.39
	1	35.15	69.63	127.62	382.06	15.52
M1-07	**0.25**	38.79	89.13	135.55	432.13	30.14
	0.5	36.68	79.52	132.38	409.03	28.64
	0.75	34.14	71.96	129.43	388.74	31.31
	1	31.49	61.38	121.08	353.08	23.93
M1-10	**0.25**	37.84	84.37	136.67	426.02	35.47
	0.5	35.96	78.81	130.23	403.46	24.04
	0.75	33.13	69.86	123.97	375.54	22.92
	1	31.05	60.72	118.82	350.06	10.34
M2-07	**0.25**	38.23	90.82	136.14	436.75	27.11
	0.5	37.96	89.61	136.03	435.07	38.61
	0.75	35.70	81.78	130.22	409.85	25.14
	1	35.08	80.45	129.89	406.04	24.77

Table 4: Repositioning results with 4 trucks

users being drastically reduced when compared to the case of no repositioning. Results have shown that nightly repositioning is not enough to keep the system balanced through the entire day, and only a dynamic approach could handle all the demand.
The results from varying CC and Rad with and without repositioning have shown that customers are able to balance the system without trucks, but only if all cooperate, and a Rad of 1 km is applied. Otherwise, no-service events will occur with a much higher frequency. Therefore, in case not all the users cooperate, the first conclusion to take is that truck repositioning is essential and cannot be withdrawn.
M1, presented in Aeschbach et al. [7] work, has shown evidence of a higher service level and lower lost user percentage results than M2, for all WPC ratios. Unlike M2 that only chooses stations with occupation percentages lower/higher than 30/70, M1 indicates the best station in the neighbourhood to the app user, even if the occupation does not lay in the required interval. Taking this fact into account, M1 will always have more alternative stations to present than the other method, therefore achieving better results. The only factor in which M2 showed better results was in the amount of extra effort. This was the expected behaviour considering that M2 searches for the nearest station that satisfies the requirements, while the other method seeks the best station in the neighbourhood.
Even though repositioning cannot be withdrawn, results have also shown that UI methods have reduced the average number of stations to visit per day, and the number of bikes to transport. This is, therefore, a way of reducing the transportation's costs and increasing the amount of free time to spend on other tasks.

Given the complexity of BSSs, there are still several ways to enrich this dissertation:
New studies can use this simulator to compare different UI or repositioning methods. The most obvious is to implement a dynamic repositioning method, which would be a useful study to strengthen the conclusions of this work. An alternative UI method should, for example, consider price incentives; Questions such as "How much money is necessary to get people to participate?" or "How much money will be saved on repositioning tasks compared to what is spent with UI?" are yet to be answered; In addition, effects of season, weather, events, etc., were not analysed in this thesis and could be subject to further analysis.

References

[1] Promoting environmentally sustainable transport (est). *United Nations - Sustainable Development*, (Accessed July, 2021). URL: `https://sustainabledevelopment.un.org/partnership/?p=365`.

[2] R. B. Nath and T. Rambha. Modelling methods for planning and operation of bike-sharing systems. *Journal of the Indian Institute of Science*, pages 1–26, 2019.

[3] M. A. Babagoli, T. K. Kaufman, P. Noyes, and P. E. Sheffield. Exploring the health and spatial equity implications of the new york city bike share system. *Journal of transport & health*, 13:200–209, 2019.

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

[5] J. Pfrommer, J. Warrington, G. Schildbach, and M. Morari. Dynamic vehicle redistribution and online price incentives in shared mobility systems. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1567–1578, 2014.

[6] Z. Haider, A. Nikolaev, J. E. Kang, and C. Kwon. Inventory rebalancing through pricing in public bike sharing systems. *European Journal of Operational Research*, 270(1):103–117, 2018.

[7] P. Aeschbach, X. Zhang, A. Georghiou, and J. Lygeros. Balancing bike sharing systems through customer cooperation-a case study on london's barclays cycle hire. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 4722–4727. IEEE, 2015.

[8] Sobre a gira - gira - bicicletas de lisboa. (Accessed July, 2021). URL: `https://www.gira-bicicletasdelisboa.pt/sobre-a-gira/`.

[9] M. C. Rodrigues. Planning nightly rebalancing in bike sharing systems. Master's thesis, Instituto Superior Técnico, June 2019.

[10] M. Dell'Amico, E. Hadjicostantinou, M. Iori, and S. Novellani. The bike sharing rebalancing problem: Mathematical formulations and benchmark instances. *Omega*, 45:7–19, 2014.

[11] J. Schuijbroek, R. Hampshire, and W. van Hoeve. Inventory rebalancing and vehicle routing in bike sharing systems. In *Tech. Rep. 2013-E1*. Tepper School of Business, Carnegie Mellon University, 2013.

[12] D. Chemla, F. Meunier, and R. W. Calvo. Bike sharing systems: Solving the static rebalancing problem. *Discrete Optimization*, 10(2):120–146, 2013.

[13] C. Fricker and N. Gast. Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity. *Euro journal on transportation and logistics*, 5(3):261–291, 2016.

[14] A. Fernández, H. Billhardt, S. Ossowski, and Ó. Sánchez. Bike3s: A tool for bike sharing systems simulation. *Journal of Simulation*, pages 1–17, 2020.

[15] L. Caggiani and M. Ottomanelli. A modular soft computing based method for vehicles repositioning in bike-sharing systems. *Procedia-Social and Behavioral Sciences*, 54:675–684, 2012.