Optimal Inventory Management of a Bike-Sharing Station

Tal Raviv and Ofer Kolka

September 2012

Bike-sharing systems allow people to rent a bicycle at one of many automatic rental stations scattered around the city, use them for a short journey and return them at any other station in the city. A crucial factor in the success of such a system is its ability to meet the fluctuating demand for both bicycles and vacant lockers at each station. In order to meet the demand, the inventory of each station must be reviewed regularly. This paper introduces an inventory model suited for the management of bike rental stations and a numerical solution method used to solve it. Moreover, a structural result about the convexity of the model is proved. The method may be applicable for other closed loop inventory systems. Extensive numerical study, based on real-life data is presented to demonstrate its effectiveness and efficiency.

Keywords: Shared mobility systems, Inventory management, Double-ended-queues

1. Introduction and Literature Review

Bike-sharing systems (BSSs) allow individuals to rent a bicycle at automatic rental stations scattered around a city, use them for a short journey, and finally, return them to any other station in the city. As of December 2010, some 238 cities around the world have deployed such systems and currently 53 are in planning (MetroBike LLC, 2011). For a review of the history of BSSs and on current trends, see DeMaio (2009).

Bike-sharing programs encourage residents to use bicycles as an environmentally sustainable and socially equitable mode of transportation. Such programs also serve to complement other modes of mass transit systems by mode sharing. In addition to these transportation functions, a municipal BSS may yield revenue for a city in the framework of a compliance carbon offset market, (Capoor and Ambrosi, 2009).

Modern BSSs are supported by information systems that provide data about the state of the system (i.e., the number of bicycle and lockers available at each station). This information is accessible on-line via the World Wide Web and in data kiosks at the stations. In addition, since the automatic rental stations identify the bicycles using radio frequency identification technology (RFID), operators can easily obtain detailed statistics regarding trips undertaken by users. This information can be used to support operational decisions and long-term planning.

A crucial factor for the success of a BSS is its ability to cope reliably with fluctuating demand. Indeed, the main complaints voiced by BSS users relate to the unavailability of bicycles at their point of origin and even worse, unavailability of lockers at their destinations. For example, in order to improve Brussels' bike-sharing program (Villo), a voluntary group of users created a web service that gathers inventory data from the city's BSS website in order to monitor the service level and create public pressure on the operator to improve the system. According to the group's web site (http://www.wheresmyvillo.be/), they want to make "JCDecaux [the operator of Villo] drastically improve the availability of bikes and parking spaces through better reallocation of bikes".

Meeting the demand for bicycles and vacant lockers is a particularly challenging problem due to inherent imbalances in the renting and return rates at the various stations. While the flow of commuters is approximately balanced over the course of a day, this is not the case for the flow of bicycles. This is because a BSS may be used as a partial substitute for other modes of transportation. For example, users may choose to ride a bicycle in one direction and take a bus on their way back for a variety of reasons: weather or traffic conditions, availability and frequency of the bus service at the time of their journey, topography of their route, etc. In some cases, the imbalance is persistent, e.g. relatively low return rates at stations located on the top of hills. In other cases, the imbalance is temporary e.g., suburban train stations are apt to face high return rates in the morning, as commuters into the city drop off their bikes and high rental rates in the afternoon as commuters exit the train and begin to make their way home. Satisfying user demand subject to such imbalances requires a dedicated fleet of light trucks to regularly transfer bicycles among stations. We refer to this activity as *repositioning* bicycles.

There are two repositioning modes. *Static* repositioning is performed during the night, when the system is nearly idle; *dynamic* repositioning takes place during the day in order to cope with looming shortages. In practice, many operators work in both modes.

If one ignores the stochastic nature of the demand, the static bicycle-repositioning problem can be classified as a variation of the pickup and delivery problem (PDP), a type of problem that has attracted considerable attention in recent years. Berbeglia et al. (2007) surveyed the literature on static PDP and classified these problems according to various parameters. Benchimol et al. (2010) and Chemla et al. (2011) studied a single vehicle, single commodity, pickup and delivery problem without time constraints with the goal of minimizing the total travel distance of the vehicle as it completed a prescribed repositioning task. These studies are motivated by BSSs. Chemla et al. (2011) present a branch-and-cut algorithm for solving a relaxation of the problem, from which a feasible solution is obtained via a Tabu Search. Both of these papers assume a known target inventory level for each station in the system.

Nair and Miller-Hooks (2011) used a stochastic programming approach to handle repositioning planning in shared mobility systems. Their model assumes that the cost of moving vehicles between two given stations is linear without considering the routing of the repositioning vehicles. This assumption is realistic for the one-way car-sharing systems that motivated their work. Vogel and Mattfeld (2010) presented a stylized model to assess the effect of dynamic repositioning efforts on service levels. Their model is useful for strategic planning but is not detailed enough to support repositioning operations.

Some authors consider strategic decisions regarding the capacity and locations of bike rental stations. Shu et al. (2010) proposed a stochastic network flow model to support these decisions. They used their model to design a BSS in Singapore based on demand forecasts derived from current usage of the mass transit system. Lin and Yang (2011) considered a similar problem but formulated it as a deterministic mathematical model. Their model is aware of the bike path network and mode sharing with other means of public transportation.

The static repositioning operation poses a new and challenging inventory routing problem. In this paper, we focus on the inventory management part of this problem.

Accordingly, we define a user dissatisfaction measure; model user behavior when facing shortages, and devise an efficient method of estimating the expected value of this measure given an initial inventory, station capacity, length of the replenishment cycle and the (stochastic) demand patterns.

The inventory problem associated with a bike sharing rental station can be classified as a closed-loop inventory problem. There is considerable literature on this topic. For a comprehensive review, see Dekker et al (2004). However, the available models assume that the replenishment operations by new and used items occur periodically. These models do not capture the minute-to-minute dynamics of a bike rental station where replenishment by returned items occurs at a much higher frequency than replenishment by new items.

The main contributions of this paper are: introducing a user dissatisfaction function to measure the performance of a station; presenting a dynamic inventory model of a bike sharing rental station; establishing the convexity of this function; and devising an efficient and accurate approximation method to estimate it. These results complement each other by providing practitioners and researchers with a set of tools that can be used to analyze and optimize many operational and strategic aspects of bike sharing systems. For some examples see the discussion in Section 6.

The work of Nair and Miller-Hooks (2011) is close to this work in the sense that it also models the stochastic nature of the demand in a vehicle sharing system. In their study, the shortage is modeled as the difference between the total supply and the total demand over the planning horizon but without considering the sequence of the events.

As an example, consider a station with a deterministic demand of 50 bicycles (rentals) per day and 40 lockers (returns) per day and assume that the inventory of the station is reviewed every night. According to the model of Nair and Miller-Hooks (2011) an initial inventory of 10 bicycles is sufficient to provide perfect service at the station. However, if most of the rentals are expected during the morning hours while most of the returns are expected during the afternoon, the station will face a high level of bicycle shortage during the morning.

Our hands on experience with demand data from bike sharing systems shows that such extremely non-homogenous demand processes are common and in fact observed in most of the stations. Therefore, this study presents a model that tracks the inventory level in the station continuously throughout the planning horizon.

As our model focuses on a single station, we omitted the interaction between the demand processes of the stations. However, we note that such interactions occur when users fail to realize their demand for bicycles or lockers at the desirable origin or destination. Thus, in a well-operated system, the influence of these interactions on the optimal inventory level of each station is negligible. In section 5, we report on a simulation study conducted using historical data from operating bike sharing systems to demonstrate the usability of our model in supporting better decision making in real systems.

The inventory model devised in this study is useful for supporting operational tasks such as inventory routing of bicycles among stations, see Raviv, Tzur and Forma (2011). It can also be used for strategic decision making regarding the system-wide inventory of bicycles as well as station size decisions (discussed in greater detail in the conclusion section). In addition we believe that our approach may be applicable for other closed-loop inventory systems.

The bike-sharing station is modeled in this study as a double-ended queuing system introduced by Kashyap (1966). It is similar to a taxi queue at an airport or train station: there are either taxis waiting for people (equivalent of bicycles) or people waiting for taxis (equivalent of renters). Only one side is queued at a time. A steady state analysis of such a system and its variants are discussed in the queuing theory literature. For a discussion and references, see Srivastava and Kashyap (1982). For more recent studies considering the effect of user impatience, see Conollya, Parthasarathyb, and Selvarajub (2002). Mendoza, Sedaghat, and Yoon (2009) used the double-ended-queue model to study a model that aims to balance production rate (supply) and demand. However, none of these studies considered controlling this queuing system by modifying its state, i.e., exogenously adding or removing items from the queues. Consequently, none of these studies analyzed the transient dynamics of this model that are required to describe a bike sharing station that faces a non-homogenous demand process and the inventory of which, is revised periodically.

Notation and a formal definition of the model are presented in Section 2. The convexity of the user dissatisfaction measure is proved in Section 3. In Section 4, a method to solve the model approximately under the assumption that the arrival processes are Poisson is devised. The results of numerical experiments that demonstrate the effectiveness and robustness of the approximation are presented in Section 5. In this section we also report on a simulation study conducted ahead of the implementation of the method presented in this study in a real bike-sharing system. We conclude with a discussion in Section 6 about possible applications of the approximation method and convexity property presented in this paper and some ideas for future research. In Appendix A, we describe a simulation based optimization procedure used in the simulation study of section 5 and in Appendix B we present an extension of our inventory model that allows customers to queue (or backlog) in the system. In Appendix C, we present the detailed results of our numerical experiment.

2. Notation and Problem Formulation

We consider a single bike-sharing station over a finite horizon $[0, \mathrm{T}]$ with the following settings: at time 0 the inventory level (number of bicycles) in the station is set. During the planning horizon, users that wish to rent or return bicycles arrive at the station according to arbitrary stochastic processes[1]. If the desired service can be provided right away, the bicycle is rented or returned and the inventory level is updated. If the service cannot be provided (i.e., empty station for a renter or full station for a bicycle returned) the user abandons the station immediately. In the case of renters, abandonment represents a decision of the user to either forgo using the system at this time or seek an available bicycle at a different station. In the case of a returner, the bicycle must be returned to another station. In both cases, we neglect the effect of interactions between stations in the system as we focus on a single station. We assume that the inventory of the station is not reviewed by a repositioning operation before time T. In Appendix B, we also discuss a

[1] The computational method introduced in Section 4 assumes a non-homogenous Poisson arrival process but the robustness of the model with respect to this assumption is demonstrated in Section 5. Moreover, some structural results derived in Section 3 are independent of the nature of the arrival process.

more general inventory model where the users may queue in the warehouse when it is empty (for renters) or full (for returners).

There are two sources of user dissatisfaction in the system according to this model and the system is penalized for each of them.

p Penalty charged for each potential renter who abandons due to a shortage of bicycles.

h Penalty charged for each returner who abandons due to a shortage of vacant lockers.

The total number of lockers in a station is denoted by C. We refer to this value as the *station capacity*. Next, we present a notation that describes a particular realization of the demand over the given horizon. Let E^R denote the set of epochs in which a demand for a bicycle occurs in given realization R where E_i^R denotes the i^{th} occurrence. I_0 is a decision variable that represents the initial bicycle inventory of the station as set by the operator at the beginning of the planning horizon. The state of the system at time t, is denoted by $I_t^R(I_0)$ where $I_t^R(I_0) \in \{0, \dots, C\}$ represents the number of bicycles currently in the station under a given realization R, and initial inventory I_0. Hereafter the realization and initial inventory arguments will be omitted for the sake of conciseness where they are apparent from the context. We denote the net demand for bicycles at time epoch t by $d_t^R \in \{-1,0,1\}$ where negative values represent a demand for lockers, i.e., users that wish to return bicycles at the station. Clearly $d_t^R = 0$ for all $t \notin E^R$. The implicit assumptions that at each demand point only one bicycle is rented or returned causes no loss of generality since the time between epochs can be arbitrarily short.

The dynamics of the station in a given demand realization is given by:

$$I_{E_i^R} = \begin{cases} 0 & I_{E_{i-1}^R} - d_{E_i^R}^R < 0 \\ C & I_{E_{i-1}^R} - d_{E_i^R}^R > C \\ I_{E_{i-1}^R} - d_{E_i^R}^R & otherwise \end{cases}$$

The number of renters that abandon the station at time epoch E_i^R is $\left(-I_{E_{i-1}^R} - d_{E_i^R}^R\right)^+$ and the number of returners that abandon is $\left(I_{E_{i-1}^R} - d_{E_i^R}^R - C\right)^+$, where $(x)^+ \equiv \max(0, x)$. The *user dissatisfaction function* (UDF) is the expected penalty due to the abandonments of returners and renters as a function of the initial inventory I_0.

$$F(I_0) \equiv \mathbb{E}_R \left\{\sum_{i=1}^{|E^R|} \left(p\left(-I_{E_{i-1}^R} - d_{E_i^R}^R\right)^+ + h\left(I_{E_{i-1}^R} - d_{E_i^R}^R - C\right)^+\right)\right\} \tag{1}$$

The bike sharing station inventory problem is to select the optimal value of $I_0 \in \{0, \dots C\}$ so as to minimize $F(I_0)$.

We point out that the value of the UDF is of interest for all initial inventory levels I_0 and not only for the one that minimizes the function. This is due to the fact that in many cases it is impossible or too costly to set the initial inventory levels of all stations at their ideal values. In order to understand the trade-off between the number of bicycles in the various stations, the cost of repositioning bicycles and the service level, a method to efficiently calculate the UDF for all the stations in the system explicitly is required.

3. Convexity of the User Dissatisfaction Function

Recall that the user dissatisfaction function (1) expresses our measure for the expected penalty caused by a shortage of bicycles and lockers at a station. In this section, we prove that this function is convex with respect to the initial inventory I_0.

The convexity of the UDF is of interest mainly since it allows optimizing the total UDF over all the stations in the system subject to some complicating constraints. This is necessary in order to solve system-wide operational and strategic optimization problems. For discussion of possible applications of such an optimization problem, see Section 6. The convexity is also an indispensable property for optimization of the UDF by bisection when its evaluation for each possible initial inventory is impossible or too costly. For example, when the Markovian assumptions underlying the approximation procedure presented in Section 4 are violated and the only viable alternative is discrete event simulation.

Let us define the marginal dissatisfaction function from additional units of initial inventory as

$$F'(I_0) \equiv F(I_0) - F(I_0 - 1).$$

Clearly, this marginal value function is a discrete counterpart of the differential in a continuous function. Equivalently, if $F'(I_0)$ is a non-decreasing function, then $F(I_0)$ is said to be convex.

Theorem 1: The UDF, $F(I_0)$, is a convex function of the initial inventory I_0.

Proof: The outline of our proof is as follows: first, note that since the expected user dissatisfaction is obtained as a weighted sum of the user dissatisfaction over all possible realizations, it is enough to show that the total user dissatisfaction is convex for any particular realization. Next note that the total user dissatisfaction in a given demand realization is a weighted sum of two components, namely, the number of potential renters and the number of potential returners that abandon the station. Therefore, it is enough to show that each one of these components is a convex function.

For a fixed realization R, let us denote these two components by $f_1(I_0), f_2(I_0)$ respectively. Consider the state of two hypothetical systems under the same demand realization that differ in their initial inventory level by one unit. Let us follow the process $I_t(I_0 - 1)$ and $I_t(I_0)$. If $I_t(I_0) > 0$ and $I_t(I_0 - 1) < C$ for all $t \in [0, T]$ then, there are no abandonees in either systems and thus $I_t(I_0) - I_t(I_0 - 1) = 1$ for all t. However, if at some time t' abandonment occurs in one of the processes then from this time on, the two processes coincide, that is $I_t(I_0) = I_t(I_0 - 1)$ for all $t \geq t'$. Consequently, the difference between the numbers of abandonees in both systems is at most one. We say that the process $I_t(I_0)$ "hits" some value k at time t' if the value of the process is changed to k at time t'. For a given scenario there are three alternatives:

1. If $I_t(x)$ hits 0 before $I_t(I_0 - 1)$ hits C for the first time, then $f_1'(I_0) = f_1'(I_0 - 1) - f_1'(I_0) = -1$ and $f_2'(I_0) = f_2'(I_0 - 1) - f_2'(I_0) = 0$.
2. If $I_t(I_0 - 1)$ hits C before $I_t(I_0)$ hits 0 for the first time, then $f_1'(I_0) = 0$ and $f_2'(I_0) = 1$.
3. If $I_t(I_0)$ never hits 0 and $I_\theta(I_0 - 1)$ never hits C, then $f_1'(I_0) = 0$ and $f_2'(I_0) = 0$.

Next, observe that $f_1'(I_0) = -1 \rightarrow f_1'(I_0 - 1) = -1$. This is due to the fact that $I_t(I_0 - 2)$ always hits 0 before $I_t(I_0 - 1)$ hits it. Note that it also possible that $f_1'(I_0) = 0$ and $f_1'(I_0 - 1) = -1$ or $f_1'(I_0) = f_1'(I_0 - 1) = 0$. That is, $f_1'(I_0)$ is a non-decreasing function and so $f_1(I_0)$ is convex. Similarly $f_2'(I_0-1)=1 \rightarrow f_2'(I_0)=1$ and thus $f_2(I_0)$ is convex. **Q.E.D.**

4. Approximation of the User Dissatisfaction Function

In this section we present a method for calculating the UDF under some sound modeling assumptions as to the nature of the stochastic process that governs the demand at each rental station. In particular, we assume that the arrival processes of renters and returners are non-homogenous Poisson processes with rates denoted by μ_t and λ_t respectively.

The state of the station can be viewed as a non-time-homogenous birth and death process depicted as a Markov chain as seen in Figure 1. The birth rate is λ_t and the death rate is μ_t.

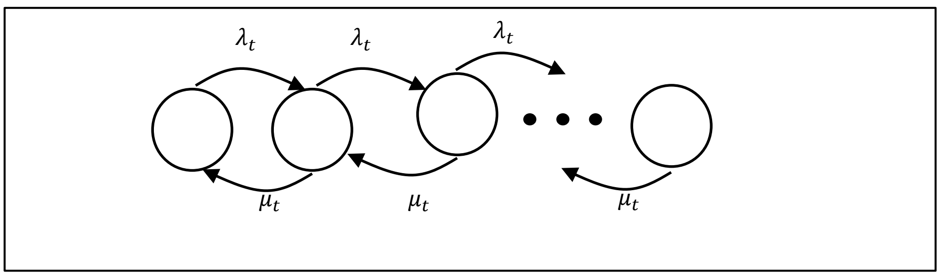

Figure 1: Continuous time Markov chain that represents the dynamics of the bicycles' inventory level

Note that since the process I_t is controlled by setting its initial state, we are interested in the transient dynamics of the process rather than in its steady state. Let $\pi_{ij}(t)$ denote the probability of the station being at state j at time t given that its initial state at time 0 was i. We use the notation $\boldsymbol{\pi}(t)$ to refer to the whole transition probability matrix. For a Poisson demand process, it is possible to state the UDF (1), in terms of the transition probabilities as follows:

$$F(I_0) = \int_0^T \left(\pi_{I_0,0}(t)\mu_t p + \pi_{I_0,C}(t)\lambda_t h\right) dt. \tag{2}$$

The first term in the integral represents the expected user dissatisfaction accumulated when the station is empty. During such a period, abandonments occur at a rate of μ_t and each bears a penalty of p. The second term represents the expected user dissatisfaction that accumulates when there are no vacant lockers at the station. During such a period of time, abandonments occur at a rate of λ_t and each bears a penalty of h.

The computational challenge in the evaluation of (2) is the calculation of the transition matrix $\boldsymbol{\pi}(t)$. There is no closed form solution for the dynamics of this system. In addition, it is important to note that no insight can be gained from a steady state analysis of some simplified version of the system, such as one with a homogenous arrival process. This is mainly since a properly operated station never approaches a steady state. Instead, the operator regulates the inventory level and periodically sets it at levels that may be far from the steady state mean. Only if the station is left untouched will it deteriorate to its steady state. Moreover, unlike other service systems such as contact centers, bike-sharing stations typically operate under low arrival rates. That is, the frequency in which the demand intensity is changed is high relative to the frequency of events in the system.

Next, we present a procedure to estimate (2) using a discretization of the Markov chain in Figure 1. In the next section, we validate this procedure by comparing it to a simulation that requires numerous replications and is thus much more demanding computationally. We note that while the arrival rates of renters and returners is non-homogenous over time, it is reasonable to assume that these rates change in a finite number of steps over a planning horizon, say every 15 minutes. This modeling assumption can be justified merely by the fact that it is typically impossible to estimate the arrival rates reliably for shorter periods.

We discretize the planning horizon into short periods of length δ. For each such period θ, we evaluate the transition probability matrix from the beginning of the period until its end. We denote this transition probability by P_θ. The transition probability matrix from time 0 to time $\theta\delta$ may be given as

$$\boldsymbol{\pi}(\theta\delta) = \prod_{\theta\prime=1}^{\theta} P_{\theta\prime}.$$

Moreover, using the recursive relation $\boldsymbol{\pi}(t) = \boldsymbol{\pi}(t-\delta) \cdot P_{t/\delta}$ one can calculate all the transition matrices $\boldsymbol{\pi}(t)$ for $t = \delta, 2\delta, \dots, T$ in only $\frac{T}{\delta}$ matrix multiplication operations once the values of P_θ are estimated. Recall that since the arrival rates are constant during each period θ, then $P_\theta = e^{R_\theta \delta}$ where R_θ is the transition rate matrix of the Markov chain in Figure 1 with transition rates that correspond to the time$[(\theta-1)\delta, \theta\delta]$. An approximation of $e^{R\delta}$ can be obtained by the identity

$$e^R = \lim_{n\to\infty}\left(\mathrm{I}+\frac{R}{n}\right)^n.$$

Where I is the identity matrix and the power operation corresponds to the matrix multiplications. The limit can now be well approximated by calculating $\left(I+\frac{R}{n}\right)^n$ for some large n. [see Ross (2010), Section 6.8 for example].

Based on the approximate value of $\pi(t)$, we can obtain a reliable approximation of the user dissatisfaction function using the following discretization procedure

$$F(I_0) \approx \delta \sum_{\theta=0}^{\frac{T}{\delta}-1}\left(\pi_{I_0,0}\big((\theta+0.5)\delta\big)p\mu_\theta + \pi_{I_0,C}\big((\theta+0.5)\delta\big)h\lambda_\theta\right). \quad (3)$$

Note that we use the transition probability to the midpoint, $(\theta+0.5)\delta$, of each discretized period as it better represents the typical state of the station during the period θ. The approximation procedure is parameterized by δ. Clearly, as $\delta \to 0$ the approximate value approaches $F(x)$ but the computational effort increases. The approximation error can be bounded as demonstrated below

$$F(I_0) \geq \delta \sum_{\theta=0}^{\frac{T}{\delta}-1}\Big(\min\Big(\pi_{I_0,0}(\theta\delta), \pi_{I_0,0}\big((\theta+1)\delta\big)\Big)p\mu_\theta + \min\Big(\pi_{I_0,C}(\theta\delta), \pi_{I_0,C}\big((\theta+1)\delta\big)\Big)h\lambda_\theta\Big). \quad (4)$$

$$F(I_0) \leq \delta \sum_{\theta=0}^{\frac{T}{\delta}-1}\Big(\max\Big(\pi_{I_0,0}(\theta\delta), \pi_{I_0,0}\big((\theta+1)\delta\big)\Big)p\mu_\theta + \max\Big(\pi_{I_0,C}(\theta\delta), \pi_{I_0,C}\big((\theta+1)\delta\big)\Big)h\lambda_\theta\Big). \quad (5)$$

Inequalities (4) and (5) are valid due to the fact that during each period θ, the accumulated expected number of abandonments is monotonous, either non-increasing or non-decreasing depending on the relation between μ_θ and λ_θ. If the renter arrival rate is greater than the returner arrival rate, $\mu_\theta > \lambda_\theta$, the probability of the station being empty, $\pi_{I_0,0}(t)$, increases during the period while the probability of the station being full, $\pi_{I_0,C}(t)$ decreases. Calculating the number of abandonments based on the lower between the probability of the station being empty (respectively, full) at the beginning of the period and at the probability of the station being empty (respectively, full) at the end of the period, $\min\left(\pi_{I_0,0}(\theta\delta), \pi_{I_0,0}\left((\theta+1)\delta\right)\right)$ [respectively, $\min\left(\pi_{I_0,C}(\theta\delta), \pi_{I_0,C}\left((\theta+1)\delta\right)\right)$], results in a lower bound while using the higher probability results in an upper bound.

In Figure 2 below, we illustrate the effectiveness of the bounds described above. The UDF presented in the figure is for a 20-hour period (6am-2am) at one of the busiest stations of Capital Bikeshare in Washington DC, a station with 15 lockers. The arrival rate of users was estimated based on data collected in the station during weekdays in the winter of 2010-2011. The average number of arrivals and returners per day during this period was 60.88 and 65.12 respectively. The computation was based on time discretization of 15 and 5 minutes. The penalty per abandoment of both types was set to one $(p = h = 1)$, accordingly, the y-axis represents the expected number of user abandoments per day. It is apparent from the graphs that while the approximation error can be bounded much more tightly with the finer discretization, both discretization levels produces very similar approximated UDFs.

One interesting observation from the graphs is that although the station faces, on average, a similar number of renters and returners, it is best to set the initial inventory of the station to be almost empty. This is due to the fact that most of the returners, in this particular station, arrive in the morning while most of the renters arrive in the afternoon, as can be observed in Figure 3. Therefore, in order to reduce the chance that returners will arrive at a full station it is better to start with few bicycles at the station. During the morning hours the number of bicycles in the station will gradually increase due to a higher rate of returns than rentals and the station will be ready for the high rental rate expected during the afternoon. This is a typical situation for many stations that are located in city centers.

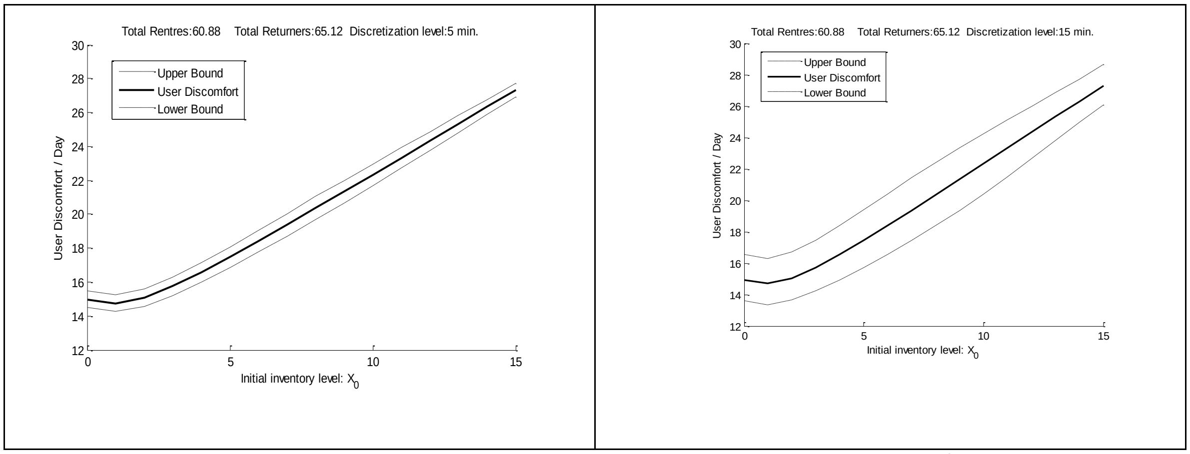

Figure 2: UDF and bounds, assuming users with no patience with discretization levels of 5 and 15 minutes

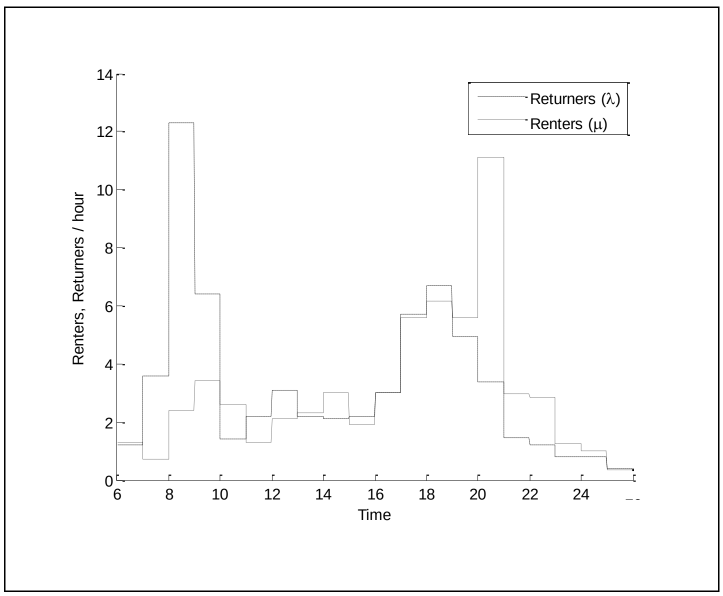

Figure 3: Renters and returners arrival rate at a station

Note that the computationally demanding part of (3)-(5) is the approximation of $\pi(t)$. This calculation requires numerous matrix multiplications that can be implemented very efficiently and effortlessly using specialized mathematical packages such as Matlab™ or the open-source package Octave. Indeed, producing the data required for drawing the figures above takes a small fraction of a second on a modest desktop.

5. Numerical Study

The goal of the numerical study in this section is to demonstrate the accuracy, efficiency and effectiveness of the approximation method presented in Section 4 in supporting inventory decisions in a real bike sharing system. The accuracy is demonstrated by comparing the obtained results to the lower and upper bounds presented in Section 4. The efficiency and effectiveness of our method is demonstrated via comparison with an alternative simulation-based optimization method (presented in Appendix A), and with the solution used by the expert planner before the method was adopted.

Our study is based on demand data collected in Tel-O-Fun (www.tel-o-fun.co.il), the bicycle sharing system operator in Tel Aviv, Israel. As of February 2012, the Tel-O-Fun system consisted of 129 stations, 2542 lockers, and about 1000 bicycles. The average demand faced by the system as of February 2012 was approximately 5000 rides per day, on regular weekdays. The demand patterns in most of the stations during this period did not change significantly, except for a uniform reduction in the volume during several particularly rainy days.

The demand data was collected during 39 regular working days between December 2011 and February 2012. We focus on the 82 busiest stations that faced almost all the demand in the system. Stations with an average demand of less than ten bicycles a day were ignored, since for these stations there was not enough data to obtain a reliable estimation of the demand process. Initially, the penalty cost associated with abandonment was set to $p = h = 1$\$ per occurrence. Later we demonstrate the robustness of our model with regard to these parameters.

In order to test the efficiency and accuracy of the approximation method, we applied it to the 82 stations with three different discretization levels ($\delta = 1\text{ min.}, 5\text{ min.}, 30\text{ min.}$). Each of these instances was solved with our method and compared to the lower and upper bounds, (4) and (5). The code for these calculations was written in MathWorks Matlab™. The code as well as the input data of our experiment is available from the first author upon request. All the experiments were run on an Intel Core i7 ™ desktop under Windows 7 ™ 64 bits. For each instance we collected the following information:

- The computation time (in seconds) including the calculation of the bounds (*CPU time)*
- The maximum relative difference between either the lower or upper bounds over all possible initial inventory levels (*Max Error*)
- The average relative difference between the bounds divided by two (*Average Error*)

- For $\delta = 5,$ and $30,$ the maximum relative difference between the values approximated by the corresponding discretization levels and value approximated with $\delta = 1$ (*Max Diff.*)

Let us denote an approximated UDF using a discretization level of δ minutes by $F^{\delta}(I_0)$. Its lower bound as obtained with the same discretization level is denoted by $LB^{\delta}(I_0)$ and the upper bound by $UB^{\delta}(I_0)$. The above quantities can be formally defined as follows:

$$Max\ Error(\delta) = \max\left\{\max_{I_0 \in \{0,\dots,C\}}\left(\frac{UB^{\delta}(I_0) - F^{\delta}(I_0)}{F^{\delta}(I_0)}\right), \max_{I_0 \in \{0,\dots,C\}}\left(\frac{F^{\delta}(I_0) - LB^{\delta}(I_0)}{F^{\delta}(I_0)}\right)\right\}$$

$$Average\ Error(\delta) = \frac{\sum_{I_0=0}^{C}\left[UB^{\delta}(I_0) - LB^{\delta}(I_0)\right]/F^{\delta}(I_0)}{2(C+1)}$$

$$Max\ Diff(\delta) = \max_{I_0 \in \{0,\dots,C\}}\left|\frac{F^{\delta}(I_0) - F^{1}(I_0)}{F^{1}(I_0)}\right|$$

In Table 1, we present summary statistics for the 82×3 tested instances. The first column of the table corresponds to the discretization level. The rest of the columns are divided into four groups. The first group presents statistics on the running time of the approximation algorithm (including the time needed to calculate the bounds); the second presents statistics on the maximal error; the third presents statistics on the average error and the last on the maximal difference. For each measure both the average and the maximum statistics over the 82 instances are reported. Detailed results for each instance are available in Appendix C, Table 2.

Discretization Level	CPU time (sec.)		Max Error		Average Error		Max Difference	
	Average	Max.	Average	Max.	Average	Max.	Average	Max.
$\delta = 1$ min.	0.264	0.297	0.48%	0.84%	0.40%	0.71%	-	-
$\delta = 5$ min.	0.059	0.069	2.46%	4.37%	1.98%	3.53%	0.28%	2.48%
$\delta = 30$ min.	0.016	0.022	16.30%	30.29%	11.85%	21.29%	7.17%	38.46%

Table 1: Summary of accuracy test of the UDF approximation method

It is apparent from Table 1 that all the test instances could be approximated very quickly by the proposed method. Although the solution time increased with discretization level, even the hardest instance, corresponding to a station with 29 lockers and time discretization of one minute, could be solved in 0.297 seconds. The average solution time with the five minute discretization is about 500 times shorter than the time needed to obtain results with equivalent accuracy via simulation. This allows our method to be used for on-line decisions in large systems and as a subroutine in algorithms that solve system-wide integrated models.

The approximation accuracy seems to improve significantly, as the discretization level is increased. The maximal error for one-minute discretization was on average about a fifth (resp., 1/30) of the one obtained by a five-minute (resp., 30 minute) discretization. The solution time is on average about five times (resp., sixteen times) longer. However, when comparing $\delta = 5$ with $\delta = 1$ the accuracy improvement is mainly due to the tightening of the bounds rather than actual changes in the approximated values. Indeed, the average (resp., maximum) relative difference between the one and the five minute discretization levels is, merely 0.28% (resp., 2.48%). Moreover, the optimal initial inventory level for all three discretization levels is the same in all the 82 tested stations.

We conclude that the method can provide a fairly accurate approximation of the UDF even with five-minute time discretization. If the goal is only to find an optimal initial inventory for each station separately, it seems that even rougher time discretization can be used safety. Therefore, if it is necessary to solve numerous instances of the problem quickly, it is recommended to use rougher time discretization in order to save time. Furthermore, it is unlikely that further refinement of the discretization will lead to a significantly more accurate approximation.

A crucial parameter for determining the optimal initial inventory level is the ratio between the penalties for bicycle shortage and locker shortage (namely p and h). Note however, that a proportional change in the values of these parameters does not change the optimal solution but rather only scales its value. Therefore, we fixed the bicycle shortage penalty and checked the effects of changes in the locker shortage penalty on the optimal solution and its value.

In Figure 4, we plot the optimal initial inventory for the six busiest stations in Tel-O-Fun against the value of h in the interval $[0.5,2]$ when $p = 1$. This allows us to examine how changes in the ratio h/p affect the optimal decision. Naturally, the optimal initial inventory is non-increasing in this ratio since higher penalties for shortage of lockers implies that we should avoid this kind of shortage at the expense of taking the risk of a higher shortage of bicycles. Therefore, a smaller initial inventory level is desired. However, it appears that the optimal inventory changes only slightly as the ratio increases from 0.5 to 2. The sharpest change in the initial inventory over the tested range is in Station 1, where the optimal initial inventory decreases from 24 to 20 over the entire range.

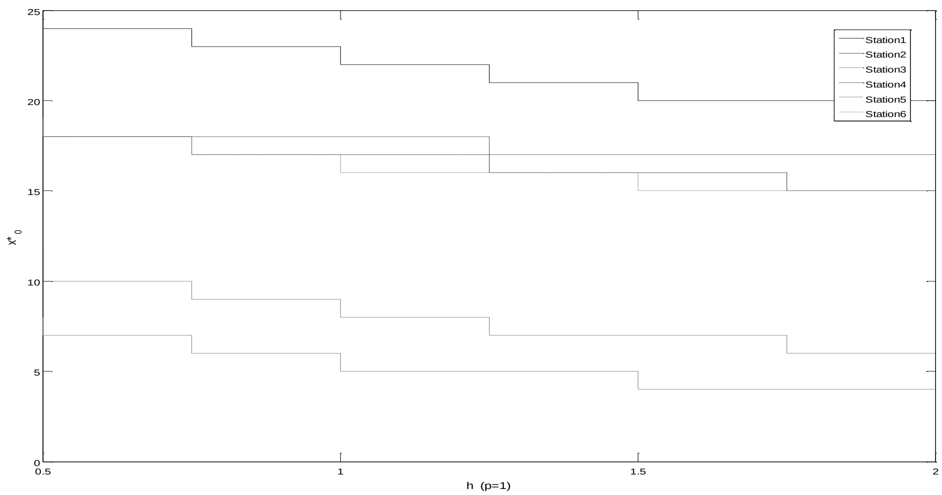

Figure 4: Sensitivity analysis for penalty parameters

In order to deepen our understanding regarding the sensitivity of the optimal inventory to the h/p ratio we checked the loss, in terms of the objective function value, when using the optimal inventory level assuming $h/p = 1$ while the actual ratio is different. In

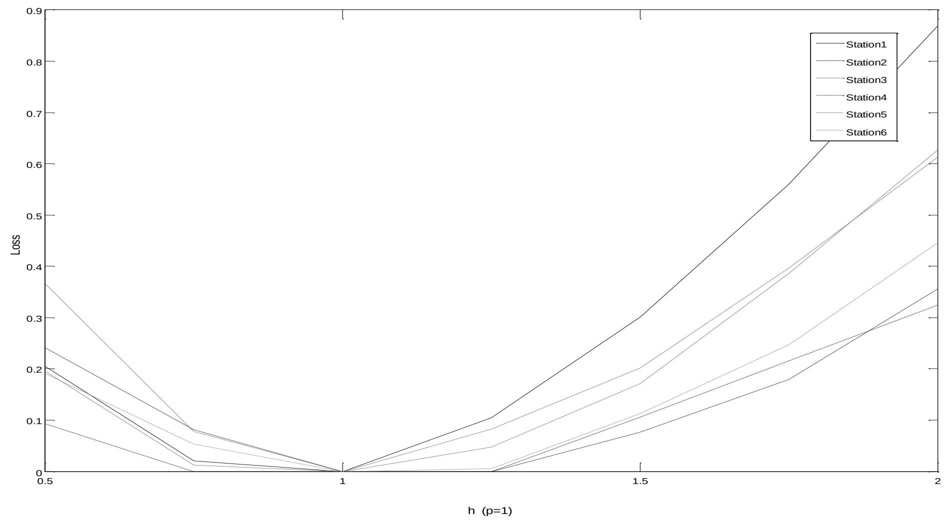

Figure **5**, we plot this loss against the h/p ratio for the same six stations. The value on the vertical axis represents the difference between the optimal value of the objective function for any given ratio and the value of the objective function with the same ratio when using the optimal inventory level obtained for $h/p = 1$. One can observe that in the worst case (again, Station 1 for $h = 2$) less than 0.9 units are lost. This corresponds to an expected bicycle shortage of about 0.9 units or an expected locker shortage of about 0.45 units. We conclude that the inventory model presented in this paper is robust to inaccuracies in the values of its shortage penalty parameters.

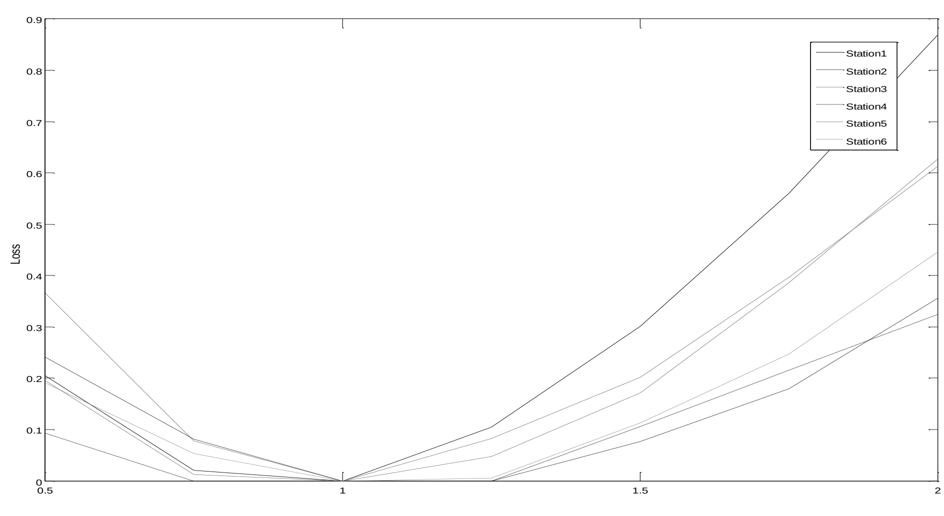

Figure 5: Optimality loss due to inaccuracy in the h/p ratio

In the rest of this section, we report on a simulation study performed ahead of the implementation of the inventory model devised in this paper in Tel-O-Fun, The goal of this simulation study is to demonstrate the usability of the model in supporting better decision making regarding initial inventory levels to be set every night. Consequently, the company adopted the method in order to aid in the planning process of the repositioning operation carried out during the night.

The main goal of the simulation study is to check the sensitivity of the results obtained from our method with respect to the assumption that the demand processes at the various stations in the system are independent. As discussed above, this simplifying assumption is undoubtedly inaccurate. However, we show that even with this assumption the model prescribes good decisions regarding the initial inventory.

The chief planner of Tel-O-Fun devised the following procedure for planning the repositioning of bicycles during the off-peak hours of the night: first, a target level for each station is defined; then the city is divided into four operational districts and one driver is assigned to each district. The night shift drivers spend their shift attempting to bring the inventory level of each station to its target value by moving bicycles among the stations in their own districts or from/to the depot. The previous practice was that the target levels were updated periodically by the planner in a trial and error procedure. The initial inventory of stations that tended to run out of bicycles was gradually increased while the initial inventory of the stations that tended to run out of lockers was gradually decreased.

While the capacity of many of the stations in the system is insufficient to avoid using static repositioning alone (even when the initial inventory levels are selected correctly), the dynamic repositioning currently eliminates most of the shortages and enables meeting nearly 100% of the demand. Consequently, the demand data collected by the system represents the true demands for rides fairly well.

The main goal of the static repositioning in Tel-O-Fun is to reduce the amount of repositioning work to be done during the day. Therefore, we evaluated the quality of a solution by counting the number of bicycles that needed to be added or removed from the stations of the system during the next day, assuming the initial bicycle inventory level prescribed by this solution. Equivalently, it is possible to count the number of shortages (of bicycles and lockers) assuming no repositioning is done during the day.

In order to benchmark our method we simulated the system with three different settings of initial inventories. Namely, 1) The *expert solution* of the planner used during these particular days; 2) The *model solution* obtained by our method based on a forecast created using demand data collected during the previous 39 working days in January and December; 3) A solution obtained by a *simulation-based optimization* algorithm described in Appendix A. This algorithm is much more computationally demanding compared to our model. The simulation was based on actual demand data collected during the next five working days immediately subsequent to the period of the "training data". In the simulated system, no dynamic repositioning was performed; accordingly, the total number of abandonments approximately represents the extent of dynamic repositioning work needed in the system. We also recorded the time of the first abandonment in each station and counted the number of stations that faced no shortages during the day.

Over the five day test period, the number of shortages under the expert solution was 1953, system wide, while the one obtained under the model solution was 1604, a savings of more than 17% in the loading/unloading work left for the dynamic repositioning vehicles. Moreover, while in the expert solution, 61% of the stations should be visited by a repositioning vehicle during the day (on average), the model solution requires a visit in less than 51% of stations every day. That is, the dynamic repositioning vehicles can reduce their travel distance. This is of particular importance because the travel cost in the city during the day is significantly higher compared to nighttime.

The simulation based optimization algorithm delivers a solution that is slightly inferior to the one obtained by our model. It yielded 1637 shortages during the five days test period and visits to 52.7% stations every day. In terms of computational effort, the simulation based procedure for all the 82 tested stations took more than half an hour, compared to a running time of less than one second of the Matlab implementation of our model with time discretization of $\delta = 30$ minutes. Recall that the same optimal initial inventories were obtained also with finer time discretization, $\delta = 1,5$.

Indeed, the savings obtained by our model compared to the simulation-based optimization (about 2% less shortages) is not spectacular and the computation time of the simulation-based optimization is not prohibitive for such a medium range operational problem. However, it is nice to have a more accurate and more efficient computational method for this task especially if one wishes to integrate the model in a decision support system with 'what-if' capabilities. Moreover, if the inventory model is to be integrated as a subroutine of a more general algorithm, such as one that solves the repositioning (routing) problem, its efficiency may be crucial. An important conclusion from the results of this simulation study is that the assumptions, on which our method is based, are at least useful, if not accurate.

6. Discussion and Conclusions

The inventory model presented in this paper can be used by operators of BSSs in their daily operations while rebalancing their systems. Moreover, the convexity property of the presented model together with the efficient approximation method make the results of this study very useful for various operational and strategic decisions that should be made by the BSS operators. In particular, the inventory model may serve as a building block for solving the following optimization problems:

Optimal Global Bicycle inventory: The decision about the optimal number of bicycles in the system is a crucial tactical decision that should be reviewed by the operator whenever demand patterns or station capacity change. Clearly, if the global inventory is too low, the renter is more likely to face bicycle shortages; if it is too high, the returners are apt to face a shortage of lockers. A rational inventory level to start with is equal to the total initial inventory in the stations of the system.

System Wide Inventory allocation: In some cases the total number of bicycles that are available for the operator is limited. We believe that in the long run this constraint should be relaxed by ordering more bicycles, since the cost of the bicycles in a bike sharing system is small compared to the infrastructure cost. However, it is clear that in the short run, system wide shortage of bicycles is unavoidable. Therefore, the operator may need to determine initial inventory simultaneously at the stations subject to global inventory constraints. This problem can be solved using the approximation method presented in this paper and thanks to the convexity property. Optimal allocation is obtained by allocating the bicycles one by one in decreasing marginal value of the UDF (across all stations in the system).

Repositioning: Preliminary results of this study were used by Raviv, Tzur and Forma (2012) to formulate a mixed integer linear model for routing inventory in order to rebalance stations using a fleet of vehicles subject to capacity and time constraints. The sum of the UDFs of all stations is incorporated into the objective function of the repositioning model rather then used to prescribe target inventory levels. This approach takes advantage of the fact that the UDF is typically flat around the optimal level and thus in many cases it is not worthwhile to visit a station that is off but near its optimal inventory level.

We believe that the single station inventory model presented in this paper may be very useful for BSS operators. However, like any mathematical model, it is based on some simplified assumptions. In particular, the model presented here omits complex interdependencies among the different stations. In fact, we see two types of such interdependencies. First, the demand process at each station may be affected by the state of its neighboring stations since unsatisfied renters and returners are likely to seek service in neighboring stations. Second, the arrival processes of returners at destination stations are affected by the state of other origin stations. Indeed, if a renter decides to abandon the system at Station A because it is empty, she will never be a returner at her presumed destination B. Moreover, in the long run, the service level experienced by users affects their demand in future periods since disappointed users are less likely to use the system again. In conclusion, we believe that future researchers should create detailed and empirically supported user behavior models. Such a model is unlikely to be efficiently computable but studying it by simulation may help to validate and fine tune the model presented here.

BSS operators generally sign service level agreements with the municipal authorities. In some of these agreements the service level is defined as the percentage of the time in which the station is allowed to be completely empty or completely full. For example, the operator may be committed to ensuring that none of the stations will be in such states more than 5% of the time each day. The parties agree upon a penalty for each breach of this commitment. We believe that this is an inaccurate measure for service level. The service level is affected not only by the length of the periods in which the station cannot provide the service but also and more importantly, by the timing of these periods. For example, an empty station next to a high school just before the beginning of the school day is desirable because such a station can accept the expected wave of returners. However, if the same station is empty in the afternoon, just before the end of the school day, it is likely to provide a poor service for the expected wave of renters. We therefore argue that the (estimated) number of unsatisfied renters and returners is a more appropriate service level measure and that it should be included in service level agreements between cities and BSS operators. Indeed, this measure cannot be calculated accurately but it can be easily estimated from the system log. However, if the operator is interested in minimizing the length of the period in which the station is empty or full, the definition of the UDF in (2) should be modified to

$$F(I_0) = \int_0^T \left(\pi_{I_0,0}(t) + \pi_{I_0,C}(t)\right) dt.$$

The adaptation of the approximation formula (3) is straightforward.

Finally, we would like to point out that the *sharing economy* is not limited to shared mobility systems. While the concept of renting equipment instead of buying it is not new, modern information technology makes it a very attractive alternative. For example, construction firms may prefer renting power tools and machinery at the time and location where they need them instead of owning and transporting them. Similarly, some individuals may prefer renting camping equipment in order to save the need to store it at their apartments, etc. Managing the supply chain of a multi-branch renting organization is a challenging task compared to a retail organization due to the need to cope with the dynamics of at least two incoming streams of goods: returned items and items acquired vertically or horizontally from other echelons. The literature on reverse logistics is extensive but primarily focused on systems with items that must be reprocessed in some way before being delivered again. However, in the rental business, returned items can be used again immediately after they are returned. Modeling such systems requires integrating inventory and queuing models. Therefore, we believe that the modeling concepts and solution methods presented in this paper may be relevant for many problems originating from the growing sector of renting/sharing organizations.

Acknowledgment: the authors wish to thank Mr. Shlomo Cohen and Mr. Danny Shpigel from FSM Ltd., Mr. Brodie Hylton and Mr. Danny Quarrel from Alta Bicycle Share Ltd. for providing data on the Capital Bikeshare and Tel-O-Fun systems; Professor Michal Tzur, Ms. Dana Pessach, and Ms. Chavatzelet Tryster, from Tel Aviv University, for helpful advice; to Mr. Avraham Edison for help in processing the demand data; and to two anonymous referees who contributed a lot in improving the presentation of this study.

References

Benchimol, M., Benchimol, M., Chappert, B., Taille, A.D.L., Laroche F., Meunier, F. and Robinet, L. (2010). Balancing the Stations of a Self Service "Bike Hire" System, to appear in *RAIRO Operations Research*.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia I., and Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey. *TOP* **15**, 1-31.

Capoor K. and Ambrosi, P. (2009). State and Trends of the Carbon Market 2009, The World Bank, Washington, DC.

Chemla, D., Meunier, F. and Wolfler Calvo, R. (2010), Bike hiring system: solving the rebalancing problem in the static case. Working paper.

Conolly B.W., P.R. Parthasarathyb, and N. Selvarajub, (2002), Double-ended queues with impatience, Computers and Operations Research, **29**, 2053–2072

Dekker, R. (2004). Reverse logistics: quantitative models for closed-loop supply chains. Berlin ; New York, Springer.

DeMaio P. (2009), Bike-sharing: History, Impacts, Models of Provision, and Future, *Journal of Public Transportation*, Vol. 12, No. 4, 2009.

Kashyap B.R.K., (1966), The double-ended queue with bulk service and limited waiting space, Operations Research, **14**, 822–834

Lin, J-R. and Yang, T-H. (2011), Strategic design of public bicycle sharing systems with service level constraints. Transportation Research Part E, 47, 284–294.

Mendoza G., M. Sedaghat, K.P., and Yoon (2009), Queueing models to balance systems with excess supply, International Business & Economics Research Journal, **8**, 91–103

MetroBike LLC (2011). observed in July 21, 2011, http://bike-sharing.blogspot.com/2010_12_01_archive.html

Nair R. and Miller-Hooks E. (2011), Fleet Management for Vehicle Sharing Operations, *Transportation Science*, **45**, 524-540

Raviv, T., Tzur, M., and Forma, I.,A., (2012), Static Repositioning in a Bike-Sharing System: Models and Solution Approaches, working paper

Shu J., M. Chou, Q. Liu, C-P. Teo, I-L. (2010), Bicycle-Sharing System: Deployment, Utilization and the Value of Re-distribution, working paper

Srivastava H.M., B.R.K. Kashyap, (1982), Special functions in queueing theory and related stochastic processes, Academic Press, New York

Ross, S. M. (2010). Introduction to probability models, Amsterdam ; Boston, Academic Press.

Vogel, P. and Mattfeld, D.C (2010), Modeling of Repositioning Activities in Bike-Sharing Systems, Proceeding of the 12th World Conference on Transport Research, July 11-15, 2010, Lisbon, Portugal.

Appendix A – Simulation based optimization

In this appendix, we describe the simulation-based optimization procedure used in the simulation study reported in Section 5. An internally documented Matlab implementation of this procedure is available upon request from the first author.

We used the demand data of the two months (39 working days) prior to the test period as input for the simulation of each station. We refer to this period as the *training period* and point out that it is the same data used to create the demand forecast used by the model. The dynamics of the inventory level in each station based on this demand were simulated assuming all possible initial inventory levels $0, \dots, C$ and the level that minimized the total number of shortages over the training period was selected for each station. This level was then used as the initial inventory level for the test period to estimate the expected number of shortages, assuming no dynamic repositioning, in the test period as reported in Section 5.

Alternatively, we could estimate the demand process based on the data of the training period and run the simulation with numerous replications. Clearly, as the number of replications grows, this procedure should converge to the solution obtained by our model. However, the usage of the actual demand data instead of estimating the demand process exhibits the attractive property of robustness to the assumption that the demand process is Poisson.

Appendix B – Extended model

Throughout this paper, we assumed that all the renters and returners abandon the station if they cannot obtain the service that they are seeking (i.e., an available bicycle or a vacant locker). This is a good approximation of the reality in well-operated bike-sharing systems since it is likely that the desired service can be obtained in a nearby station.

However, in this appendix we define a more general inventory model where renters that arrive at an empty warehouse can decide either to queue (or backlog) and to wait for an item or to abandon. Similarly, returners may decide to wait for a vacant space in the warehouse.

We note that while this generalization is unnecessary for bike-sharing systems, it may be useful for other closed loop inventory systems where the rental and return processes are intense enough to make waiting for an item a reasonable choice in some cases or in settings where backlogging is a viable alternative.

Extended Model

The state of the system I_t in the extended model is assumed to take any integer value $I_t \in \{-\infty, \dots, \infty\}$, where values $I_t \in \{0, \dots, C\}$ represent a warehouse with I_t items and no queues. A value $I_t < 0$ represents an empty warehouse and a queue of $-I_t$ represents would-be renters. A value $I_t > C$ represents a full warehouse with C items and a queue of $C - I_t$ would-be returners.

We model the decision whether to queue or not queue as a Bernoulli random variable where the probability of success is determined by the length of the queue and the time of day. Indeed, these two factors affect the distribution of the waiting time of each user since the queue of renters is served at a rate that is equal to the returners' rate and vice versa. We further assume that the decision to abandon is made immediately upon the arrival of the user at the warehouse and after witnessing the length of the queue. The likelihood of the renter (resp., returner) joining the queue at time t is denoted by the function $\beta_t(I_t)$ [resp., $\sigma_t(I_t)$]. Clearly, $\beta_t(I_t) = 1$ for all $I_t > 0$ and $\sigma_t(I_t) = 1$ for all $I_t < C$. The original model solved in Section 4 is clearly a special case of this model where $\beta_t(I_t) = 0$ for all $I_t \leq 0$ and $\sigma_t(I_t) = 0$ for all $I_t \geq C$.

We further assume that the arrival processes of potential renters and returners are Poisson processes with rates μ_t and λ_t. Accordingly, the actual demand rates faced by the station are $\mu_t\beta_t(I_t)$ and $\lambda_t\sigma_t(I_t)$. Consequently, the system can be represented as an unbounded non-time-homogeneous birth and death process as depicted in Figure 6.

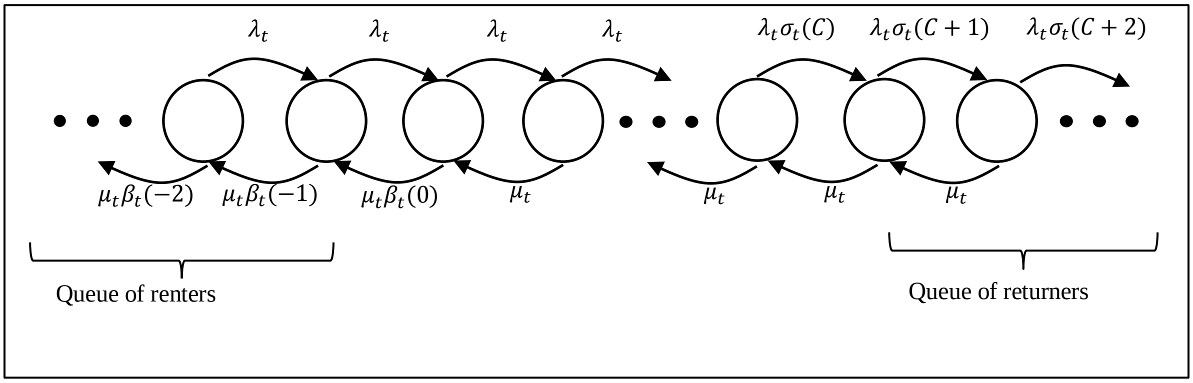

Figure 6: Continuous time Markov chain representing the dynamics of the items' inventory level

In the extended model the user dissatisfaction stems from two different factors: abandonment due to the inability of the system to provide the service within an acceptable time period and the waiting time at the station. Each of these phenomena can affect both renters and returners. The system is penalized for each of these event types as detailed below:

p^A Penalty charged for each renter who abandons due to shortage of items.

h^A Penalty charged for each returner who abandons due to shortage of storage space.

p^Q Penalty charged per time unit of a renter waiting for an available item.

h^Q Penalty charged per time unit of a returner waiting for a vacant storage space.

Note that, p^A and h^A are equivalent to the p and h parameters of the original model. Using the same notation as in section 4, we defined the extended UDF as follows:

$$\tilde{F}(I_0) = \int_0^T \left(\sum_{j=-\infty}^{0} \pi_{I_0,j}(t)\left[p^A \mu_t\big(1-\beta_t(j)\big) - jp^Q\right] + \sum_{j=C}^{\infty} \pi_{I_0,j}(t)\left[h^A \lambda_t\big(1-\sigma_t(j)\big) + (C-j)h^Q\right] \right) dt. \tag{6}$$

The first summation term in the integral represents the expected user dissatisfaction accumulated when the station is empty. During such a period, abandonments occur at a rate of $\mu_t\big(1-\beta_t(j)\big)$ and each bears a penalty of p^A. In addition, the total waiting time of renters is accumulated at a rate that is equal to the queue length $-j$. The second summation term represents the expected user dissatisfaction that accumulates when there are no vacant lockers at the station. During such a period of time, abandonments occur at a rate of $\lambda_t\big(1-\sigma_t(j)\big)$ and each bears a penalty of h^A. In addition, the total waiting time of returners is accumulated at a rate that is equal to the queue length, $C-j$.

A discretization method, similar to the one presented in Section 4, can be used to approximate (6). However, since the state space of the extended chain is infinite, one needs to truncate all the states that are extremely unlikely to occur within the planning horizon. Fortunately, it is reasonable to assume that the tendency of users to queue in the station drops rapidly with the length of the queue and therefore states with values that are much higher than the station capacity or much lower than zero can be safely truncated.

Approximation procedure

In order to carry out the discretization, we assume that the tendency to join the queue is constant for each discretized period. That is, for each short period θ these tendencies can be represented by a constant $\beta_\theta(I_t)$ and $\sigma_\theta(I_t)$. Based on the approximate value of $\pi(t)$, we can obtain a reliable approximation of the user dissatisfaction function using the following expression:

$$\tilde{F}(I_0) \approx \delta \sum_{\theta=0}^{\frac{T}{\delta}-1} \left(\sum_{j=-L}^{0} \pi_{I_0,j}(t+0.5\delta)\left[p^A \mu_\theta\left(1-\beta_\theta(j)\right) - jp^Q\right] \right.$$
$$\left. + \sum_{j=C}^{U} \pi_{I_0,j}(t+0.5\delta)\left[h^A \lambda_\theta\left(1-\sigma_\theta(j)\right) + (j-C)h^Q\right] \right). \tag{7}$$

where L (respectively, U) denotes an upper bound on the length of the queue of renters (respectively, returners). We suggest setting the value of L and U as follows

$$L = \max_{\theta \in \left\{1,\dots,\frac{T}{\delta}\right\}} \left\{ -\operatorname{argmax} \left\{ I_t \in \{-\infty, \dots, 0\} : \frac{\beta_\theta(I_t)\mu_\theta}{\lambda_\theta + \beta_\theta(I_t)\mu_\theta} \le \epsilon \right\} \right\}$$

$$U = \max_{\theta \in \left\{1,\dots,\frac{T}{\delta}\right\}} \left\{ \operatorname{argmin} \left\{ I_t \in \{C, \dots, \infty\} : \frac{\sigma_\theta(I_t)\lambda_\theta}{\mu_\theta + \sigma_\theta(I_t)\lambda_\theta} \le \epsilon \right\} \right\}$$

for some small $\epsilon > 0$. That is, the chain is truncated on arrival at the first state from which on, the probability of an increase in the queue length upon the next event is negligible. The approximation procedure is parameterized by ϵ and δ. Clearly, as $\delta, \epsilon \to 0$ the approximated value approaches $F(I_0)$ but the computational effort increases. Extensive numerical experiment, whose result can be on obtained from the first author upon request shows that this procedure yield results that agree nicely with simulation.

Appendix C – Detailed Results of Tel-O-Fun Case study

In this appendix, we present the detailed results of our numerical study. In Section 5, Table 1, we presented summary statistics of the accuracy test with various discretization levels. Here in Table 2, we present the results calculated for each of the 82 instances. The first column identifies the station (the true identity is concealed as per the request of the operator). The second column specifies the station capacity. The rest of the columns are divided into three groups, each correspond to one of the three tested discretization levels. For each level we present the computation time, *max error* and *average error*. For definition and discussion of these measures see Section 5.

Station ID	C	$\delta = 1$ min.			$\delta = 5$ min.			$\delta = 30$ min.		
		Time (sec.)	Max Error	Average Error	Time (sec.)	Max Error	Average Error	Time (sec.)	Max Error	Average Error
1	20	0.257	0.42%	0.32%	0.058	2.10%	1.59%	0.016	13.24%	9.50%
2	20	0.258	0.36%	0.29%	0.057	1.80%	1.45%	0.016	11.36%	8.72%

3	18	0.255	0.39%	0.31%	0.056	1.97%	1.55%	0.014	12.52%	9.33%
4	20	0.256	0.65%	0.53%	0.057	3.32%	2.67%	0.016	23.37%	16.05%
5	20	0.264	0.53%	0.35%	0.057	2.66%	1.73%	0.014	17.23%	10.41%
6	20	0.269	0.81%	0.58%	0.059	4.14%	2.88%	0.015	29.00%	17.43%
7	20	0.258	0.79%	0.54%	0.055	4.11%	2.68%	0.016	30.29%	16.08%
8	20	0.269	0.74%	0.53%	0.060	3.81%	2.67%	0.016	26.66%	16.10%
9	20	0.263	0.43%	0.33%	0.060	2.15%	1.67%	0.017	13.38%	10.04%
10	17	0.257	0.39%	0.30%	0.057	2.00%	1.50%	0.015	13.02%	9.03%
11	17	0.261	0.49%	0.38%	0.058	2.46%	1.88%	0.015	15.30%	11.24%
12	19	0.248	0.43%	0.35%	0.055	2.19%	1.76%	0.015	13.85%	10.55%
13	20	0.261	0.63%	0.46%	0.062	3.18%	2.30%	0.016	21.17%	13.87%
14	20	0.271	0.38%	0.31%	0.059	1.89%	1.54%	0.017	12.17%	9.25%
15	20	0.263	0.30%	0.26%	0.057	1.52%	1.32%	0.015	9.44%	7.93%
16	20	0.269	0.30%	0.25%	0.062	1.50%	1.26%	0.018	9.36%	7.58%
17	20	0.264	0.43%	0.33%	0.062	2.16%	1.65%	0.017	13.68%	9.90%
18	20	0.272	0.36%	0.30%	0.061	1.84%	1.50%	0.017	11.87%	8.95%
19	16	0.260	0.43%	0.40%	0.058	2.19%	1.99%	0.015	14.65%	11.84%
20	20	0.276	0.36%	0.31%	0.060	1.82%	1.53%	0.017	10.94%	9.11%
21	20	0.272	0.45%	0.41%	0.058	2.29%	2.06%	0.016	15.90%	12.31%
22	20	0.254	0.42%	0.35%	0.057	2.16%	1.75%	0.016	15.00%	10.39%
23	17	0.268	0.47%	0.37%	0.060	2.40%	1.85%	0.016	15.16%	11.10%
24	19	0.252	0.41%	0.34%	0.054	2.05%	1.68%	0.015	12.53%	10.04%
25	20	0.254	0.65%	0.45%	0.058	3.31%	2.26%	0.016	21.20%	13.61%
26	20	0.258	0.44%	0.36%	0.059	2.24%	1.82%	0.017	15.09%	10.92%
27	20	0.262	0.50%	0.42%	0.058	2.50%	2.09%	0.016	15.53%	12.34%
28	20	0.256	0.48%	0.42%	0.056	2.42%	2.10%	0.015	16.01%	12.53%
29	20	0.257	0.44%	0.37%	0.059	2.21%	1.87%	0.015	13.42%	11.20%
30	20	0.257	0.50%	0.43%	0.058	2.50%	2.14%	0.016	15.49%	12.83%
31	17	0.265	0.43%	0.39%	0.056	2.18%	1.96%	0.014	14.82%	11.62%
32	20	0.262	0.47%	0.36%	0.060	2.35%	1.78%	0.016	15.65%	10.69%
33	29	0.297	0.84%	0.71%	0.066	4.37%	3.53%	0.021	29.80%	21.29%
34	19	0.247	0.38%	0.29%	0.055	1.92%	1.47%	0.014	12.42%	8.85%
35	20	0.253	0.49%	0.46%	0.057	2.48%	2.30%	0.015	15.50%	13.83%
36	20	0.273	0.51%	0.44%	0.061	2.63%	2.18%	0.017	17.15%	13.02%
37	28	0.295	0.60%	0.44%	0.069	3.23%	2.19%	0.022	23.90%	12.95%
38	20	0.265	0.60%	0.53%	0.058	3.03%	2.67%	0.016	18.77%	16.01%
39	20	0.261	0.67%	0.55%	0.059	3.47%	2.74%	0.017	23.99%	16.44%
40	20	0.269	0.39%	0.34%	0.058	2.01%	1.68%	0.016	13.73%	10.06%
41	20	0.255	0.57%	0.40%	0.057	2.97%	1.98%	0.015	21.84%	11.79%
42	20	0.264	0.43%	0.39%	0.057	2.23%	1.96%	0.016	15.64%	11.64%
43	18	0.253	0.48%	0.44%	0.056	2.40%	2.19%	0.015	15.18%	13.07%
44	20	0.256	0.32%	0.28%	0.058	1.59%	1.40%	0.016	9.72%	8.38%
45	20	0.255	0.47%	0.42%	0.058	2.41%	2.09%	0.016	15.86%	12.52%
46	23	0.255	0.45%	0.38%	0.061	2.35%	1.90%	0.017	17.04%	11.21%
47	20	0.275	0.48%	0.42%	0.062	2.41%	2.12%	0.017	15.20%	12.67%
48	20	0.265	0.59%	0.50%	0.060	3.02%	2.50%	0.016	20.28%	14.96%
49	20	0.271	0.56%	0.44%	0.058	2.94%	2.20%	0.016	21.05%	13.11%
50	19	0.265	0.46%	0.38%	0.056	2.34%	1.90%	0.014	14.82%	11.34%
51	20	0.257	0.39%	0.31%	0.057	2.00%	1.57%	0.016	13.61%	9.22%
52	20	0.256	0.43%	0.35%	0.056	2.19%	1.74%	0.015	14.72%	10.32%
53	20	0.259	0.31%	0.27%	0.057	1.53%	1.35%	0.016	9.34%	8.08%
54	20	0.259	0.53%	0.43%	0.057	2.74%	2.16%	0.015	19.39%	12.83%
55	19	0.248	0.53%	0.48%	0.055	2.71%	2.42%	0.015	17.10%	14.40%
56	20	0.254	0.54%	0.46%	0.060	2.74%	2.32%	0.016	17.57%	13.70%
57	20	0.257	0.54%	0.48%	0.057	2.73%	2.38%	0.016	17.59%	14.22%
58	20	0.256	0.39%	0.30%	0.061	1.99%	1.48%	0.015	14.07%	8.86%
59	20	0.258	0.33%	0.30%	0.058	1.69%	1.50%	0.015	10.80%	8.93%
60	20	0.257	0.56%	0.46%	0.057	2.84%	2.32%	0.016	18.39%	13.86%
61	20	0.256	0.49%	0.38%	0.057	2.49%	1.92%	0.015	15.96%	11.50%
62	20	0.268	0.61%	0.49%	0.058	3.11%	2.43%	0.017	21.09%	14.61%
63	19	0.246	0.38%	0.35%	0.056	1.92%	1.74%	0.014	11.90%	10.44%
64	20	0.256	0.50%	0.36%	0.057	2.52%	1.80%	0.016	16.64%	10.85%
65	20	0.258	0.47%	0.41%	0.069	2.39%	2.07%	0.018	15.99%	12.34%

66	20	0.281	0.64%	0.59%	0.059	3.27%	2.96%	0.020	22.15%	17.77%
67	20	0.285	0.56%	0.42%	0.064	2.92%	2.10%	0.019	20.47%	12.50%
68	20	0.272	0.56%	0.48%	0.064	2.86%	2.39%	0.016	18.74%	14.36%
69	16	0.261	0.62%	0.47%	0.055	3.17%	2.35%	0.013	21.34%	14.18%
70	18	0.281	0.47%	0.43%	0.056	2.44%	2.14%	0.015	17.06%	12.73%
71	20	0.279	0.28%	0.25%	0.059	1.41%	1.26%	0.017	8.58%	7.58%
72	17	0.267	0.45%	0.42%	0.061	2.31%	2.11%	0.015	15.43%	12.68%
73	20	0.281	0.38%	0.33%	0.064	1.95%	1.64%	0.017	13.52%	9.78%
74	20	0.284	0.56%	0.47%	0.058	2.86%	2.35%	0.019	19.48%	14.13%
75	20	0.293	0.33%	0.29%	0.061	1.66%	1.47%	0.018	10.17%	8.75%
76	20	0.267	0.36%	0.31%	0.060	1.81%	1.55%	0.015	11.24%	9.28%
77	20	0.256	0.31%	0.28%	0.058	1.55%	1.39%	0.017	9.86%	8.33%
78	20	0.269	0.45%	0.38%	0.062	2.30%	1.91%	0.017	15.18%	11.38%
79	20	0.272	0.59%	0.46%	0.062	3.01%	2.28%	0.016	19.81%	13.71%
80	20	0.270	0.51%	0.38%	0.057	2.60%	1.92%	0.016	16.98%	11.49%
81	18	0.252	0.62%	0.49%	0.056	3.16%	2.47%	0.014	21.75%	14.82%
82	20	0.257	0.32%	0.29%	0.058	1.61%	1.46%	0.016	10.22%	8.77%

Table 2: Results the accuracy test

In our simulation study in Section 5, we compared the performance of the system under the initial inventory prescribed by our method with results obtained from simulation-based optimization (See appendix A) and the current practice. The detailed results for the five-day test period are presented in Table 3. The first column identifies the station. The capacity of the station is presented in the second column. The rest of the columns in the table are divided into three groups. The first refers to the expert solution, the second to the optimization method introduced in this study and the last to simulation based optimization. In each group, the first column reports the target initial inventory (X_0) suggested by the method. The second column reports the total number of bicycles that should be removed and/or added to the station during the five-day test period in order to avoid shortages. The third column reports the number of days (out of the five) in which at least one visit of the repositioning vehicle was required at the station.

Station	Capacity	Expert Solution			Our Model			Simulation Based		
		X_0	Work	Visits	X_0	Work	Visits	X_0	Work	Visits
1	20	8	2	1	11	0	0	10	0	0
2	20	5	6	2	11	0	0	9	0	0
3	18	4	9	4	9	0	0	8	0	0
4	20	8	0	0	3	2	1	2	3	1
5	20	7	0	0	9	0	0	6	0	0
6	20	2	19	4	2	19	4	1	15	4
7	20	14	34	4	18	28	4	19	28	4
8	20	5	8	2	4	6	1	3	5	1
9	20	4	12	3	7	11	2	7	11	2
10	17	5	7	3	8	5	1	8	5	1
11	17	10	26	5	10	26	5	9	25	5
12	19	6	1	1	8	0	0	7	0	0
13	20	5	0	0	8	0	0	8	0	0
14	20	5	1	1	9	0	0	8	0	0
15	20	13	5	2	11	1	1	12	3	2

16	20	5	9	2	9	11	1	9	11	1
17	20	10	0	0	10	0	0	10	0	0
18	20	10	1	1	11	2	1	11	2	1
19	16	14	20	4	12	21	5	12	21	5
20	20	5	0	0	5	0	0	5	0	0
21	20	14	18	4	16	15	2	15	16	3
22	20	10	62	5	15	41	4	14	43	4
23	17	5	25	4	12	20	1	11	19	1
24	19	4	2	2	9	0	0	9	0	0
25	20	5	38	5	10	17	4	8	23	5
26	20	12	11	3	12	11	3	11	10	3
27	20	12	48	5	15	33	5	14	38	5
28	20	17	28	4	15	25	5	14	24	5
29	20	8	16	4	13	8	3	13	8	3
30	20	14	3	1	12	1	1	10	0	0
31	17	15	16	4	13	12	4	12	11	4
32	20	15	8	2	15	8	2	15	8	2
33	29	2	20	5	2	20	5	2	20	5
34	19	4	12	3	9	0	0	9	0	0
35	20	4	2	1	7	5	2	6	3	2
36	20	15	29	5	14	30	5	14	30	5
37	28	20	63	4	23	57	4	22	59	4
38	20	8	3	2	8	3	2	8	3	2
39	20	2	15	4	2	15	4	2	15	4
40	20	14	12	2	13	12	2	12	12	2
41	20	16	17	3	18	22	5	18	22	5
42	20	16	36	5	15	37	5	14	38	4
43	18	16	50	5	13	41	5	13	41	5
44	20	7	5	1	9	8	2	9	8	2
45	20	8	56	5	3	31	5	2	28	5
46	23	16	21	4	16	21	4	15	22	5
47	20	14	33	4	14	33	4	13	32	5
48	20	2	51	5	2	51	5	2	51	5
49	20	16	38	5	17	33	5	18	30	5
50	19	12	21	3	14	15	3	12	21	3
51	20	2	107	5	9	86	5	7	90	5
52	20	2	33	5	9	6	3	8	9	3
53	20	5	6	3	10	2	1	9	1	1
54	20	20	87	5	18	81	5	18	81	5
55	19	6	59	5	6	59	5	6	59	5
56	20	2	89	5	2	89	5	2	89	5
57	20	6	50	5	3	41	5	4	44	5
58	20	18	33	5	11	18	2	10	18	2
59	20	20	161	5	18	153	5	18	153	5
60	20	4	14	4	4	14	4	3	19	5
61	20	8	3	1	10	1	1	9	2	1
62	20	2	17	2	2	17	2	2	17	2
63	19	5	6	1	8	9	3	8	9	3
64	20	3	1	1	5	0	0	4	0	0
65	20	8	7	2	13	3	1	14	4	1
66	20	2	29	4	7	16	4	6	16	4
67	20	18	41	4	17	39	4	18	41	4
68	20	2	113	5	2	113	5	2	113	5
69	16	2	8	4	4	2	1	4	2	1
70	18	6	11	3	6	11	3	6	11	3
71	20	2	5	3	8	0	0	8	0	0
72	17	2	2	2	2	2	2	3	2	1
73	20	13	20	3	13	20	3	13	20	3
74	20	10	9	3	14	0	0	13	1	1
75	20	3	14	4	9	1	1	7	3	1
76	20	5	4	2	8	0	0	6	2	2
77	20	8	0	0	13	0	0	12	0	0
78	20	17	64	5	13	48	5	13	48	5

79	20	2	5	2	4	3	2	3	4	3
80	20	5	9	3	5	9	3	4	8	2
81	18	4	9	2	11	4	2	12	7	3
82	20	6	0	0	9	0	0	9	0	0
Average		**8.37**	**23.60**	**3.06**	**9.80**	**19.56**	**2.55**	**9.29**	**19.96**	**2.63**
Total		**686**	**1935**	**251**	**804**	**1604**	**209**	**762**	**1637**	**216**

Table 3: Results of Tel-O-Fun simulation study