TEL AVIV UNIVERSITY

The Iby and Aladar Fleischman Faculty of Engineering

The Zandman-Slaner School of Graduate Studies

Management of Stock Levels in a Bike-Sharing System

A thesis submitted toward the degree of

Master of Science in Industrial Engineering

by

Ofer Kolka

August 2011

TEL AVIV UNIVERSITY

The Iby and Aladar Fleischman Faculty of Engineering

The Zandman-Slaner School of Graduate Studies

Management of Stock Levels in a Bike-Sharing System

A thesis submitted toward the degree of

Master of Science in Industrial Engineering

by

Ofer Kolka

This research was carried out in the Department of Industrial Engineering

Under the supervision of Dr. Tal Raviv

August 2011

Acknowledgments

First and foremost, I want to express my deepest gratitude to my supervisor, Dr. Tal Raviv, for his guidance, support, insights, enthusiasm and friendship that made this journey productive, as well as enjoyable. It was a privilege and pleasure to work with him over the past three years and learn on the research process and much more beyond that.

I want to thank my parents, Rachel and Arie for their endless support, for providing all the conditions to succeed in my studies and for initially encouraging me to take this long journey. I could not have done it without them.

Last but definitely not least, I want to thank my loving wife, Michal, who had the patience to bear me over this long period of time. Thank you for listening, encouraging and pushing me ahead at difficult times. This is yours as much as it is mine.

Abstract

Recently, cities around the globe introduced systems that allow people to rent a bicycle at one of many automatic rental stations scattered in the city, use them for a short travel and return them at any other station in the city. A typical Bike-Sharing system in a large city consists of hundreds of stations and thousands of bicycles.

One major issue, reported by operators of Bike-Sharing systems, is the availability of bicycles and lockers. The service level provided by the system may be adversely affected, either due to lack of bicycles in the origin stations, or due to lack of lockers in the destinations. Operators employ fleets of trucks to reposition bicycles among the stations in order to satisfy the fluctuating demand for bicycles and vacant lockers. Optimal operation of such a fleet poses an intricate stochastic inventory routing problem.

In this study we solve the inventory part of this problem. Namely, we develop a method to prescribe the number of bicycles to be added or removed from a station every time it is visited by a repositioning truck so as to minimize the shortage. Additionally, we study the strategic problem of determining the optimal stations capacity based on forecasted demand patterns. We calculate an efficiency frontier for the expected shortage (of bicycles or lockers) versus the station capacity.

Both the operational and strategic problems are complicated by the fact that the demand patterns are non homogenous, asymmetric and the frequency of events in each station is relatively low. Hence, the system cannot be assumed to reach steady state and thus an analysis of its transient behavior is required.

We model the system as a series of continuous time Markov chains, each representing a time interval during which the demand rates for bicycles and lockers is assumed to be constant. The probabilities of a station being empty or full at each time interval are calculated. Based on these probabilities, an expected total penalty for bicycles or lockers shortage is calculated.

The procedure is very fast and its output agrees well with the estimation obtained by a simulation model with numerous replications. We show that the optimal inventory level prescribed by our procedure is robust with respect to inaccuracies in the demand forecast.

We analyze the expected total penalty as a function of the initial inventory and prove its convexity. This constitutes the basis for convenient analyses in cases where

the events frequency is general and does not fit the Markovian assumptions of our model. Moreover, it allows optimization of the initial inventory level of a station in the system subject to various constraints.

Table of contents

List of Figures

List of Tables

List of Symbols and Abbreviations

Symbol / Abbreviation	Meaning	Explanation
Bicycle shortage penalty	Penalty which is charged for a *renter* abandonment	
Bicycle backlogging penalty	Penalty which is charged for waiting time of *renters*	
Bicycle surplus penalty	Penalty which is charged for a *returner* abandonment	
Locker backlogging penalty	Penalty which is charged for waiting time of *returners*	
p^A	*Bicycle shortage penalty*	
p^W	*Bicycle backlogging penalty*	
h^A	*Bicycle surplus penalty*	
h^W	*Locker backlogging penalty*	
T	Horizon time	The number of time periods until the next visit of the truck.
d	Discretization level	
$\mu(t)$	The expected renters demand rate at a station during period *t*	
$\lambda(t)$	The expected returners demand rate at a station during period *t*	
C	Station capacity	The number of stalls in a station.
IC(C)	Infrastructure cost as a function of C per locker per time unit	Cost of sidewalk space and equipment
$\beta(I_t)$	probability that a renter who arrives at the station at time *t* decides to join the renters queue	
$\sigma(I_t)$	probability that a returner who arrives at the station at time *t* decides to join the returners queue	
A	A stochastic arrival process of potential renters	
B	A stochastic arrival process of potential returners	
s	*scenario*	Any realization of processes *A* and *B* (jointly)
f(s)	Density function over all scenarios	
$\mathrm{I_t(s,X_O)}$	Inventory level	Number of bicycles available in a station given an initial inventory level X_O and scenario s at time t.
$\Theta_L(s,X_O)$	$\min\{t : I_t(s,X_O)=0\}$	
$\Theta_U(s,X_O)$	$\min\{t : I_t(s,X_O)=C\}$	
$G_-(X_O)$	Expected number of renters abandonment during the interval [0,T] as a function of X_O	
$g_-(s,X_O)$	Number of renters abandonment given scenario *s* and X_O	
$G_+(X_O)$	Expected number of returners abandonment during the interval [0,T] as a function of X_O	

$g_+(s,X_O)$	Number of returners abandonment given scenario *s* and X_O	
$H_-(X_O)$	Renters expected waiting time during the interval [0,T] as a function of X_O	A renter (that didn't abandon) waits at a station until he is first in line and a pair of bicycles arrives.
$h_-(s,X_O)$	Total waiting time of renters under scenario *s* and given X_O	
$H_+(X_O)$	Returners expected waiting time during the interval [0,T] as a function of X_O	A returner (that didn't abandon) waits at a station until he is first in line and a pair of bicycles arrives.
$h_+(s,X_O)$	Total waiting time of returners under scenario *s* and given X_O	
$\mathbb{I}_{\{Condition\}}$	Logic sign	$\mathbb{I}$ equals 1 as long as the condition holds and 0 else.
$F(X_O)$	Expected total penalty for the '*Replenishment problem'*	The expected weighted sum of shortage and surplus costs during the interval [0,T] as a function of X_O
$\pi(t)$	Transition probability matrix	
$P(t)$	Single period transition probability matrix for one time period	
M	Limit for the approximation of the 'transition probability matrix'	
R	Matrix from the approximation formula	
$\tilde{P}_n(t)$	The steady state probability of time interval *t*	
$\tilde{F}(X_O)$	Expected total cost for the '*Capacity problem'*	
UP	User patience	A renter or returner patience (in minutes) to join the queue in case the station is empty (resp., full)

1. Introduction and literature review

A Bike-Sharing system is a municipal system that users can access 24 hours a day. One can gain access to a bicycle either by inserting a credit card or by paying a periodical fee for a membership card. The bicycles can then be returned at any of the stations in the city (Posner 2008, TMD Encyclopedia 2008). A typical Bike-Sharing System consists of a fleet of bicycles, a network of automated stations where bicycles are stored, and bicycle redistribution and maintenance programs (TMD Encyclopedia 2008). The Systems are particularly appropriate in large cities where the demand for short trips is high and it is possible to have a dense network of stations, but may also be feasible in suburban areas and campuses (TMD Encyclopedia 2008).

A Bike-Sharing system has many advantages over other modes of public transportation for short-distance urban trips (Demaio 2004, Demaio 2008, TMD Encyclopedia 2008):

- Bicycles are relatively inexpensive to purchase and maintain.
- Bicycles generally do not add to vehicular congestion.
- Bicycles do not create noise and pollution in their operation.
- Bicycles provide the user with the added benefit of exercise.

However, in comparison to other modes of transportation, bicycles have their drawbacks (Demaio 2004):

- They can be uncomfortable in harsh weather.
- They can be used in ways unsafe to riders and pedestrians.
- They may be inaccessible to people with certain disabilities.
- They may be difficult to use in some topography.
- They are more appropriate for shorter distances.
- They can be subject to vandalism and theft.

For a bicycle to be a significant component of the urban transportation system, the city must provide a proper infrastructure of bicycle paths. Data collected by the city of Portland, Oregon, demonstrates a strong correlation between a connected bikeway system constructed to the highest standards, and increases in bicycle use. The count data shows an enormous increase over time in bicycle use in the city parts with the improved facilities (Birk and Geller 2005). The quantity and quality of the facilities were recognized as the key factors. In the 1995 Harris Poll survey, 20% of Americans said they would commute by bicycle or on foot more regularly if more improved facilities were provided (Oregon Department of Transportation, 1995). Various modal plans have been established in many cities around the world for providing guidelines to cities and counties for developing local bicycle and pedestrian infrastructure. The plans establish policies and implementation strategies, design methods, maintenance and safety information. Finally, targets are defined and proper guidance is given for achieving them (see Oregon Department of Transportation, 1995, Minnesota Transportation Department 2005).

There have been three generations of Bike-Sharing systems over the past 40 years. The first attempt to implement a Bike-Sharing program dates back to 1968 with the first generation Bike-Sharing system in Amsterdam, The Netherlands. Bicycles painted in white were scattered throughout the city. Individuals were to find a bicycle, ride it to their destination, and leave it for the next user. However, the bicycles were stolen and the system collapsed within days. In 1995 in Copenhagen, Denmark, a second generation of Bike-Sharing systems was launched with improvements. These bicycles were specially manufactured for intense utilitarian use and could be picked up and returned at specific locations throughout the central city with a coin deposit. However, theft of bicycles in these second generation systems continued to be a problem due to the anonymity of the customers (Demaio 2004, Demaio 2008). This gave a rise to a third generation of Bike-Sharing systems – the smart bike. Smartening earlier Bike-Sharing systems with electronic lockers or bicycle locks, telecommunication systems, and smartcards or magnetic stripe cards, has allowed better tracking because the customer’s identity is known. Customers not returning a bicycle within the allotted time for its use are charged for the replacement cost of the bicycle. These technological features offer great improvements over earlier systems, which had no high-tech features for checkout or return, and relied solely on customer honesty (Demaio 2004). In addition, such systems gather demand data online. Analyses of this data may be used for future design decisions and for ongoing operational ones.

Froehlich et al. (2009) use the data collected from third generation Bike-Sharing systems to explore patterns of user behavior. They provide a spatiotemporal analysis of 13 weeks of bicycle station usage from the Bicing system in Barcelona, apply clustering techniques to identify shared behaviors across stations and show how these behaviors relate to location, neighborhood, and time of day. Their models are able to predict station usage with an average error of only two bicycles and can classify station state (full, empty, or in-between) with 80% accuracy up to two hours into the future. Their experiments indicate that 10 to 15 weekdays of historic data are enough to build station models.

Two models of Bike-Sharing exist - one designed for community use and the other for residential use. In the community Bike-Sharing model, an individual checks out a bicycle from one of many locations and returns it to another location. The residential Bike-Sharing model requires bicycles to be returned at the same location from where they were checked out. The residential model, which is used in Japan, is designed for denser cities where living and bicycle parking spaces are at a premium (Demaio 2004). The focus of this study is on community Bike-Sharing.

Bike-Sharing programs across Europe are used by both tourists and residents. The systems, in general, are quite successful (Becker 2008). The largest system is Vélib launched in July 2007 in Paris. As of July 2010 there are about 1800 renting stations, 25,000 bicycles and the average usage increased up to about 166,500 rentals a day.

Beyond Europe, the interest in the concept of public bicycles is also rising, e.g. in the US (Richard 2008), in Canada (http://www.bixisysteme.com/accueil), Australia (Gardiner 2008), Argentina (Diaz 2008), China (Woodland 2008) or Israel (Bar-Eli 2009). During the last years the number of cities that already implemented Bike-Sharing systems or plans to do so, increased. The success of the concept was proven in Lyon, Paris, Munich and Barcelona, where large scale and automated bicycle rental services have been implemented and offer thousands of public bicycles to the citizens. As of December 2010, there are about 238 Bike-Sharing services around the world. In 2009 the number was about 160, which shows a 49% increase. Additionally, there are another 53 services that are in planning stages and may soon be operational (see Figure 1 for the increase in the number Bike-Sharing systems since 2004) suggesting that nearly every large city wants to offer such a service. The successes of the new systems can be attributed to their dense coverage of the cities and to the information technology that allow coping with theft and improving their operations. Environmental issues awareness is also a factor in the success of Bike-Sharing systems. Cities around the world have begun to embrace Bike-Sharing as a way to improve quality of life and meet greenhouse gas reduction targets (Posner 2008, Demaio 2010).

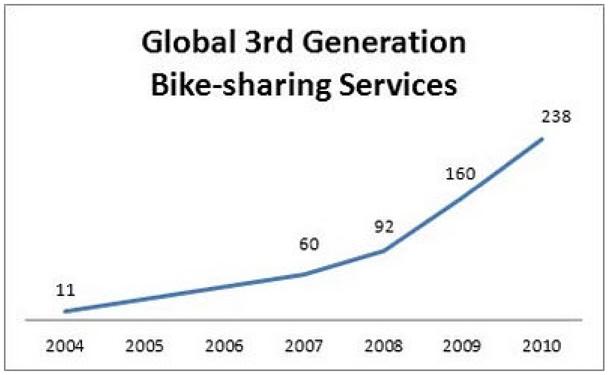

Fig 1. Number of Bike-Sharing systems across the world since 2004

As the interest in Bike-Sharing systems increased, more companies became involved in the industry (Demaio 2008). Companies such as ASK (http://www.ask-rfid.com) provide smart cards with an embedded microchip that allows users to rent bicycles in a third generation Bike-Sharing system (Collins 2005). Nonprofit organizations as BIKES BELONG or HUMANA whose goal is to maximize bicycle funding for healthier communities, cleaner air, less traffic and thriving bicycle businesses are trying to increase awareness for the advantages of using bicycles through Bike-Sharing (Tucker 2008). Another example for the arising industry is consulting groups such as Alta Bicycle Share (http://www.altabicycleshare.com) that designs, deploys, and manages bicycle share systems.

Shortage of bicycles and vacant lockers are the main complaint voiced by users of Bike-Sharing systems. In Brussels, for example, a voluntary group of users created a web service that pulls inventory data from the city's Bike-Sharing (Villo) website in order to monitor the shortage and create a public pressure on the operator to improve it. According to the group's web site (http://www.wheresmyvillo.be), their main cause is to make "JCDecaux (the operator) drastically improve the availability of bicycles and parking spaces, through better reallocation of bicycles". Their website displays statistics about the proportion of time in which at least one bicycle (resp., locker) was available in each station during the last seven days. As of October 14, 2010, the ten worst stations, out of the system's 180 stations, could not provide a single bicycle more than 33% of the time (resp., provide a single locker more than 23% of time).

The riders complain that the system's lockers can run out toward the end of the morning rush hour, leaving customers temporarily stranded. Likewise, the lockers are sometimes full, so riders have to search for parking (Rosental 2008, Price 2008). The operator of Vélib in Paris estimates that 93 percent of trips end satisfactorily.

Vélib uses 23 light flatbed trucks with the capacity of carrying 20 bicycles, and 2 heavy trucks with the capacity of carrying 62 bicycles to shift bicycles among the stations, and to move bicycles back upstream after and during rush hour (Price 2008). In hilly cities, many riders prefer to ride downhill but to use other modes of transportation uphill. Thus, the stations that are situated at high locations become empty quickly, while lower stations become full. In such situations, the operator must regularly transfer bicycles uphill.

Another possible method to balance the Bike-Sharing system is by using economic incentives. For example, the Vélib operators in Paris give extra riding time for the riders who drop the bicycles at stations uphill. Vélib operators are also considering to add lockers at popular stations and to increase redistribution at night (Price 2008, "Epic Bike-Sharing Post" 2008).

Among other reasons, the distribution problem is an important factor for which the location of rental stations should be well planned according to the expected demand. A methodical redistribution of bicycles is needed to guarantee the availability of bicycles and avoid frustration for users (Niches 2008).

To the best of our knowledge, the distribution method in the operating Bike-Sharing systems is ad-hoc and heuristic. That is, the current operation of these systems is probably non-optimal. In general, the research existing on Bike-Sharing is not extended. Moreover, there are only a few papers that deal with management of the inventory levels in Bike-Sharing systems (Demaio 2004).

A more general problem relative to the distribution of bicycles at each station is the decision regarding the frequencies and the routes that the trucks should follow. Several papers considered this routing problem with static repositioning, i.e. it is assumed that bicycles are moved during slack hours when the system is nearly inactive.

Chemla et al. (2010) consider a static routing problem that is a variation of the Many to Many Pickup and Delivery Problem. They consider a single truck and assume that each station can be visited several times and no time limit is imposed on the truck to satisfy the whole demand. An initial and desired inventory of bicycles at each station is given. The goal is to find the minimal cost route that can bring the system to a desirable state. They describe a branch-and-cut algorithm for solving a relaxation of the problem and show how to build a feasible solution based on the one obtained by this relaxation. In addition, the same algorithm is used to prove that the solutions obtained with the relaxations are often the optimal ones. An upper bound of the optimal solution of the problem is obtained by a tabu search.

Raviv et al. (2010) consider an inventory-routing problem of determining the routes that repositioning trucks should follow, and the number of bicycles that should be removed or placed in each station at each visit of the vehicles. As opposed to pickup and delivery problems such as the problem suggested by Chemla et al, the quantities picked-up or delivered to a station are not given. The quantities Raviv et al use are derived from the model formulated in this paper. They use an objective function that considers the users satisfaction and present several approaches for modeling the problem as a Mixed Integer Linear programming based on different assumptions. Based on the solution of a variety of instances, one approach is found to be very effective in solving problems with up to 60 nodes and two vehicles.

Nair and Miller-Hooks (2010) formulate the Vehicle Sharing Programs (VSP) fleet management problem as a stochastic MIP. They suggest static redistribution at the beginning of the planning horizon. The goal is to minimize the redistribution operation cost so that a required service level (a proportion of all short term demand scenarios) is provided, given the initial inventory levels in each station. The service level offered is quantified using a framework that they develop. The stochastic MIP has a non convex feasible region. Two solution techniques are presented: One technique is based on enumeration. The main idea is to transform the non convex feasible space to a disjunctive set of convex spaces. This transformation leads to a family of MIPs, one for each convex set. The second technique achieves quicker performance using a cone generation method. This technique is constrained to the assumption that the random demand vector is uncorrelated. In an application of the proposed framework to a car sharing system in Singapore, the operational strategies were found to be robust in simulation studies.

Literature on the similar shared mobility systems of motor vehicles, called Car-Sharing, is more extensive, (see for example Kek et al 2006, Barth et al. 2004, Mukai and Watanabe 2005, Uesugi et al. 2007). However, these studies offer little guidance for Bike-Sharing, as the technologies and issues are quite different (Demaio 2004).

Over the last decade, car sharing has emerged as an alternative to owning a vehicle. Most of this form of transportation has been taking place in Europe, North

America, Japan and Singapore. Conventional car sharing systems usually requires users to pickup and return vehicles at the same stations. Stiff competitions from public transportation systems and competing car sharing companies have prompted some operators to provide users with the flexibility in return stations (*one-way car sharing systems*) and some car sharing systems provides users with flexibility in return time (Kek et al 2006). A key issue that arises from such systems is similar to the Bike-Sharing systems problem - the dynamically disproportionate distribution of vehicles across stations, with no prior knowledge. As a result, periodic relocation becomes necessary to ensure an even distribution of vehicles to serve customer demands. These systems differ from Bike-Sharing systems in two key aspects (Kek et al 2006, Barth et al. 2004, Mukai and Watanabe 2005):

- Due to the size of a car, it cannot be relocated by a truck. The cars are being distributed one by one.
- The cars are usually reserved in advanced, thus it is usually known when a customer is coming, where the pickup location is and where the drop-off location is.

Kek et al. (2006) suggested a novel three-phase Optimization-Trend Simulation (OTS) decision support tool for one-way car sharing systems. The tool assists the operators to find a set of near-optimal manpower and relocation parameters for their vehicle relocation systems. Two Level of Service indicators are used- Full-Port Time (FPT) for time periods that a station is completely full, and Zero-Vehicle Time (ZVT) for the time periods that a station is completely empty. Simulation tests, based on a set of real operational data, have produced statistics that indicate a better system performance than the existing system. The three-phase OTS tool recommends a set of parameters for vehicle relocation operations, enabling a reduction in staff cost, an improvement of ZVT and a maintenance of the low FPT level.

Mukai and Watanabe (2005) focused on location balances of cars in a one-way car sharing system. They proposed a relocation algorithm for waiting cars based on virtual spring forces. The algorithm divides a service area equally for keeping homogeneity of location balances among waiting cars, dynamically. Results of a simulation experiment show that the algorithm is effective for these kinds of systems.

Uesugi et al. (2007) present a one-way car sharing system method for optimizing car assignment according to distribution balance of parked cars. The optimization is done by assignment of the optimal number of cars to group of users with similar ride-on and drop-off stations. It is assumed that users make reservations (including ride-on and drop-off stations), i.e. the demand for cars and for parking spaces is deterministic. The decision of the number of assigned cars is based on a square residual error sum for the number of parked cars and the optimum number of parked cars. Thus, groups of users select the number of using vehicles which minimizes a residual sum of squares between a ride-on station and a drop-off station. Future work of this paper must also consider a proper incentives system, so that users will behave according to the proposed model.

Barth et al. (2004) introduce two user-based relocation mechanisms called trip joining or trip splitting in a one-way car sharing system. When the system realizes that it is becoming imbalanced, it urges users to take separate vehicles when more vehicles are needed at the destination station (trip splitting). Conversely, if two users are at the origin station at the same time traveling to the same destination, the system can urge them to rideshare (trip joining). This concept has been implemented both on a real world university campus shared vehicle system and in a high-fidelity computer simulation model. The model results show that there can be as much as a 42% reduction in the number of relocations using these techniques.

Inventory models studied in the Reverse Logistics literature have some relevance for inventory management of a Bike-Sharing system. Reverse Logistics focus on two alternatives for fulfilling the demand - order the required raw materials externally or overhaul old products returned by users. Typically, the producer has little control on the return flow in terms of quantity, quality and timing (Fleischmann et al. 1997). In Reverse logistics inventory models, the total demand and return events during the planning horizon is of interest. This is due to the minor proportion of old products returned by users relative to the overall products sold and due to the availability of external products for purchasing during the planning horizon. However, considerations of inventory management of a Bike-Sharing system are different in the sense that the order of events (rents and returns) is crucial. This is because the flow of products (bicycles and lockers) is stochastic (in opposed to the classical reverse logistic literature where ordering is possible), and thus cannot be predicted or controlled.

Mahadevan et al. (2003) studied a single remanufacturing facility that receives a stream of returned products according to a Poisson process. The assumption is that demand is uncertain and also follows a Poisson process. The decision problems for the remanufacturing facility are when to release returned products to the remanufacturing line and how many new products to manufacture. They employed a ‘‘push’’ policy that combines these two decisions. Modeling the system using simulation, the authors observed the quasi-convexity of the objective function in the decision variable, and found some unusual behavior, such as costs decreasing when lead times increase. Thus, they developed several heuristics based on traditional inventory models. The two first approaches rely on an approximation of the manufacturing and remanufacturing sources by a single aggregate channel. The third approach explicitly considers the impact of both channels separately. They also investigated the performance of the system as a function of return rates, backorder costs and manufacturing and remanufacturing lead times, and developed approximate lower and upper bounds on the optimal solution.

Fleischmann et al. (2002) present an inventory model comprising Poisson demand and returns. Purchase orders arrive after a fixed lead-time, and any un-served demand is backlogged. In addition, there are returns of items into the inventory according to a Poisson process independent of demand. For this model, the authors have shown optimality of an (s,Q) policy for ordering new items and have pointed out how to determine optimal values of the control parameters.

Fleischmann and Kuik (2003) considered a single inventory point facing independent stochastic demand and item returns. Using general results on Markov decision processes, they showed average cost optimality of an '(s; S) order policy' in this model. The key result concerns a transformation of the model into an equivalent traditional '(s; S) model' without return flows, using a decomposition of the inventory position.

We conclude that while shortage of bicycle and vacant lockers is a major concern for user and operators of Bike-Sharing systems, only few studies considered the optimal management of stock levels in a these systems.

In this study we introduce an inventory model for a Bike-Sharing rental station, assuming stochastic and non-stationary check-out and return processes. We introduce a method to analyze this model, discover some of its important structural properties and derive some managerial insights. We focus on optimal decisions in two levels:

1. Operational Level – this is the *'Replenishment Problem'*, which decides upon the optimal initial inventory level at the station to be replenished at each truck visit,

so that the inconvenience of the users (i.e, abandonment or waiting time caused due to shortage of bicycles or lockers) will be minimized. The decision is based on forecasted information of the expected demand and the time of the next visit of the distribution truck in the station. The goal is to devise a method to calculate this optimal initial inventory level and to devise and evaluate fast approximation methods to carry out these calculations.

2. Strategic Level– this is the '*Capacity Problem'*, which decides upon the capacity of the stations where the tradeoff is infrastructure cost (sidewalk space and equipment) vs. minimum users discomfort for a given capacity.

2. <u>Problem definition, assumptions and research goal</u>

2.1 Environment characteristic

We consider the following basic inventory management problem: the inventory level of a station is reviewed periodically (say every night), during the period between the reviews there are rental events (i.e., arrival of users at a station who wish to rent a bicycle) and return events (i.e., arrival of users who return a bicycle). Both arrival processes are general stochastic and non-homogenous over time. When a user wishing to rent a bicycle arrives at the station and no bicycles are available, he may either abandon or wait until one becomes available. Similarly, if a user wishes to return a bicycle arrives at a station with no vacant locker, he may either wait or abandon and go to another station. Clearly, both types of shortage events are undesirable. A penalty cost is associated with user abandonments and with user waiting time. These penalties are special cases of shortage and backlogging costs.

In this study we are interested in analyzing the expected cost due to such shortage events as a function of the initial inventory set by the truck at the review time. This model is different from the classical models of the reverse logistics literature in the sense that the order of events (rents and returns) is crucial. This calls for an analysis method that is typical for queuing systems. However, the fact that the state of the system is externally controlled (by replenishing the station) requires transient rather than steady state analysis, as discussed in §3.1.

The goal of the operator in a Bike-Sharing system is to balance each station so it will be able to meet the fluctuating demand for bicycles, but also to provide enough vacant lockers to allow the riders to return their bicycles. The number of lockers in each station is tightly constrained due to area shortage (this area is typically allocated at the expense of sidewalk space or parking spots).

Our model assumes that the station is served by a repositioning truck at fixed intervals, say every night. The truck can either add some bicycles to the station or remove some.

As mentioned earlier, operating a Bike-Sharing system requires taking decisions on the frequencies and the routes that the trucks should follow. The problem studied here is a sub-problem of this inventory routing problem studied by Raviv et al (2010). Indeed, their objective function is constructed based on preliminary results of this study.

2.2 Objective function

The objective is to decide upon the optimal initial inventory level at the station to be replenished at each truck visit. A solution is measured by the expected total penalty between two consecutive visits of the truck. This total penalty consists of four penalties:

1. *Bicycle shortage penalty* which is charged for each potential customer (referred to as a *renter*) that arrives at the empty station and decides to abandon.
2. *Bicycle backlogging penalty* which is charged for waiting time of renters who decided to wait for a bicycle.
3. *Bicycle surplus penalty* that is charged for each user who tries to return a bicycle (referred to as *returner*) at a fully occupied station and decides to abandon and return the bicycle at other station
4. *Locker backlogging penalty* which is charged for waiting time of returner who decides to wait for a bicycle.

2.3 Problem formulation

In this section we introduce our notation and formalize the description presented in previous sections.

Decision variable

X_O	The initial inventory level which is set by the truck.

Note that for the problem of determining the optimal capacity of a station discussed in the sequel, the station's capacity is also a decision variable in some of our models below.

State Variables

I_t	System's state, i.e. the inventory level (number of bicycles available) in the station at time *t*. If $0 \leq I_t \leq C$ then I_t is the number of bicycles in the system. If $I_t \leq 0$ then the station is empty and there are $-I_t$ users

waiting in the queue for a bicycle. If $I_t \geq C$ then the station is full and there are $I_t - C$ users waiting in the queue for a locker. The state increases by one whenever a returner arrives and decreases by one whenever a renter arrives. At all other times I_t is constant.

Parameters

p^A	*Bicycle shortage penalty*
p^W	*Bicycle backlogging penalty* (per time unit)
h^A	*Bicycle surplus penalty*
h^W	*Locker backlogging penalty* (per time unit)
$\mu(t)$	The expected bicycle demand rate at the station at time *t*
$\lambda(t)$	The expected bicycle return rate at the station at time *t*
C	The station capacity, i.e. the number of lockers in a station
T	Time horizon, i.e. the time until the next visit of the truck.
IC(C)	Infrastructure cost (for the planning horizon) as a function of the station's capacity. We expect this function to be non-decreasing in reality, due to extra cost that is charged for more lockers. This function may be a step function with a fixed price for different ranges of the capacity.
$\beta(I_t)$	Probability that a renter arriving at the station at time *t* decides to join the queue at the station and to wait for service. 1-$\beta(I_t)$ is the probability that the renter abandons. $\beta(I_t)$ is assumed to be an increasing function of the current system's state and $\beta(I_t)=1$ for $I_t > 0$. This reflects the fact that the tendency of a renter to join the queue is determined by his expected waiting time, which is affected by the particular time of his arrival and the current length of the queue. If a renter arrives at the station when bicycles are available he will definitely join the (empty) queue.
$\sigma(I_t)$	Probability that a returner arriving at the station at time *t* decides to join the queue at the station and to wait for service. 1-$\sigma(I_t)$ is the probability that the returner abandons. $\sigma(I_t)$ is assumed to be a

decreasing function of the current system state and $\sigma(I_t)=1$ for $I_t < C$. If a returner arrives at the station when lockers are available he will definitely join the (empty) queue.

Next we introduce two abstract mathematical models that are aimed to minimize the total cost incurred at a station.

1. The Replenishment problem:

$$\underset{X_O}{\text{Min}}\ \ \mathrm{F}\left(p^A, h^A, p^W, \mathrm{h}^W, \mu(t), \lambda(t), \mathrm{C}, \mathrm{T}, X_O\right) \qquad (1)$$

where $\mathrm{F}\left(p^A, h^A, p^W, \mathrm{h}^W, \mu(t), \lambda(t), \mathrm{C}, \mathrm{T}, X_O\right)$ is the total expected cost given the above parameters and the initial inventory X_0. This model supports the operational challenge faced by the operators of the repositioning fleet. That is to decide what should be the inventory level set by the truck when visiting a station of capacity C with forecasted demand processes $\mu(t)$ and $\lambda(t)$, assuming the next visit at the station is expected after T units of time and that the capacity constraint of the trucks is not binding.

2. Capacity problem objective function:

$$\underset{X_O, C}{\text{Min}}\ \ \tilde{\mathrm{F}}\left(p^A, h^A, p^W, \mathrm{h}^W, \mu(t), \lambda(t), \mathrm{C}, \mathrm{T}, X_O, \mathrm{IC}\right) \qquad (2)$$

This model supports a medium term design problem of deciding the capacity of a station, assuming that the station will be reviewed periodically (say every night) and that the typical demand patterns during each cycle are given by demand processes $\mu(t)$ and $\lambda(t)$. For this model the infrastructure cost for the planning horizon, IC, must be known.

Note that these models are based on a single station analysis and independently in other close by stations. That is, the interaction within a network of stations is not modeled in this paper.

3. The Replenishment problem

In this section the Replenishment problem objective function is analyzed, whereas the Capacity problem objective function is considered in Chapter 4.

3.1 Steady state considerations

The majority of research in the queuing theory field focuses on steady-state analysis. The usefulness of this kind of analysis stems from the fact that many systems approach their asymptotic behavior quickly (Halfin and Whitt 1981, Whitt 1984).

Many service systems reveal significant time variation in the arrival rates. However, queuing models with non stationary arrival processes are difficult to analyze. A common engineering practice is to use the stationary Erlang loss model with a constant arrival rate obtained as an average over an appropriate time interval during which the system is most heavily loaded, e.g., a busy hour. With this peak hour approach, the assumed arrival rate in the model is usually greater than or equal to the real arrival rate the majority of the time, so that the computation results tends to be conservative (Davis et al. 1996). A more detailed way to analyze a queuing system is to divide the arrival process to subintervals, where each is characterized by approximately constant arrival rates (Massey and Whitt 1996). It is then assumed that the system approaches steady state at the beginning of each subinterval. Consequently, relevant measurement can be calculated.

In the Replenishment problem, assuming Poisson stochastic processes of renters and returners with demand rates of $\mu(t)$ and $\lambda(t)$ respectively, the problem can be presented as a $M/M/1/\infty$ birth and death process, as shown in Figure 2.

The birth rate is $\lambda(t)$ for all states in which vacant lockers are available in the stations (hence $\sigma(I_t)=1$) and $\lambda(t)\sigma(I_t)$ for $I_t \geq C$. Similarly, the death rate is $\mu(t)\beta(I_t)$ for $I_t(X_O) \leq 0$ and $\mu(t)$ otherwise.

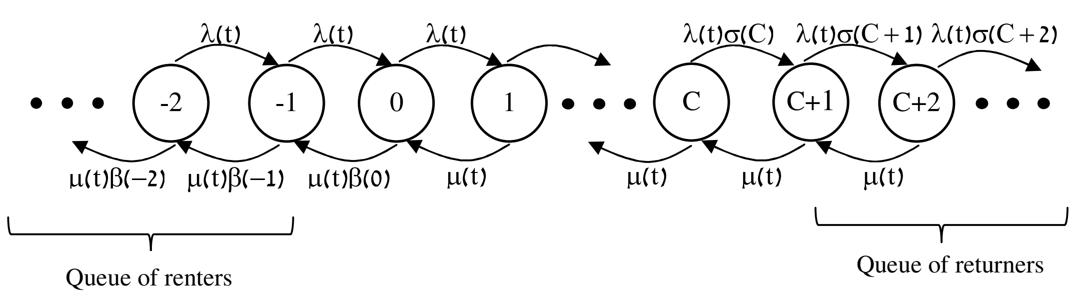

Fig 2. The Replenishment problem as death and birth process

However, for a Bike-Sharing environment, steady state analysis is unsuitable for a number of reasons. First, the demand for bicycles and lockers is asymmetric, i.e. if the process would stabilize there is a high probability that the station's state would be either empty or full. The system's operators want to avoid such situations, and thus replenish the bicycles in the station. Namely, the replenishment action prevents the system from approaching steady state. Second, steady state analysis is reasonable when: 1) the time line can be divided into intervals during which the arrival processes are approximately homogenous; 2) the rate of events is high enough so that the system approaches its steady state very quickly, with respect to the current rates. However, if the behavior of the system is very different than steady state behavior during a large share of each interval, this kind of analysis is useless. This is the case for a Bike-Sharing system environment since the variation in arrival rate is very different for each hour of the day, while the expected number of renters and returners arrivals per hour is typically small.

As a result, a transient analysis of the process must be considered. This analysis is hard to perform analytically, thus an efficient numerical approximation method is needed.

In §3.3 our approximation method is presented. In §3.3.2.8 a comparison between steady state and transient state analysis is made on a number of test problems to support the claim that the former is inappropriate for this setting.

3.2 Objective function convexity

In this section we consider three different cases of the Replenishment problem and show that in some interesting settings the expected total penalty function is convex with respect to the initial inventory, X_O. In §3.4 we discuss some important merits of this observation for Bike-Sharing system's operators.

3.2.1 Users with no patience

We first consider the special case of the Replenishment Problem where $\beta(I_t)=0,\ \forall I_t \le 0$ and $\sigma(I_t)=0,\ \forall I_t \ge C$. That is, all renters abandon an empty station, and all returners abandon a full station. No user waits at the station for an available bicycle or a vacant locker.

Note that if the rental station network is dense, this case is likely to be a good approximation of the reality since users may abandon the station and seek service in neighboring stations.

Assume a stochastic arrival process *A* of potential renters and arrival process *B* of returners. Any realization of these two processes (jointly) is called a *scenario.*

Let us denote the inventory level at time *t*, given initial inventory level X_O and scenario *s* by $I_t(s, X_O)$.

We denote the expected number of shortage events (shortage in bicycles) during the interval $[0, T]$ as a function of the initial inventory at time 0 by $G_-(X_O)$ and the expected number of surplus events (shortage in lockers) during this interval by $G_+(X_O)$. Clearly, $G_-(X_O)$ is a non-increasing function of X_O while $G_+(X_O)$ is a non-decreasing one.

The expected total penalty in a station during the interval $[0, T]$ is then given by:

$$F(X_O) = p^A \cdot G_-(X_O) + h^A \cdot G_+(X_O) \qquad (3)$$

To prove the convexity of $F(X_O)$ we will first prove some properties of $G_-(X_O)$ and $G_+(X_O)$.

Let us define $\min\{t : I_t(s, X_O) = 0\}$ as $\Theta_L(s, X_O)$ and $\min\{t : I_t(s, X_O) = C\}$ as $\Theta_U(s, X_O)$. Note that $\Theta_L(s, X_O)$ (resp., $\Theta_U(s, X_O)$) is the first time where the station is empty (resp., full) under arrival scenario *s* assuming initial inventory level of X_O. Let us use the convention that if no such event occurs, $\Theta_L(s, X_O) = T$ (resp., $\Theta_U(s, X_O) = T$).

Lemma 1: if *p* be the probability of all the scenarios where $\Theta_L(s, X_O + 1) < \Theta_U(s, X_O)$ (i.e., the process $\{I_t(s, X_O + 1)\}_t$ hits zero for the first time before the process $\{I_t(s, X_O)\}_t$ hits C for the first time), then $G_-(X_O + 1) - G_-(X_O) = -p$.

Proof: First note that if at time *t*, $I_t(s, X_O) = C$ [resp., $I_t(s, X_O + 1) = 0$] then $I_{t'}(s, X_O) = I_{t'}(s, X_O + 1)$ for all $t' \geq t$.

An example of that for the case where $\Theta_L(s, X_O + 1) < \Theta_U(s, X_O)$ is presented in Figure 3. The figure describes two processes: The first has initial inventory level of

X_O=2 and the second has initial inventory level of X_O+1=3. The times of events in the example are marked with a circle. At time t_1 the first pair of bicycles is taken from the station and at time t_2 a second pair is taken. At this point, $I_{t_2}(s,2)=0$. At time t_3 a renter arrives at the station. Due to lack of bicycles in the station for the process that begins with initial inventory level of X_O=2, the first shortage event occurs. For the process that begins with initial inventory level of X_O=3, the last pair of bicycles is taken. From this point on, $I_{t'}(s,X_O)=I_{t'}(s,X_O+1)$ for all $t' \geq t_3$. We can see an example of that when at time t_4 a pair of bicycles is returned to the station.

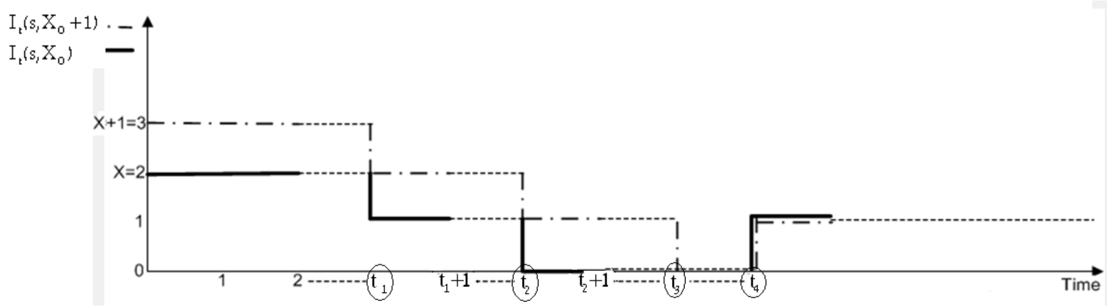

Fig 3. Example of the Replenishment problem - users with no patience

Now we consider the following sets of scenarios:

S_1 The set of all scenarios *s* where $\Theta_L(s,X_O+1) \geq \Theta_U(s,X_O)$, i.e. the first shortage event of $\{I_t(s,X_O)\}_t$ occurs after the first surplus event of $\{I_t(s,X_O+1)\}_t$ or there were no shortage and surplus events.

S_2 The set of all scenarios *s* where $\Theta_L(s,X_O+1) < \Theta_U(s,X_O)$, i.e. the first shortage event of $\{I_t(s,X_O)\}_t$ in the interval $[0,T)$ occurs prior to the first surplus event of $\{I_t(s,X_O+1)\}_t$

Let us denote the number of shortage events in a particular scenario *s* assuming initial inventory level X_O by $g\_(s,X_O)$.

In a member *s* of S_1, $\{I_t(s,X_O)\}_t$ and $\{I_t(s,X_O+1)\}_t$ coincide at the first surplus event. If such an event occurs it occurs before the first shortage event and hence

$$g\_(s,X_O+1)-g\_(s,X_O)=0, \tag{4}$$

i.e., the number of shortage events in $\{I_t(s, X_O)\}_t$ equals the number of shortages in $\{I_t(s, X_O+1)\}_t$.

In member *s* of S_2, $\{I_t(s, X_O)\}_t$ and $\{I_t(s, X_O+1)\}_t$ coincide immediately after the first shortage events and hence

$$g\_(s, X_O+1) - g\_(s, X_O) = -1 \tag{5}$$

i.e., the number of shortage events in $\{I_t(s, X_O)\}_t$ is greater exactly by one than the number of shortages in $\{I_t(s, X_O+1)\}_t$.

Now, to calculate $G\_(X_O+1) - G\_(X_O) \equiv \int_{s\in S_1\cup S_2} g\_(s, X_O+1)\cdot f(s)\,ds - \int_{s\in S_1\cup S_2} g\_(s, X_O)\cdot f(s)\,ds$ (where f(s) is the density function over all scenarios):

$$\begin{aligned}
&G_(X_O+1) - G_(X_O) = E_S\left[g_(s, X_O+1)\right] - E_S\left[g_(s, X_O)\right] = E_S\left[g_(s, X_O+1) - g_(s, X_O)\right] \\
&= E_S\left[g_(s, X_O+1) - g_(s, X_O) \mid s\in S_1\right]\cdot P(s\in S_1) + E_S\left[g_(s, X_O+1) - g_(s, X_O) \mid s\in S_2\right]\cdot P(s\in S_2) \\
&= 0 + (-1)\cdot P(s\in S_2) = -p
\end{aligned} \tag{6}$$

The first equality is by the definition of $G\_(X_O)$. The second is due to linearity property of the expectation operator. The third equality is obtained by conditioning on all possible sets of scenarios that were introduced before. We then substitute (4) and (5) in the expectation and exploit the definition of the probability *p*. ■

Lemma 2: For any integer $0 \le X_O \le C-2$ the following inequality holds $\left[G\_(X_O+1) - G\_(X_O)\right] \le \left[G\_(X_O+2) - G\_(X_O+1)\right]$.

Proof: By Lemma 1 it is enough to show that the probability that $\Theta_L(s, X_O+1) \le \Theta_U(s, X_O)$ is not smaller than the probability that $\Theta_L(s, X_O+2) \le \Theta_U(s, X_O+1)$. To see this, observe that in any scenario where $\Theta_L(s, X_O+2) \le \Theta_U(s, X_O+1)$ also $\Theta_L(s, X_O+1) < \Theta_U(s, X_O)$. This is due to the fact that in all the scenarios where the process $\{I_t(s, X_O+1)\}_t$ hits zero before it hits C, the process $\{I_t(s, X_O)\}_t$ does so also. ■

Lemma 3 let $\tilde{p}$ be the probability of all scenarios where $\Theta_L(s, X_O+1) > \Theta_U(s, X_O)$, then $G_+(X_O+1) - G_+(X_O) = \tilde{p}$.

Proof: consider the following sets of scenarios:

S_1 The set of all scenarios *s* where $\Theta_L(s, X_O+1) > \Theta_U(s, X_O)$, $t \in [0, T)$.

S_2 The set of all scenarios *s* where $\Theta_L(s, X_O+1) \le \Theta_U(s, X_O)$.

Let us denote the number of shortages in scenario *s* assuming initial inventory level X_O by $g_+(s, X_O)$.

In member *s* of S_1, $\{I_t(s, X_O)\}_t$ and $\{I_t(s, X_O+1)\}_t$ coincide immediately after the first surplus event and hence

$$g_+(s, X_O+1) - g_+(s, X_O) = 1 \tag{7}$$

In member *s* of S_2, $\{I_t(s, X_O)\}_t$ and $\{I_t(s, X_O+1)\}_t$ coincide at the first shortage event and hence

$$g_+(s, X_O+1) - g_+(s, X_O) = 0 \tag{8}$$

Now to calculate

$$G_+(X_O+1) - G_+(X_O) \equiv \int\limits_{s \in S_1 \cup S_2} g_+(s, X_O+1) \cdot f(s)\, ds - \int\limits_{s \in S_1 \cup S_2} g_+(s, X_O) \cdot f(s)\, ds \; : \tag{9}$$

$$\begin{aligned} G_+(X_O+1) - G_+(X_O) &= E_S\left[g_+(s, X_O+1)\right] - E_S\left[g_+(s, X_O)\right] = E_S\left[g_+(s, X_O+1) - g_+(s, X_O)\right] \\ &= E_S\left[g_+(s, X_O+1) - g_+(s, X_O) / s \in S_1\right] \cdot p(s \in S_1) + E_S\left[g_+(s, X_O+1) - g_+(s, X_O) / s \in S_1\right] \cdot p(s \in S_2) \\ &= 1 \cdot p(s \in S_1) + 0 = \tilde{p} \end{aligned}$$

The equalities are obtained in a similar way as in Lemma 1. ■

Lemma 4: For any integer $0 \le X_O \le C-2$ the following inequality holds $\left[G_+(X_O+1) - G_+(X_O)\right] \le \left[G_+(X_O+2) - G_+(X_O+1)\right]$

Proof: By Lemma 3 it is enough to show that the probability that $\Theta_L(s, X_O+2) > \Theta_U(s, X_O+1)$ is not smaller than the probability that $\Theta_L(s, X_O+1) > \Theta_U(s, X_O)$. To see this, observe that in any scenario where $\Theta_L(s, X_O+1) > \Theta_U(s, X_O)$ also $\Theta_L(s, X_O+2) > \Theta_U(s, X_O+1)$. ■

Observe that a function $f:\{0,\ldots,C\}\to\Re$ is convex if and only if the series of differences $\{f(i+1)-f(i)\}_{i\in\{0,\ldots,C-1\}}$ is non-decreasing. This is equivalent to second order conditions in the continuous case. The next corollary follows directly from the above observation together with Lemmas 2 and 4.

Corollary 1: The functions $G_-(X_O)$ and $G_+(X_O)$ are convex.

We are ready to prove the main result of this section.

Theorem 1: The expected total penalty function $F(X_O)$ is convex.

Proof: Recall that for any $\alpha,\beta\geq 0$ and any convex functions $g(x), h(x)$ defined over the same domain, the function $f(x)=\alpha g(x)+\beta h(x)$ is also convex. Now our claim follows directly from this observation and Corollary 5. ■

We conclude by establishing bounds on the "marginal saving" obtained by adding or removing a single bicycle in the station.

Corollary 2: $-p^A\leq F(X_O+1)-F(X_O)\leq h^A$.

Proof: let us look at the scenario where $\Theta_L(s,X_O+1)>\Theta_U(s,X_O)$. Let us denote the probability for such a scenario by p_1. Using the same arguments as in Lemma 1 and 3 we know that:

$$g_+(s,X_O+1)-g_+(s,X_O)=1 \tag{10}$$

$$g_-(s,X_O+1)-g_-(s,X_O)=0 \tag{11}$$

Now let us look at the scenario where $\Theta_L(s,X_O+1)<\Theta_U(s,X_O)$. Let us denote the probability for such a scenario by p_2. Using the same arguments as in Lemma 1 and 3 we know that:

$$g_+(s,X_O+1)-g_+(s,X_O)=0 \tag{12}$$

$$g_-(s,X_O+1)-g_-(s,X_O)=-1 \tag{13}$$

Note that if no shortage events of $\{I_t(s,X_O)\}_t$ and no surplus events of $\{I_t(s,X_O+1)\}_t$ occurred during the interval $[0,T]$ then:

$$g_+(s,X_O+1)-g_+(s,X_O)=0 \tag{14}$$

$$g_-(s,X_O+1)-g_-(s,X_O)=0 \tag{15}$$

Now,

$$\begin{aligned} F(X_O+1)-F(X_O) &= p^A\cdot\left[G_-(X_O+1)-G_-(X_O)\right]+h^A\cdot\left[G_+(X_O+1)-G_+(X_O)\right] \\ &= p_1\cdot h^A-p_2\cdot p^A \end{aligned} \tag{16}$$

The results are based on the proofs of Lemma 1 and 3.

Thus,

$$-p^A\le F(X_O+1)-F(X_O)\le h^A\,. \blacksquare \tag{17}$$

This result implies that by adding another pair of bicycles to the initial inventory level we can reduce the expected total penalty up to one shortage penalty or increase it up to one surplus penalty.

3.2.2 Users with infinite patience

Now let us consider the case where renters (resp., returners) who arrive at an empty station (resp., full) wait until they can rent (resp. return) the bicycles.

In this case, we measure the expected total penalty by the expected total waiting time of customers in the station. When the station is replenished at time T, all the users that are waiting at the station are served (note that this assumption, although reasonable, is not necessarily true in practice).

Note that in this case there is no point in counting the number of "lost sales" because users of the system do not abandon, consequently, if the system is stable everyone is served eventually.

The expected total waiting time function is given by:

$$F(X_O)=p^W\cdot H_-(X_O)+h^W\cdot H_+(X_O) \tag{18}$$

where $H_{-}(X_O)$ is the renters expected waiting time for bicycles during the interval $[0,T]$ as a function of the initial inventory at time 0, and $H_{+}(X_O)$ is the returners expected waiting time for vacant lockers during this interval.

As in §3.2.1, the state of the system given initial inventory level X_O and scenario *s* at time *t* is denoted by $I_t(s,X_O)$. Note, however that in this case $I_t(s,X_O)$ may be negative or greater than C. This represents situations where users are waiting for bicycles or vacant lockers. For example, if $I_t(s,X_O)=-3$, then given initial inventory level of X_O and scenario *s*, there are 3 renters waiting for bicycles at time *t*. Similarly, if $I_t(s,X_O)=C+3$, then given initial inventory level of X_O and scenario *s*, there are 3 returners waiting for vacant lockers at time *t*.

An example for this case appears in Figure 4. The diagram describes two processes: The first has initial inventory level of X_O=2 and the second has initial inventory level of X_O+1=3. The difference between the states of these processes is always one. At time t_1 the first pair of bicycles is taken from the station, at time t_2 a second pair is taken. At time t_3 another renter arrives at the station, where in process $\{I_t(s,2)\}_t$ there are no bicycles. At this point, $I_{t_3}(s,2)=-1<0$. At time t_4 process $\{I_t(s,3)\}_t$ also receives a negative value where $I_{t_4}(s,3)=-1$. As a result, the difference between the two processes stays fixed.

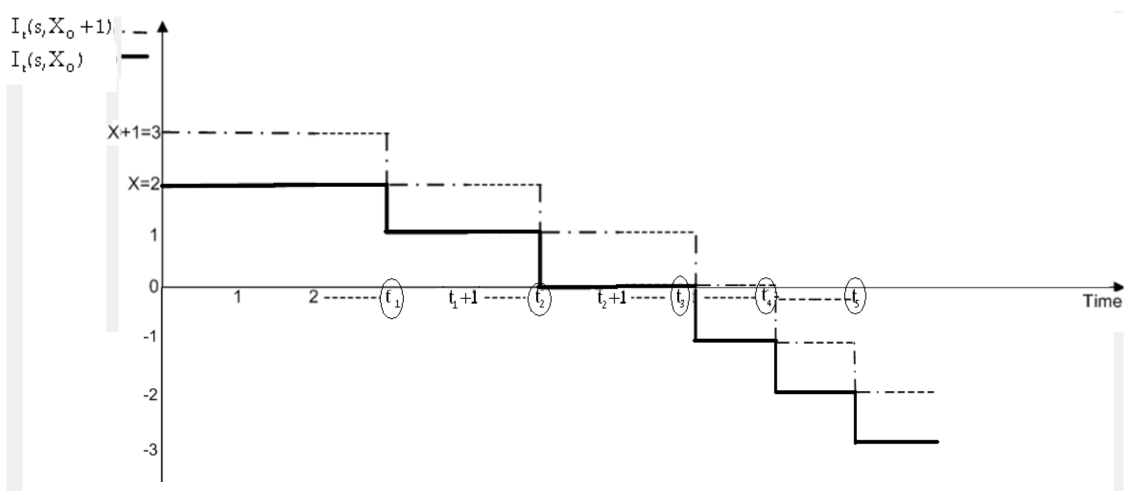

Fig 4. Example of the Replenishment problem - users with infinite patience

Lemma 5: For any integer $0 \le X_O \le C-2$ the following inequality holds $[H_-(X_O+1)-H_-(X_O)] \le [H_-(X_O+2)-H_-(X_O+1)]$.

Proof: Let us fix a scenario of renters arrivals *s* and let us denote the expected total waiting time of renters under this scenario given initial inventory X_O by $h_-(s,X_O)$. Note that the difference $h_-(s,X_O)-h_-(s,X_O+1)$ [resp., $h_-(s,X_O+1)-h_-(s,X_O)$] equals exactly to the total time during which $I_t(s,X_O)<0$ [resp., $I_t(s,X_O+1)>C$)].
That is

$$h_-(s,X_O)-h_-(s,X_O+1)=\int_0^T \mathbb{I}_{\{I_t(s,X_O)<0\}}\,dt \tag{19}$$

where

$$\mathbb{I}_{\{I_t(s,X_O)<0\}}=\begin{cases}1 & ,I_t(s,X_O)<0\\ 0 & ,\text{else}\end{cases}.$$

Similarly

$$h_-(s,X_O+1)-h_-(s,X_O+2)=\int_0^T \mathbb{I}_{\{I_t(s,X_O+1)<0\}}\,dt \tag{20}$$

Now, note that $\mathbb{I}_{\{\{I_t(s,X_O)\}_t<0\}} \ge \mathbb{I}_{\{\{I_t(s,X_O+1)\}_t<0\}}$ because $\{I_t(s,X_O+1)\}_t=\{I_t(s,X_O)\}_t+1$ and hence

$$h_-(s,X_O)-h_-(s,X_O+1)\ge h_-(s,X_O+1)-h_-(s,X_O+2) \tag{21}$$

and so

$$h_-(s,X_O+1)-h_-(s,X_O)\le h_-(s,X_O+2)-h_-(s,X_O+1) \tag{22}$$

Now, since this is true for every scenario, the inequality holds for the expectation as well. ■

Lemma 6: For any integer $0 \le X_O \le C-2$ the following inequality holds $[H_+(X_O+1)-H_+(X_O)] \le [H_+(X_O+2)-H_+(X_O+1)]$.

Proof: Let us fix a scenario of returners arrival *s* and let us denote the expected total waiting time of returners under this scenario given initial inventory X_O by $h_+(s,X_O)$.

Note that the difference $h_-(s,X_O+1)-h_-(s,X_O)$ equals exactly to the total time during which $I_t(s,X_O)<0$ [resp., $I_t(s,X_O+1)>C$)].

That is,

$$h_+(s,X_O+1)-h_+(s,X_O)=\int_0^T \mathbb{I}_{\{I_t(s,X_O+1)>C\}}\,dt \tag{23}$$

Similarly

$$h_+(s,X_O+2)-h_+(s,X_O+1)=\int_0^T \mathbb{I}_{\{I_t(s,X_O+2)>C\}}\,dt \tag{24}$$

Now, note that $\mathbb{I}_{\{\{I_t(s,X_O+2)\}_t>C\}}\geq \mathbb{I}_{\{\{I_t(s,X_O+1)\}_t>C\}}$ because $\{I_t(s,X_O+1)\}_t+1=\{I_t(s,X_O+2)\}_t$ and hence

$$h_+(s,X_O+1)-h_+(s,X_O)\leq h_+(s,X_O+2)-h_+(s,X_O+1) \tag{25}$$

Now, since this is true for every scenario, the inequality holds for the expectation as well. ■

Corollary 3: As we previously mentioned, a function $f:\{0,...,C\}\rightarrow\Re$ is convex if and only if the series of differences $\{f(i+1)-f(i)\}_{i\in\{0,...,C-1\}}$ is non-decreasing. Thus, the functions $H_-(X_O)$ and $H_+(X_O)$ are convex.

Theorem 2: The expected total penalty function $F(X_O)$ is convex

Proof: Recall that for any $\alpha,\beta\geq 0$ and any convex functions $g(x),h(x)$ defined over the same domain, the function $f(x)=\alpha g(x)+\beta h(x)$ is also convex. Now our claim follows directly from this observation and Corollary 10. ■

3.2.3 Users with finite patience

Previously we saw that the two extreme cases of user behavior models, i.e., when all users are impatient and when all users have unlimited patience, resulting in convexity of the objective function. We point out that the convexity proofs above make no assumptions on the nature of the arrival process *A* and *B*. In particular, our results are valid for non homogenous arrival processes and for non Markovian ones.

It is interesting to point out that the convexity property does not hold for more general user behavior models, i.e., when some users abandon and others choose to queue at the station.

Consider for example the following deterministic scenario *s* - two arrivals of renters and no arrivals of returners. At time *t*=1, an impatient renter arrives and at time *t*=2 a renter with unlimited patience arrive. It is easy to check that:

- With $X_O = 2$ the station will not be penalized for waiting time of users and for abandonments.
- With $X_O = 1$ the station will accumulate total waiting of T-2 over the planning horizon and there will be no abandonment.
- With $X_O = 0$ the station will accumulate total waiting of T-2 over the planning horizon and one renter will abandon

Now, if we denote the expected total penalty function of renters under this scenario given initial inventory X_O by $f(s, X_O)$, then $f(s,1) - f(s,0) = -p^A$ and $f(s,2) - f(s,1) = -p^W \cdot (T-2)$. Clearly it is possible to choose parameters values such that $f(s,1) - f(s,0) > f(s,2) - f(s,1)$, which implies that $f(s, X_O)$ is not convex. Nevertheless, we believe that with most real life stochastic demand processes, the objective function is convex in the relevant domain, i.e. the marginal contribution of an additional bicycle at the station is decreasing.

3.3 The Replenishment problem with Poisson arrival processes and users with no patience

In this section we present an analysis of a station under the following assumptions:

- The users have no patience, i.e., abandon immediately if bicycles or lockers are not available.
- The renters and returners arrive at the station according to Poisson processes. These processes are not necessarily homogenies over time and their rates are represented by step functions (the rate in each time interval is stationary).
- The arrival process of users is unaffected by the system state. In reality users may decide not to arrive at the station based on prior information about the system state obtained remotely, e.g., using the system's web site. However, we assume

that such users should be considered as abandonees and hence our analysis is not affected.

One can think of a station under these assumptions as an M/M/1/C queuing model with a service process equal to the arrival process of renters and arrival process equal to arrival process of returners. This system is depicted as a Markov chain in Figure 5. Recall, however, that in order to solve the Replenishment problem a transient analysis is needed. Note that the Poisson assumption is reasonable due to the fact that users arrive at the system independently.

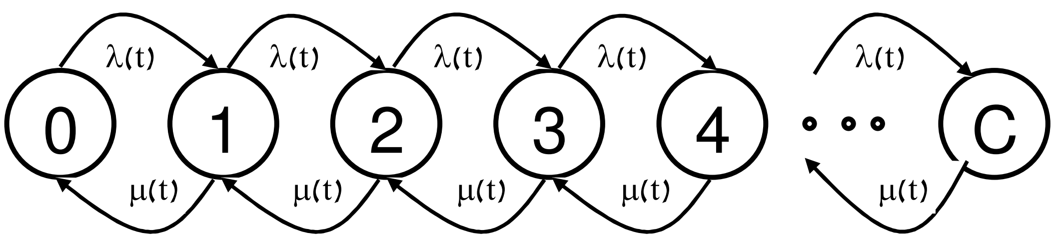

Fig 5. The Replenishment problem with Poisson arrival processes and users with no patience as death and birth process

The rest of this section is organized as follows: In §3.3.1 we formulate the problem in terms of the above Markov chain. In §3.3.2 we develop an approximation procedure for this function and set an experimental study where we set up the algorithm's parameters, validate it and analyze its computing time. Next we compare a few of our results to calculations based on steady state analysis and illustrate its problematic nature for the Replenishment problem. Finally, we draw some operational insights from the results and check their sensitivity to inaccuracies in the input.

3.3.1 Markov Chain Model for the Replenishment Problem

Let $\pi(t)$ be the *transition probability matrix* for the interval [0,t]. Then, the formulation of the objective function is as follows:

$$F\left(p^A, h^A, \mu(t), \lambda(t), C, T, X_O\right) = \int_{t=0}^{T} \left[p^A \cdot \mu(t) \cdot \pi_{X_o,0}(t) + h^A \cdot \lambda(t) \cdot \pi_{X_o,C}(t) \right] dt \quad (26)$$

where, $\pi_{i,j}(t)$ is the probability of the station to switch from state *i* at time 0 to state *j* at time *t* (calculation procedure for these probabilities is presented in §3.2.2). In particular, $\pi_{X_o,0}(t)$ is the probability that a station will be empty at time *t* assuming initial inventory X_O, and $\pi_{X_o,C}$ is the probability of having no vacant lockers at this time. The term $\lambda(t)\cdot\pi_{X_o,C}(t)$ [resp., $\mu(t)\cdot\pi_{X_o,0}(t)$] represents the rate of returners (resp., renters) abandonments at time *t*. The integration over the penalty accumulation rates yields the expected total penalty during the time period [0,T], representing the interval between consecutive visits of the repositioning truck.

In the term of (26), the Replenishment problem is defined as

$$\min_{X_o\in\{0,\ldots,C\}} F\left(p^A, h^A, \mu(t), \lambda(t), C, T, X_O\right) \tag{27}$$

3.3.2 Approximation procedure

Next we present a numerical approximation procedure for calculating $F\left(p^A, h^A, \mu(t), \lambda(t), C, T, X_O\right)$ as in (26) for a given set of parameters and for all values of $X_O \in \{0,\ldots,C\}$ and hence to solve (27). Note that the value of the penalty function is of interest for all values of initial inventory 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 at all stations to their ideal values.

The approximation is done by discretizing the time of the planning horizon into a short period of length *d*. Note however that it is important to make sure that the length of the intervals in which the arrival rates are given (and during each one of them the rates are assumed to be constant) is divisible by *d*. We also assume that *T* is divisible by *d*. Hence, T/d is the number of time periods until the next visit of the truck. We start with straight forward discretization of the integral in (26). That is,

$$F\left(p^A, h^A, \mu(t), \lambda(t), C, T, X_O\right) \approx \sum_{t=1}^{T/d}\left[p^A\cdot\mu(t)\cdot\pi_{X_o,0}(t) + h^A\cdot\lambda(t)\cdot\pi_{X_o,C}(t)\right] \tag{28}$$

In order to carry out the summation (28), one needs to calculate the transition probability matrix for all integer times $t = 1,\ldots,T/d$. To this end we define and calculate the *single period transition probability P(t)* which is the transition probability matrix from the beginning of a period to its end. Using these matrices, one can obtain $\pi(t)$ recursively. That is,

$$\pi(t) = \pi(t-1)\cdot P(t) \tag{29}$$

Next, P(t) can be approximated as follows (Ross 1997):

$$P(t) = e^{Rt} = \lim_{M\to\infty}\left(I + R\frac{t}{M}\right)^M \tag{30}$$

where *I* is the *Identity matrix* and matrix *R* is defined as follows for each pair *(i,j)*:

$$R_{ij} = \begin{cases} q_{ij} & ,if \quad i \neq j \\ -\sum_k q_{ik}, \ \forall k \neq i & ,else \end{cases} \tag{31}$$

where q_{ij} is the *instantaneous transition rate* from *i* to *j*.

If for example C=2 and $\mu(t) = \mu$, $\lambda(t) = \lambda$, matrix *R* would be:

$$R = \begin{pmatrix} -\lambda & \lambda & 0 \\ \mu & -(\lambda+\mu) & \lambda \\ 0 & \mu & -\mu \end{pmatrix}$$

In this example there are three possible states (0,1,2). To illustrate the construction of the matrix *R,* let us examine the second row (state 1). The *Instantaneous transition rate* to states 0 and 2 ($q_{10,}q_{12}$ resp.) is μ and λ respectively and hence $R_{10} = \mu,\ \ R_{11} = -(\mu+\lambda),\ \ R_{12} = \lambda$.

The approximation procedure is based on the formulation of the objective function and the approximation formula calculation of the transition probabilities given above. The calculation performed by the procedure is efficient due to its use of matrix calculation. It uses shortage and surplus penalty vectors for every possible X_O, so that the calculation for all possible initial states is done by only one calculation.

The algorithm was implemented in Matlab, see code in Appendix A.

3.3.2.1 Control parameters

From (28) and (30) we conclude that the objective function values are mainly affected by two parameters:

d	The length of each time period over the planning Horizon.
M	The value that is used for approximating the limit from the transition probabilities matrix approximation formula.

The value of *d* determines the level of discretization being used for the objective function approximation. As mentioned, T/d is the number of time periods until the next visit of the truck. The smaller *d* is, the finer the discretization level is and the better the approximation is. It is required to find such a value for *d* so that the approximation will be good enough and the calculation time will be reasonable.

For the M parameter, the higher M is, the more accurate the transition probabilities matrix approximation is. In (30) we showed that theoretically $M \to \infty$. There is a need to find a reasonable value of M (in the computation time sense) so that the value of each cell of the matrix $P(t)$ will be approximately converged, i.e. if we were to increase the value of M, the outcome would be approximately the same.

3.3.2.2 Experimental study

We conducted an experimental study in order to calibrate, verify and validate the approximation procedure. The experimental study is organized as follows: in §3.3.2.3 the experiment environment is defined and test problems are presented. In §3.3.2.4 we look for proper values for the control parameters (calibration and verification). This is done so we can produce an approximately converged solution on the one hand, and so that the computing time will be reasonable, on the other hand. In §3.3.2.5 the procedure's results validity is tested by a comparison to the results of a simulation model. This study is important as a performance evaluation of the algorithm. Next in §3.3.2.6 we discuss the procedure and the simulation computing time. This concerns us due to the fact that this operational decision occurs in a dynamic environment and it is essential to have fast performances. In §3.3.2.7 we draw some operational insights concerning the nature of the optimal initial inventory level of the test problems considered. In §3.3.2.8 we perform a numerical comparison between steady state analysis and our procedure. Finally, in §3.3.2.9 we perform a sensitivity analysis of the results by exploring the influence of noise in the system.

3.3.2.3 Experimental Design

To evaluate the presented approximation procedure, several test problems were solved. These test problems represent a variety of renters and returners demand patterns in a realistic size station (30 lockers in a station). The problems that were examined present several possible cases of demand processes in a single station. A short description of each problem appears in Table 1 and graphs which present the renters and returners demand patterns are shown in Figures 6-10.

The renters and returners demand data describes the rate of arrival events in time intervals. The basic time interval used in these experiments is fifteen minutes.

We assume that the bicycles replenishment operation is carried out between midnight and 6am, while the system is idle. The demand is then realized during an eighteen hours horizon, 6am until midnight. While Bike-Sharing systems operate 24/7, the actual demand observed during the nights of weekdays in most systems is negligible. Thus, the renters and returners demand data is given for an eighteen hour planning horizon (in 72 time intervals of fifteen minutes). The actual data used in our experiment is presented in Appendix B.

We assume that the shortage penalty and the surplus penalty are equal to 1 ($p^A = h^A = 1$). That is, the expected total penalty equals the number of renters arriving to an empty station and the number of returners arriving to a full capacity station.

The results are being tested by their expected total penalty for each initial X_O and more importantly, by the optimal X_O selected.

Table 1. Test problems description

Problem	Description
Homogenous – symmetric	Homogenous and identical rate of demand (85 renters and returners arrivals each)
Homogenous – non symmetric	Homogenous and not identical rate of renters and returners demand. The renters demand is 1.2 times stronger than the returners demand (102 renters arrivals and 85 returners arrivals)
Peaks – symmetric	Symmetric peaks - Morning peak of renters demand and afternoon peak of returners demand. Typical for a suburban station (85 renters and 85 returners)
Peaks – non symmetric	Non symmetric peaks - Morning peak of renters demand and afternoon peak of returners demand. The afternoon peak is twice as strong as the morning peak (85 renters arrivals and 170 returners arrivals)
Random symmetric	Random demand processes (85 renters and 85 returners)

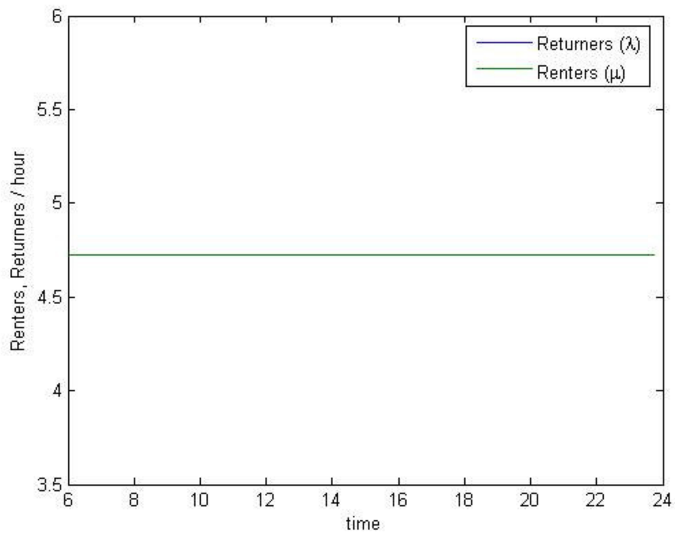

Fig 6. Homogenous – symmetric renters and returners demand

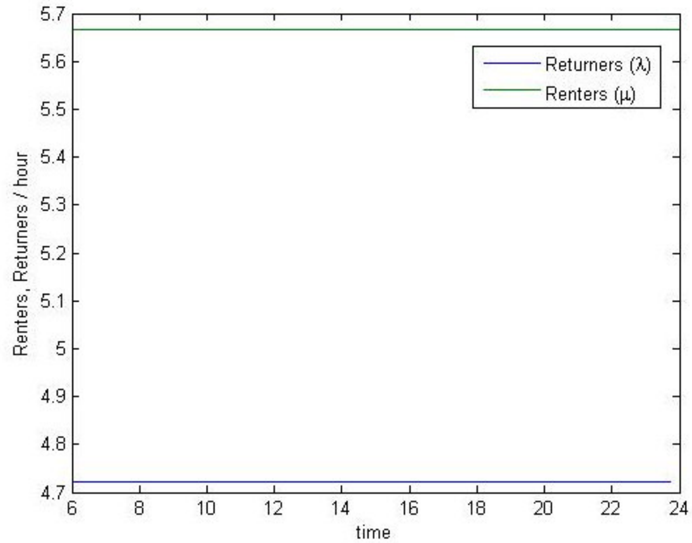

Fig 7. Homogenous – non symmetric renters and returners demand

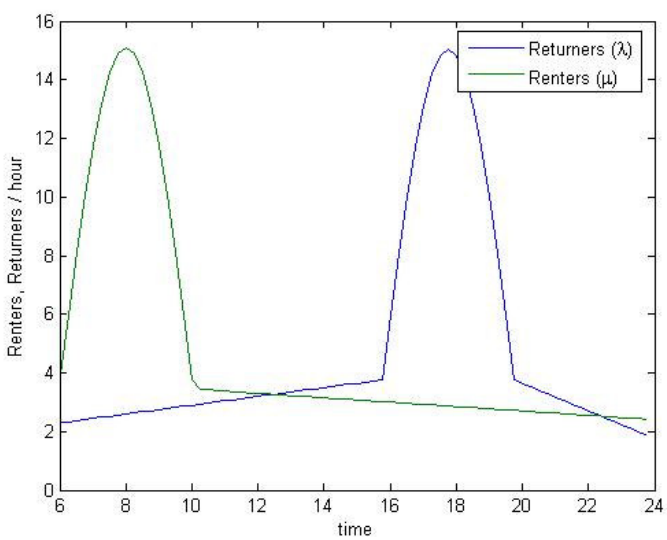

Fig 8. Peaks – symmetric renters and returners demand

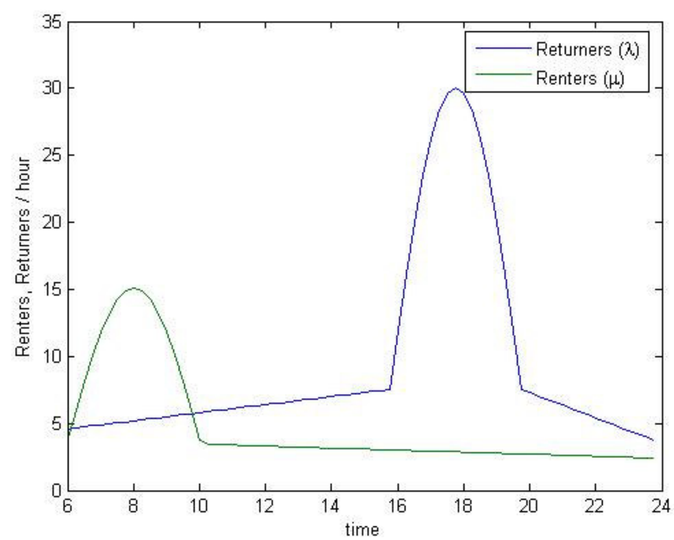

Fig 9. Peaks – non symmetric renters and returners demand

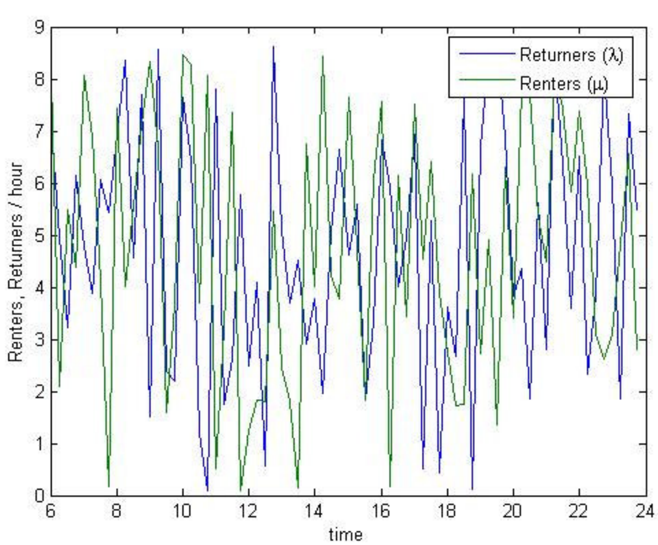

Fig 10. Random symmetric renters and returners demand

3.3.2.4 Convergence verification for the transition probability matrix

In §3.3.2.1 the need to find a proper value for the control parameter M in the approximation procedure was explained. The target is to find such a value that will produce an approximately converged solution, on the one hand, and will be computed in reasonable time, on the other hand.

A set of experiments were conducted considering the test problems using the approximation procedure using M values of 300 and 1000. A comparison of the results has been made for four of the problems with 1 and 15 minutes discretization levels (namely, *d* =1 and *d*=15). As previously mentioned, the data basic time interval is fifteen minutes. Hence, when the discretization level is less than 15 minutes, each rate is divided by $15/d$ and the result will appear $15/d$ times. For example, if the discretization level is 5 minutes (*d*=5), then $15/d = 3$ and each rate will appear 3 times. In this case, the value of T/d is 216. A proper procedure has been written for this purpose (see Appendix C).

The results are shown in Tables 2-7 and in Figures 6-11. As one can see, the difference between the tested problems results with M = 300 and M = 1000 is negligible for all the tested problems. More importantly, the optimal X_O was the same in each comparison that was made. The most significant differences were found in the *homogenous non-symmetric* problem with a discretization level of 1 minute. Nevertheless, even in this case the average difference is 1.17% (with standard deviation of 0.91%) which is fairly small. Moreover, the optimal policy is also identical in this case.

From the results, we conclude that M = 300 is a proper value to use in the model, so that the values in the transition probabilities matrix will approximately converge. Note that we prefer using M = 300 in order to achieve better computing time.

The next stage will be the quality validation of the results the model produces.

Table 2. Homogenous – symmetric problem with 15 minutes discretization: M=300 vs. M=1000

X_o	$F(X_o)$ - M = 300	$F(X_o)$ - M = 1000	Difference (%)
0	9.6193	9.6203	0.010395767
1	9.076	9.0761	0.001101807
2	8.3017	8.3016	0.001204573
3	7.516	7.5158	0.00266099
4	6.7873	6.7872	0.00147334
5	6.1258	6.1256	0.00326488
6	5.5307	5.5306	0.001808089
7	5.0011	5.0011	0
8	4.5363	4.5363	0
9	4.1355	4.1354	0.002418087
10	3.7977	3.7977	0
11	3.5225	3.5224	0.002838893
12	3.3092	3.3091	0.003021878
13	3.1571	3.157	0.003167464
14	3.0661	3.066	0.003261472
15	**3.0356***	**3.03568***	0
16	3.0661	3.066	0.003261472
17	3.1571	3.157	0.003167464
18	3.3092	3.3091	0.003021878
19	3.5225	3.5224	0.002838893
20	3.7977	3.7977	0
21	4.1355	4.1354	0.002418087
22	4.5363	4.5363	0
23	5.0011	5.0011	0
24	5.5307	5.5306	0.001808089
25	6.1258	6.1256	0.00326488
26	6.7873	6.7872	0.00147334
27	7.516	7.5158	0.00266099
28	8.3017	8.3016	0.001204573
29	9.076	9.0761	0.001101807
30	9.6193	9.6203	0.010395767

Table 3. Homogenous – non symmetric problem with 15 minutes discretization: M=300 vs. M=1000

X_o	$F(X_o)$- M = 300	$F(X_o)$- M = 1000	Difference (%)
0	21.0749	21.076	0.005219479
1	20.5308	20.531	0.000974146
2	19.6721	19.6717	0.002033337
3	18.719	18.7187	0.00160265
4	17.7617	17.7614	0.001689028
5	16.8196	16.8192	0.002378178
6	15.8951	15.8948	0.001887374
7	14.9899	14.9896	0.002001348
8	14.1059	14.1056	0.00212677
9	13.2453	13.245	0.002264954
10	12.4102	12.41	0.001611578
11	11.6032	11.603	0.001723662
12	10.8265	10.8262	0.002770979
13	10.0824	10.0823	0.000991827
14	9.3739	9.3737	0.002133584
15	8.7034	8.7032	0.002297953
16	8.074	8.0739	0.001238543
17	7.489	7.4888	0.002670584
18	6.9517	6.9516	0.001438497
19	6.4665	6.4664	0.001546432
20	6.038	6.0379	0.001656178
21	5.6716	5.6715	0.001763171
22	5.3739	5.3739	0
23	5.1529	5.1529	0
24	5.0181	5.0181	0
25	**4.9812***	**4.9812***	0
26	5.0568	5.0569	0.001977535
27	5.2624	5.2624	0
28	5.6096	5.6096	0
29	6.0597	6.0601	0.006600987
30	6.4341	6.4353	0.018650627

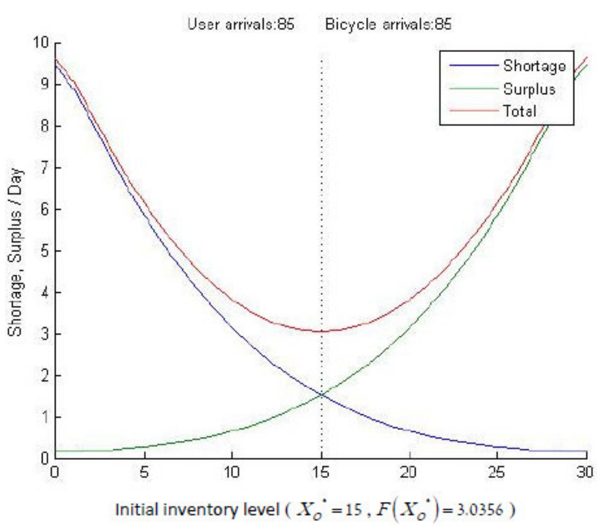

Fig 11. Homogenous – symmetric problem with 15 minutes and M=1000

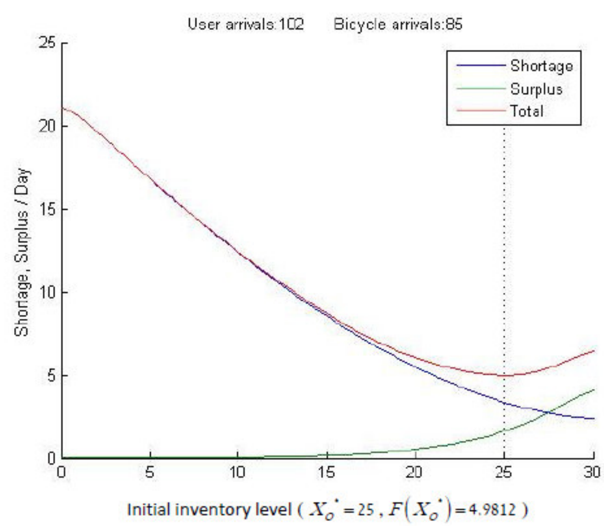

Fig 12. Homogenous – non symmetric problem with 15 minutes and M=1000

Table 4. Homogenous – non symmetric problem with 1 minute discretization: M=300 vs. M=1000

X_o	$F(X_o)$ - M = 300	$F(X_o)$ - M = 1000	Difference (%)
0	21.0749	21.4982	2.008550456
1	20.5308	20.5534	0.110078516
2	19.6721	19.5728	0.504775799
3	18.719	18.603	0.619691223
4	17.7617	17.6469	0.646334529
5	16.8196	16.706	0.675402507
6	15.8951	15.7824	0.709023536
7	14.9899	14.8782	0.745168413
8	14.1059	13.9954	0.783360154
9	13.2453	13.1362	0.823688403
10	12.4102	12.3027	0.866222946
11	11.6032	11.4975	0.910955598
12	10.8265	10.7227	0.958758602
13	10.0824	9.9809	1.006704753
14	9.3739	9.2747	1.058257502
15	8.7034	8.6069	1.108762093
16	8.074	7.9803	1.160515234
17	7.489	7.3981	1.213780211
18	6.9517	6.864	1.26156192
19	6.4665	6.382	1.30673471
20	6.038	5.9566	1.348128519
21	5.6716	5.5936	1.375273291
22	5.3739	5.2993	1.388191072
23	5.1529	5.0815	1.385627511
24	5.0181	4.9497	1.363065702
25	**4.9812***	**4.9156***	1.31494419
26	5.0568	4.9938	1.245847176
27	5.2624	5.2013	1.161067194
28	5.6096	5.5601	0.882415859
29	6.0597	6.0965	0.60729079
30	6.4341	6.8057	5.775477534

Table 5. Peaks – symmetric problem with 15 minutes discretization: M=300 vs. M=1000

X_o	$F(X_o)$ - M = 300	$F(X_o)$ - M = 1000	Difference (%)
0	51.296	51.2956	0.000779788
1	50.7443	50.7429	0.002758931
2	49.9454	49.9435	0.003804154
3	49.0827	49.0805	0.004482231
4	48.1959	48.1934	0.005187163
5	47.2921	47.2894	0.005709199
6	46.3749	46.372	0.006253383
7	45.4468	45.4437	0.006821162
8	44.5094	44.5062	0.007189493
9	43.5639	43.5605	0.007804627
10	42.6111	42.6075	0.008448503
11	41.6514	41.6478	0.008643167
12	40.6855	40.6817	0.009339937
13	39.7137	39.7098	0.009820289
14	38.7363	38.7324	0.010068076
15	37.7538	37.7498	0.01059496
16	36.7664	36.7625	0.010607511
17	35.7749	35.7709	0.011181024
18	34.7797	34.7758	0.011213438
19	33.782	33.7781	0.01154461
20	32.783	32.7793	0.011286337
21	31.7847	31.7811	0.011326204
22	30.7892	30.7858	0.011042833
23	29.7997	29.7967	0.010067215
24	28.8204	28.8178	0.009021388
25	27.8569	27.8548	0.007538527
26	26.919	26.9173	0.006315242
27	26.0271	26.0262	0.003457934
28	25.2329	25.2326	0.001188924
29	24.6472	24.6476	0.001622902
30	**24.3926***	**24.3939***	0.005329485

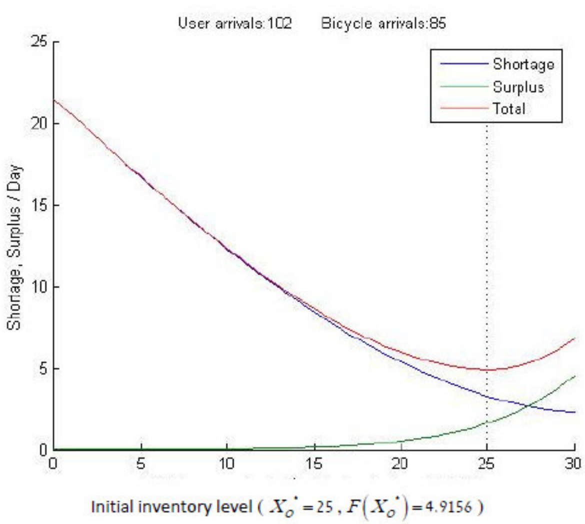

Fig 13. Homogenous – non symmetric problem with 1 minute and M=1000

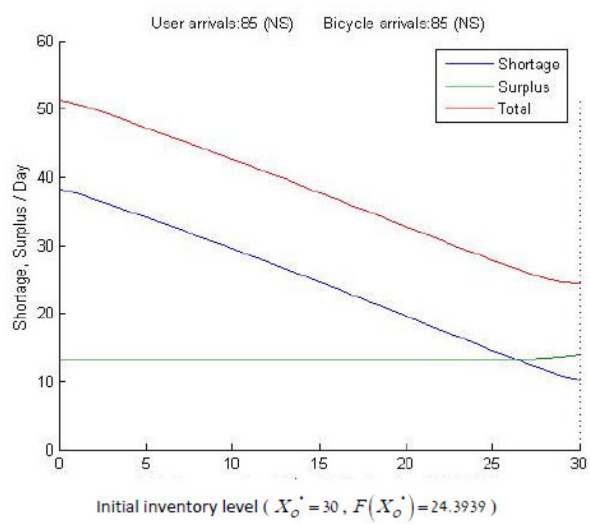

Fig 14. Peaks – symmetric problem with 15 minutes and M=1000

Table 6. Peaks – non symmetric problem with 15 minutes discretization: M=300 vs. M=1000

X_o	$F(X_o)$ - M = 300	$F(X_o)$ - M = 1000	Difference (%)
0	108.5834	108.5803	0.002854948
1	107.9959	107.992	0.003611248
2	107.1746	107.1702	0.00410545
3	106.29	106.2853	0.004421865
4	105.3818	105.377	0.004554866
5	104.4583	104.4533	0.004786599
6	103.5237	103.5185	0.005023004
7	102.5812	102.576	0.005069155
8	101.6337	101.6284	0.005214806
9	100.6842	100.6789	0.005263984
10	99.7357	99.7305	0.00521378
11	98.7922	98.7869	0.005364796
12	97.8577	97.8527	0.00510946
13	96.9375	96.9327	0.004951644
14	96.0376	96.0331	0.004685665
15	95.1647	95.1604	0.004518482
16	94.3261	94.3222	0.004134593
17	93.5299	93.5266	0.003528283
18	92.7847	92.7819	0.003017739
19	92.0992	92.0969	0.002497307
20	91.4821	91.4804	0.001858287
21	90.9416	90.9406	0.001099607
22	90.4859	90.4854	0.000552572
23	90.1224	90.1226	0.00022192
24	89.8591	89.8599	0.000890283
25	89.7054	89.7067	0.001449188
26	**89.6753***	**89.677***	0.001895728
27	89.7903	89.7923	0.002227412
28	90.0732	90.0755	0.002553479
29	90.4931	90.4957	0.002873147
30	90.8529	90.8564	0.003852381

Table 7. Peaks – non symmetric problem with 1 minute discretization: M=300 vs. M=1000

X_o	$F(X_o)$ - M = 300	$F(X_o)$ - M = 1000	Difference (%)
0	107.4876	107.4874	0.000186068
1	106.5253	106.525	0.000281623
2	105.534	105.5337	0.000284269
3	104.5418	104.5415	0.000286967
4	103.5493	103.5489	0.000386289
5	102.5574	102.557	0.000390025
6	101.5671	101.5668	0.000295371
7	100.5798	100.5794	0.000397694
8	99.597	99.5966	0.000401619
9	98.6208	98.6205	0.000304195
10	97.6541	97.6537	0.000409609
11	96.7	96.6996	0.00041365
12	95.7628	95.7624	0.000417699
13	94.8473	94.847	0.000316298
14	93.9595	93.9592	0.000319287
15	93.1057	93.1054	0.000322214
16	92.2933	92.2931	0.0002167
17	91.5302	91.53	0.000218507
18	90.8244	90.8242	0.000220205
19	90.1844	90.1842	0.000221768
20	89.6182	89.6181	0.000111584
21	89.1337	89.1336	0.000112191
22	88.738	88.738	0
23	88.4379	88.4379	0
24	88.2398	88.2399	0.000113328
25	**88.1515***	**88.1516***	0.000113441
26	88.1837	88.1838	0.0001134
27	88.3536	88.3538	0.000226363
28	88.6879	88.688	0.000112755
29	89.221	89.2212	0.000224162
30	89.9483	89.9485	0.00022235

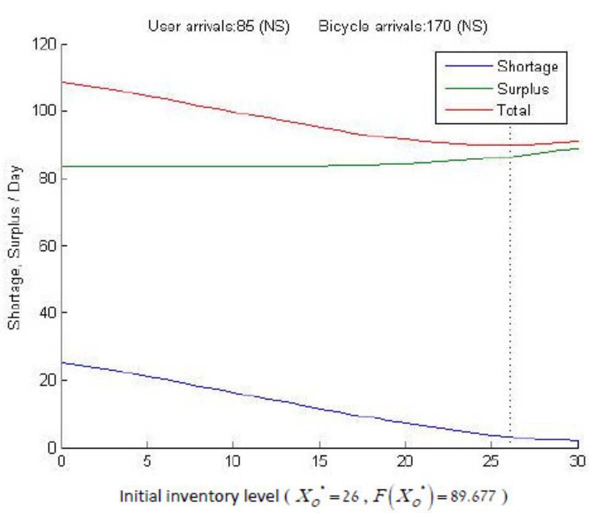

Fig 15. Peaks – non symmetric problem with 15 minutes and M=1000

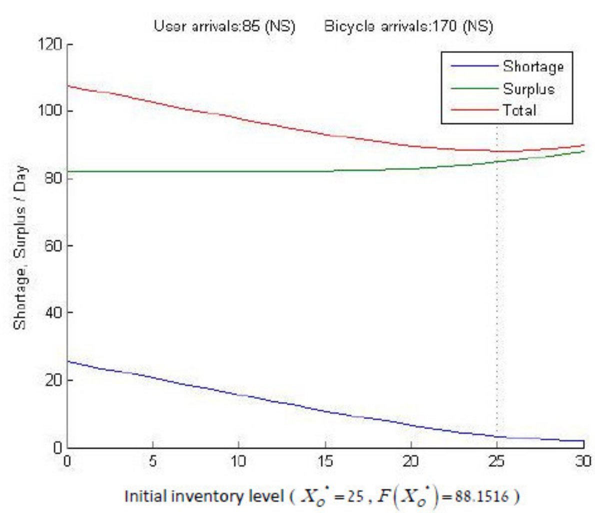

Fig 16. Peaks – non symmetric problem with 1 minute and M=1000

3.3.2.5 Validation of the approximation procedure results

Our main goal in this subsection is to evaluate the accuracy of the approximation procedure. Since no analytical method to resolve the Replenishment problem is available, we check whether the results obtained from our procedure agree with estimates obtained from a simulation study that is based on numerous replications. We note that since such a simulation takes a long time to run, it is not a viable alternative to operators of Bike-Sharing systems who have to solve the Replenishment problem on a regular basis for hundreds of stations in an ever changing environment. The validity study is based on a comparison between the results for the test problems calculated by the approximation procedure and by a simulation model. The simulation model includes one station with a capacity of 30 lockers. When a renter arrives to the station, a query regarding the number of bicycles available in the station is being made - if there are bicycles available, the current bicycles counter decreases by 1 and if there are no bicycles available, the shortage penalty is updated.

When a returner arrives at the station, a query regarding the number of lockers available in the station is being made - if there are lockers available, the current bicycles counter increases by 1 and if there are no lockers available, the surplus penalty is updated. The arrival rates of renters and returners are taken from the demand data for the tested problem (see Appendix B). The data collected during the simulation run is the expected shortage penalty, the expected surplus penalty and the expected total penalty. The simulation model was built using the ROCKWELL ARENA 12.0 simulation package. The simulation length was set to 18 hours without a warm-up period due to the fact that we are interested in the transient state and not in the steady state, as was explained in §3.1. The expected total penalty was estimated based on 1000 replications for all of the problems except for the *Peaks – non symmetric* problem where 3000 replications were used, to be further explained. The 95% confidence intervals were calculated and appeared to be narrow. The shortage penalty and surplus penalty are set to 1 (as in the procedure experiments), so the expected total penalty equals the number of renters arriving at an empty station and the number of returners arriving at a full capacity station. The simulation model was applied to each possible level of initial inventory in order to estimate the whole total penalty function. Common streams of random numbers (inter arrival times) were used for the entire model configuration, i.e., all possible initial inventories.

Another goal in this set of experiments is to compare different discretization levels. We conjecture that the greater the discretization level is, the better the approximation is. The discretization levels that were explored are 1, 5 and 15 minutes. From the results presented in Tables 8-10, Figures 17-21 and Appendix D, we conclude that the majority of the results is in the confidence level of 95% for all the problems.

The 15 minutes discretization produces valid results for the *homogenous* problems (symmetric and non – symmetric) except for the margins (0 and C), and the optimal X_0 agrees with the simulation. In the *peaks* problems (symmetric and non – symmetric) most of the results are outside of the confidence interval, but are very close to the interval margins (less then 3%). The fact that the optimal X_0 equals to the simulation results in the symmetric case and off by one locker (26 instead of 25) in the non-symmetric case supports that. In the *Random - symmetric* problem case, the situation is the same as the *Peaks – symmetric* problem and the optimal X_0 agrees with the simulation.

The 1 and 5 minutes discretization levels produce valid results in all cases (except the full station margin in the *peaks – non symmetric* problem in the 1 minute discretization).

An encouraging phenomenon was discovered when running the simulation with 3000 replications in the *Peaks – non symmetric* problem. We have noticed that the more replications the simulation is based on, the better the 1 minute discretization becomes. Thus we conclude that the approximation procedure is highly precise, so even the simulation model is less accurate in some cases.

As for the comparison between the various discretization levels, we see that in each discretization level, the results behave the same in all of the problems except for slight changes. Tables 11-13 show the comparison between all the discretization levels in two aspects: The largest differences in the expected total penalty and the gap between the optimal initial inventory level. In the two *homogenous* problems (the symmetric and non-symmetric) the difference between the expected total penalty of the 1 minute and 5 minutes, or the 5 minutes and 15 minutes discretization for any given X_0 is less than 4%, and less than 1% for X_0, that are not in the margins (0 and C). If we compare the 1 minute and 15 minutes discretization for any given X_0, the difference is less then 6%, and less then 2% for X_O that are not in margins (0 and C). More importantly, all of the discretization levels agree on the same optimal X_O.

In the *Peaks – symmetric* problem, the difference between the expected total penalty of the 1 minute and 5 minutes or the 5 minutes and 15 minutes discretization for any given X_O is less then 5%, and for the 1 minute and 15 minutes, the difference is less than 7%. Here also all of the discretization levels agree on the same optimal X_O.

The *Peaks –non symmetric* problem produced slightly smaller differences than the symmetric peaks problem, with less than 2% differences in the expected total penalty for the 1 minute and 5 minutes, or the 5 minutes and 15 minutes discretization, and less than 3% for the 1 minute and 15 minutes. Even though, here the 15 minutes discretization level does not agree with the 1 and 5 minutes discretization level on the optimal X_O (26 for the 15 minutes discretization and 25 for the 1 and 5 minutes discretizations). This difference is not meaningful of course, due to the small difference in the optimal X_O value (3.22%).

The *Random – symmetric* problem produced differences that are smaller than 8% for the 1 minute and 5 minutes or the 5 minutes and 15 minutes discretization and less than 11% for the 1 minute and 15 minutes.. More importantly, all of the discretization levels agree on the same optimal X_O.

Figures 17-21 show the expected total penalty as a function of X_O. We see that the lines are in the exact same trend, a fact that strengthens the model validity. Empirically, the higher the discretization level is, the smaller the expected total penalty is, clearly seen from the graphs. However, the location of the simulation graph between the model's graphs is not permanent. Notice that the differences are minor and thus this insight is not very significant. Moreover, note that we calculate the penalties with respect to the state at the end of each time interval. If we would calculate those penalties with respect to the state in the middle of each time interval, we expect the results of different discretization levels to be closer. We elaborate on this issue in the Conclusions and future research Chapter.

In conclusion, the approximation procedure is proven valid in a 95% confidence level and the differences between the various discretization levels are minor considering the expected total penalty and negligible considering the optimal X_O.

Hence, we conclude that the algorithm is appropriate for the use of decision makers in a dynamic environment such as a Bike-Sharing system.

Another interesting phenomenon observed from the results is that the graphs produced by the approximation procedure are not always convex at the margin. This phenomenon can be observed visually in the 15 minutes discretization level graphs.

A possible explanation can be that due to the low probability of being at the margin at each time interval, the margins of the graph are the outcome of summing very small numbers and hence larger numerical errors occur.

A possible work around for this issue, if one wishes to use the convexity property in optimizing the initial inventory in the whole system (subject to some constraints), is to approximate the non-convex fragments by linear functions.

Table 8. Homogenous problems comparison to simulation

Problem	Discretization level [minutes]	Number of results in the 95% confidence interval	X_O^* Gap
Homogenous - symmetric	15	30/31	0
Homogenous - symmetric	5	31/31	0
Homogenous - symmetric	1	31/31	0
Homogenous – non symmetric	15	30/31	0
Homogenous – non symmetric	5	31/31	0
Homogenous – non symmetric	1	31/31	0

Table 9. Peaks problems comparison to simulation

Problem	Discretization level [minutes]	Number of results in the 95% confidence interval	X_O^* Gap
Peaks - symmetric	15	3/31	0
Peaks - symmetric	5	31/31	0
Peaks - symmetric	1	31/31	0
Peaks – non symmetric	15	1/31	1
Peaks – non symmetric	5	31/31	0
Peaks – non symmetric	1	30/31	0

Table 10. Random problem comparison to simulation

Problem	Discretization level [minutes]	Number of results in the 95% confidence interval	X_O^* Gap
Random - symmetric	15	1/31	0
Random - symmetric	5	31/31	0
Random - symmetric	1	31/31	0

Table 11. Homogenous problems discretization levels comparison

Problem	Discretization level comparison [minutes]	Largest expected total penalty difference	X_O^* Gap
Homogenous - symmetric	1 and 5	1.36%	0
Homogenous - symmetric	5 and 15	2.86%	0
Homogenous - symmetric	1 and 15	4.26%	0
Homogenous – non symmetric	1 and 5	1.93%	0
Homogenous – non symmetric	5 and 15	3.77%	0
Homogenous – non symmetric	1 and 15	5.78%	0

Table 12. Peaks problems discretization levels comparison

Problem	Discretization level comparison [minutes]	Largest expected total penalty difference	X_O^* Gap
Peaks - symmetric	1 and 5	1.73%	0
Peaks - symmetric	5 and 15	4.57%	0
Peaks - symmetric	1 and 15	6.22%	0
Peaks – non symmetric	1 and 5	0.62%	1
Peaks – non symmetric	5 and 15	1.56%	0
Peaks – non symmetric	1 and 15	2.16%	0

Table 13. Random problem discretization levels comparison

Problem	Discretizatio n level comparison [minutes]	Largest expected total penalty difference	X_O^* Gap
Random - symmetric	1 and 5	3.19%	0
Random - symmetric	5 and 15	7.94%	0
Random - symmetric	1 and 15	10.89%	0

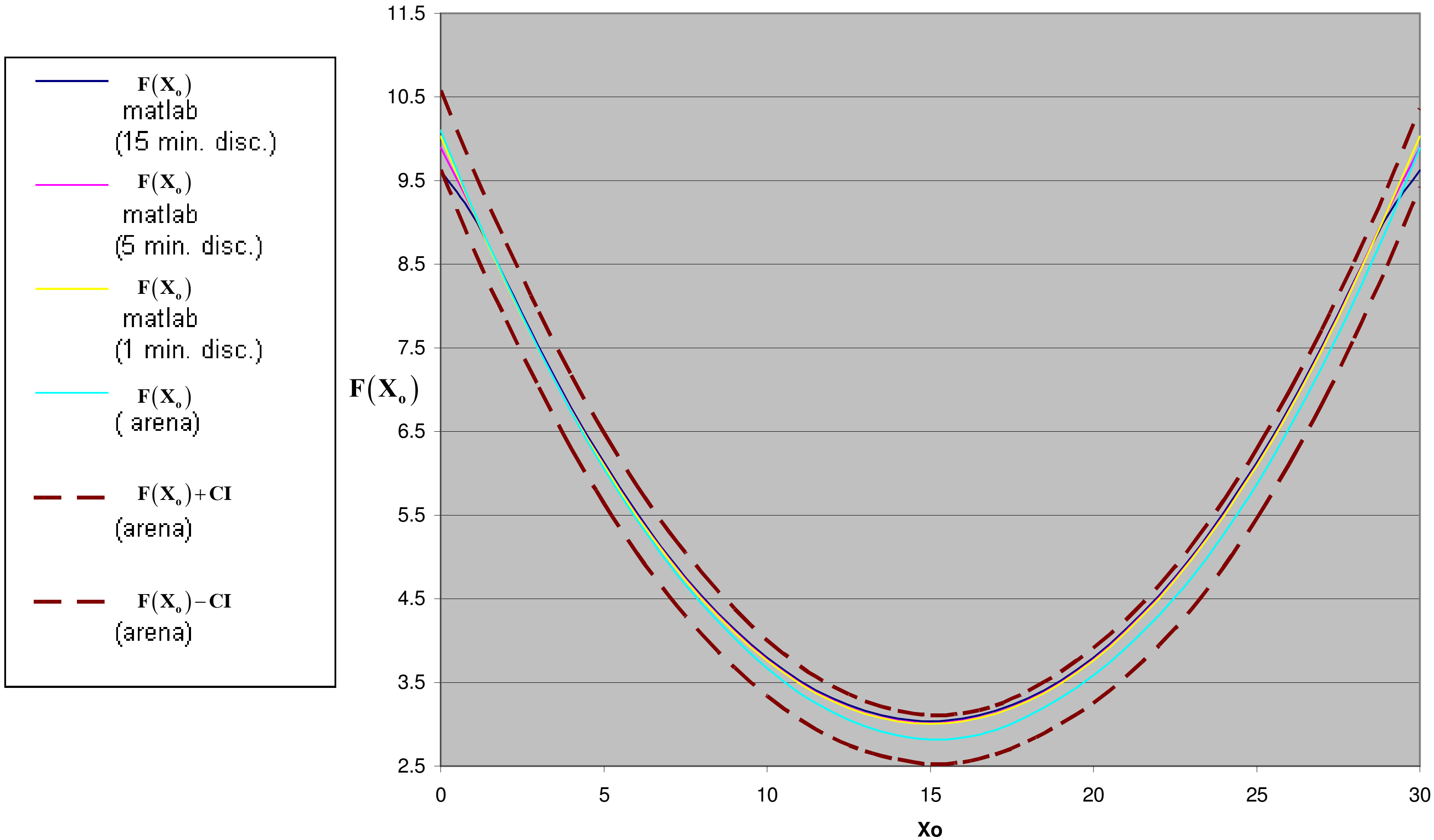

Fig 17. Homogenous –symmetric problem validation

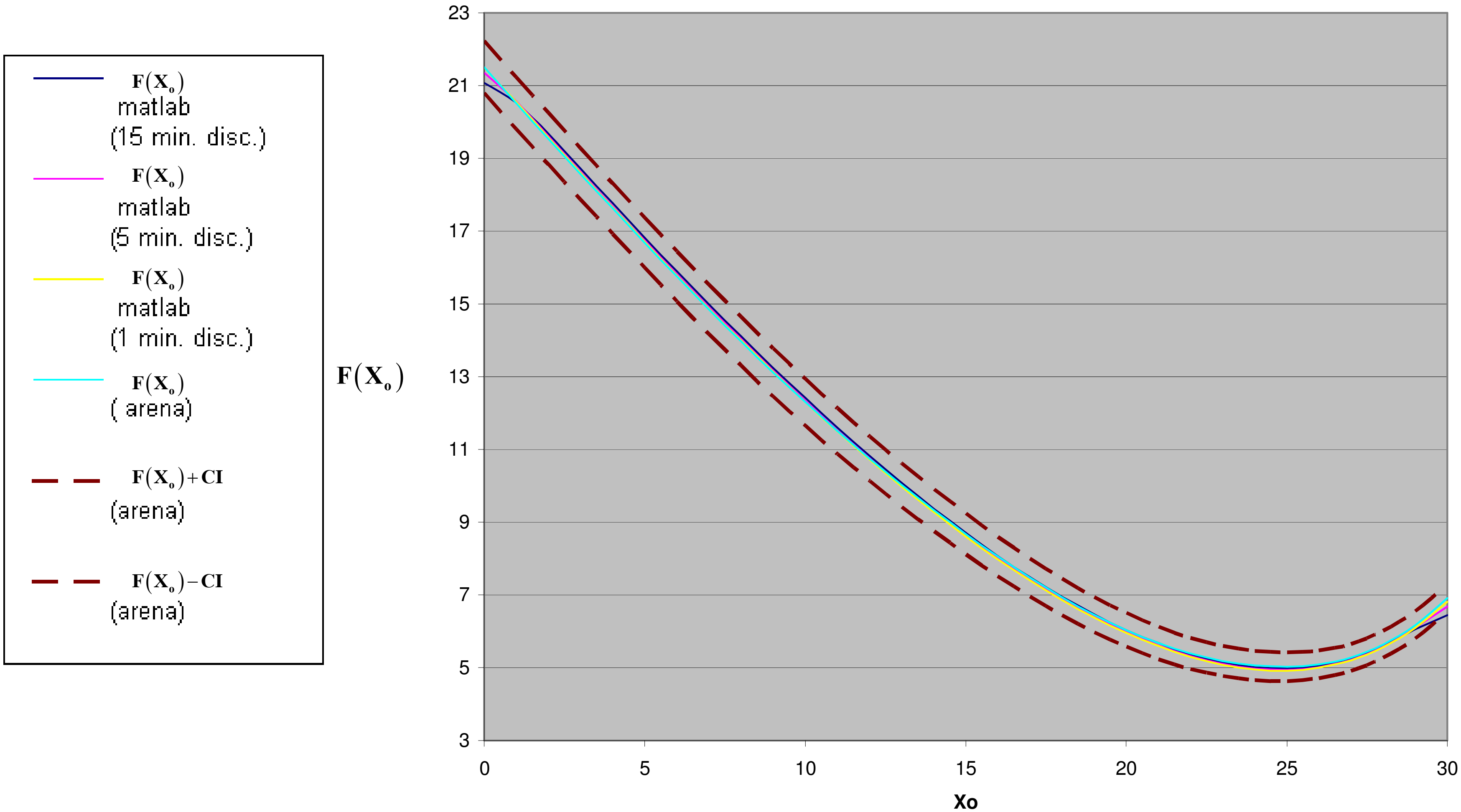

Fig 18. Homogenous – non symmetric problem validation

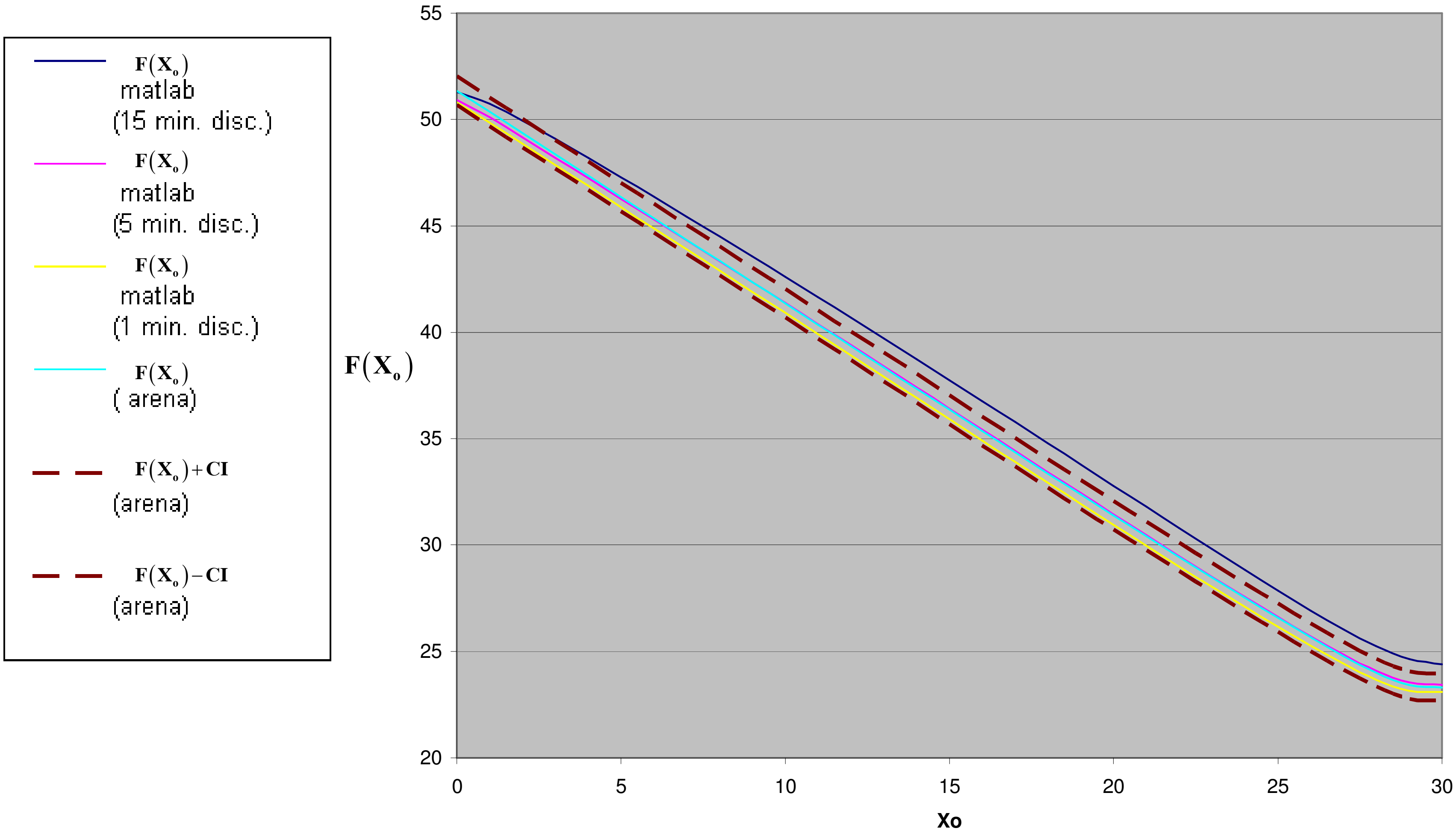

Fig 19. Peaks –symmetric problem validation

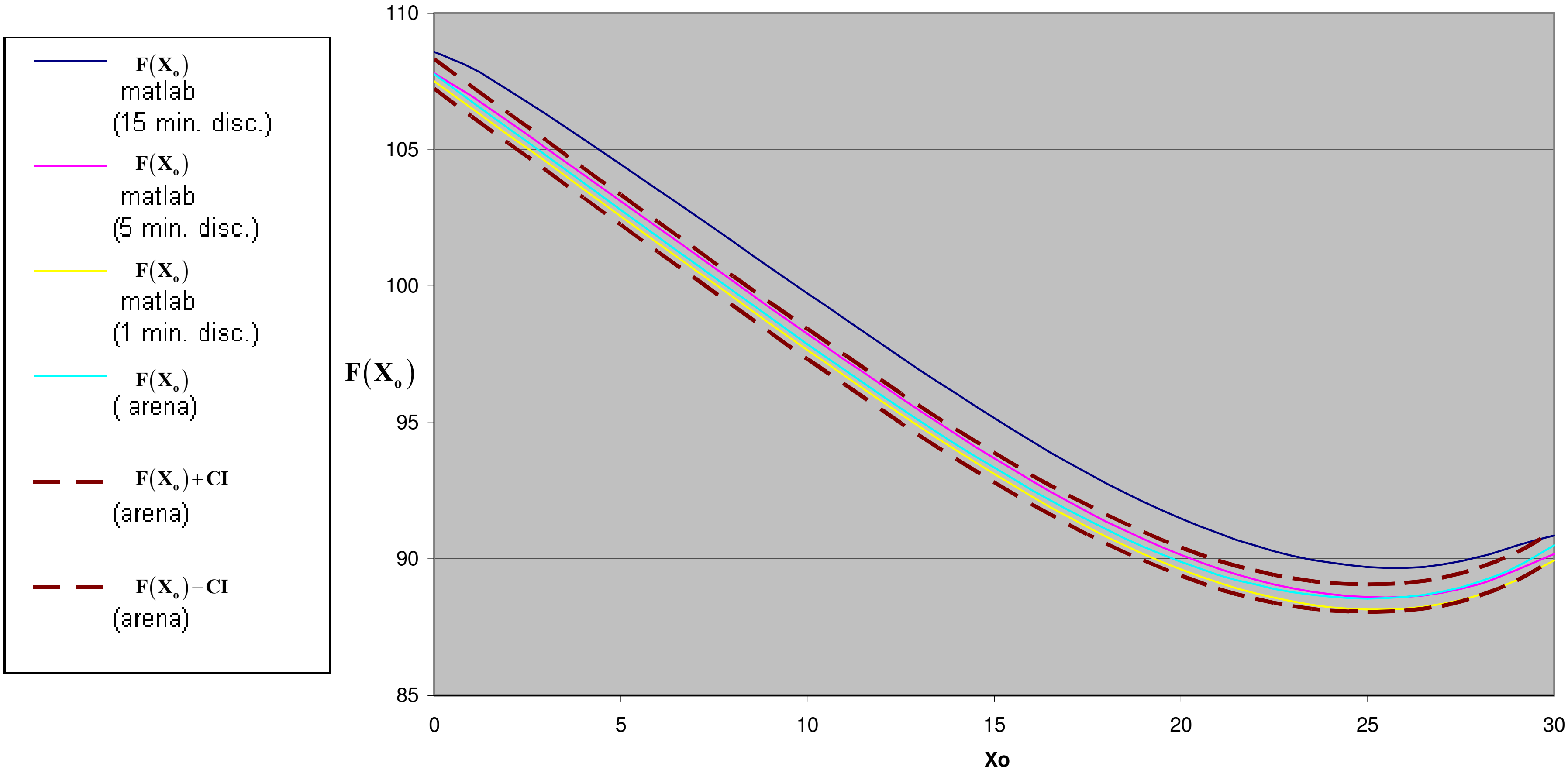

Fig 20. Peaks – non symmetric problem validation

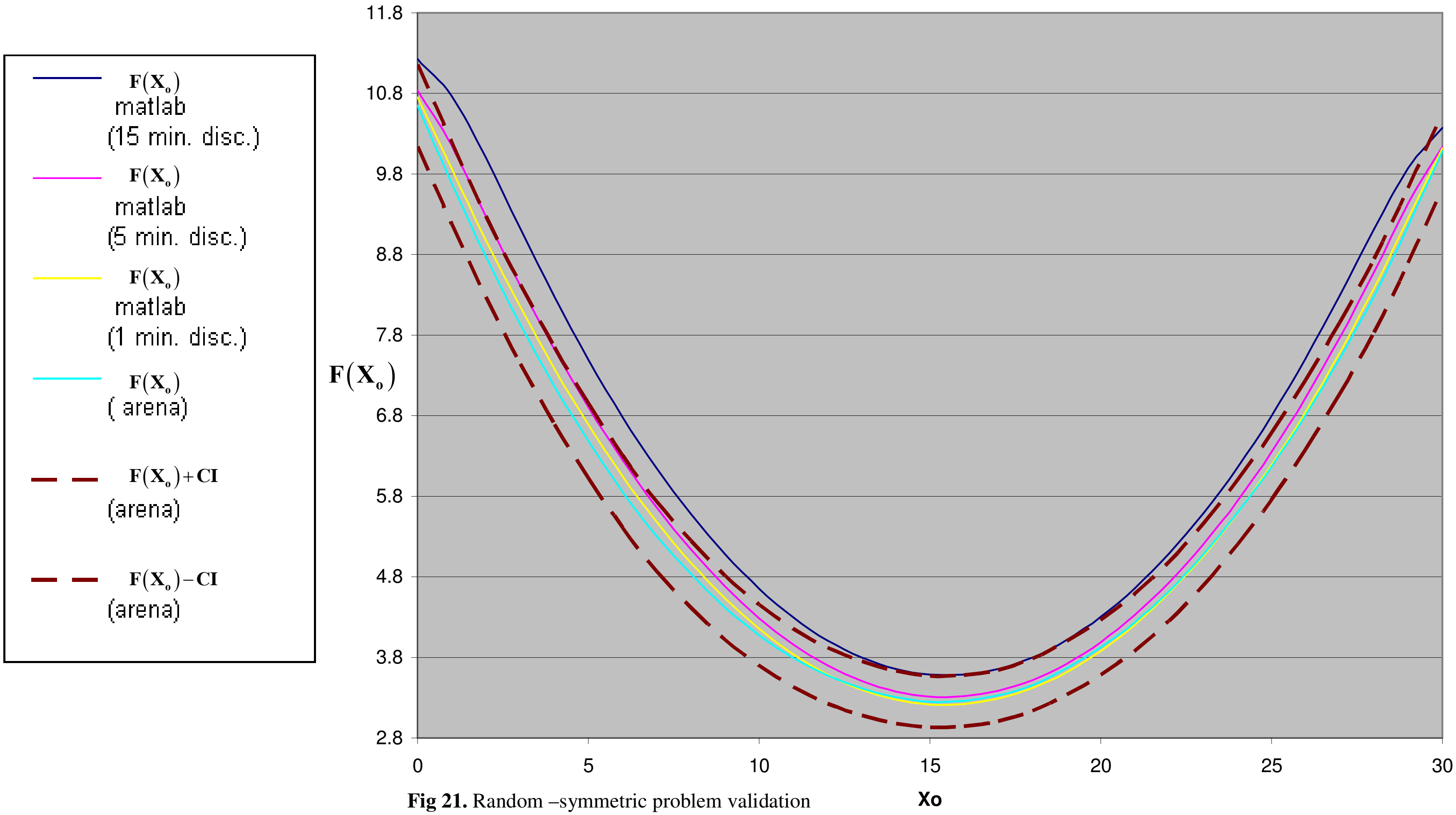

Fig 21. Random –symmetric problem validation

3.3.2.6 Computing time

Another important factor that must be examined for decision making in a dynamic environment such as Bike-Sharing system is the run time of the computation model.
A comparison between the approximation model and the simulation computing time is shown in Table 14. As shown in the table, the model produces efficient computing time, where the results were obtained in mean time of 0.0968, 0.256 and 1.27 seconds in the 15, 5 and 1 minutes discretization levels respectively, on an Intel Pentium 4, 2.4 GHz personal computer.

Comparison with the simulation model results reveals an enormous difference. The simulation model solves the problem for each X_O separately, so a complete solution (all possible X_O) requires 31,000 replications in the 1000 replications per one X_O case, and 93,000 replications in the 3000 replications per one X_O case. This produces extremely high run times, so the mean run time for a single alternative of the X_O takes 7 minutes, and the complete problem solution takes a mean time of 3.5 hours in the 1000 replications case. In the 3000 replications case, the mean run time for a single alternative of the X_O takes 20 minutes and the complete problem solution takes a mean time of 10 hours. The standard deviations of the run times are negligible in both cases (the approximation procedure and the simulation model) and therefore do not appear in the table.

We conclude from these results that the simulation model can only be in use for validation purpose and it is not highly recommended for decision makers. The short run time of the approximation procedure makes it much more attractive for decision makers who use the model in a dynamic environment.

Note that the test problems we considered focus on a station with a capacity of 30 lockers, which is the typical size in Bike-Sharing systems. Nevertheless, stations with larger capacities exist in these systems and it is worth mentioning that our procedure yields quicker calculations in these cases also. Let us define the number of feasible initial inventory levels as *m*, the average renters rate as $\bar{\mu}$ and the average returners rate as $\bar{\lambda}$. The complexity of the simulation model is affected by the capacity of the station, the planning horizon length and by the number of events. The mean complexity of each repetition of the simulation is $\left(O(T \cdot m \cdot [\bar{\mu} + \bar{\lambda}])\right)$. Note that it is very likely that in larger stations there will be more events. Conversely, our approximation procedure is affected by the capacity of the station (due to matrix multiplications, see (29)) the planning horizon length and the discretization level (and not by the number of events). Hence, the complexity is expected to be $O(T/d \cdot m^3)$.

To verify that our procedure is indeed quicker than the simulation model, even when a larger station is considered, we performed the procedure on all the 5 test problems (15 minutes discretization level), but in this case we considered a station with 70 lockers, which is the largest capacity existing in Bike-Sharing systems. The mean computing time was 0.4782 seconds. Hence, our procedure is recommended also for larger stations.

Table 14. Mean computing time of the approximation procedure and the simulation model

Method	Computing time
Approximation procedure – 15 minutes discretization	0.0968 Sec
Approximation procedure –5 minutes discretization	0.256 Sec
Approximation procedure – 1 minutes discretization	1.27 Sec
Simulation	3.5 Hours

3.3.2.7 *Operational insights*

After the model was validated, we drew some operational insights concerning the nature of the optimal initial inventory level of the test problems, enabling decision makers to develop an intuition for making good decisions. As we believe that part of our test problems could represent a realistic behavior of a Bike-Sharing station, some of these insights can be helpful in practice.

Tables 15-19 show the results of the expected total penalty for every X_O and Figures 22-36 contain the proper graphs which include expected shortage penalty, expected surplus penalty and expected total penalty for every X_O. The experiments show that the results are very intuitive.

In the *Homogenous – symmetric* problem, it is seen that the optimal X_O is right in the center of the station. The intuition is the equality of the renters and returners demand rates, which makes the middle of the station the "safest" place that guarantees the lowest expected shortage and surplus penalties. It is also seen that the expected shortage and surplus penalties are perfectly antisymmetric to each other, e.g. the expected shortage penalty for a station with $X_O = x,\ x \in \{0,\ldots,30\}$ equals the expected surplus penalty for a station with $X_O = 30 - x$ and vice versa.

In the *Homogenous – non symmetric* problem, the optimal X_O is 25 out of 30 lockers. This is also not a surprising result. The ratio between the full capacity (30) and the optimal X_O (25) equals exactly to the ratio between the homogenous renters and returners demand rates (which is 1.2). The renters demand rate intensity is bigger than the returners demand rate intensity and thus the optimal X_O is closer to a full station.

The optimal X_O for the *Peaks – symmetric* problem is full capacity (30). This makes sense due to the intensity of morning peak demand of renters. The afternoon peak of returners demand is practically meaningless (shown in Figures 28-30) so that the expected surplus penalty approximately does not change at all as a function of X_O. It is intuitive to set X_O to full capacity, assuring us the lowest expected shortage penalty. We know that the afternoon peak will cause multitude surplus events, but these are difficult to prevent because the station will be nearly empty in any case after the morning rush.

The *Peaks –non symmetric* problem is a good example of testing the effect of the afternoon peak. We purposely set the return rate to be twice as big as the demand rate so we can be convinced that the afternoon peak has a negligible effect on the decision of the optimal X_O. Indeed, the optimal X_O is 26 in the 1 and 5 minutes discretization solution and 25 in the 15 minutes discretization solution, i.e. even when the difference between the afternoon peak and the morning peak is significantly large, the optimal X_O is very close to full capacity.

The case of the *Random – symmetric* problem is aimed to examine how the approximation procedure handles unstable renters and returners demand processes, where the amount of customer arrivals and bicycles return is equal on the planning horizon. We see that as the *Homogenous – symmetric* problem, when there are no significant peaks in the renters or returners demand and the problem is symmetric, the optimal X_O is in the area of the center of the station. In this case the optimal X_O is exactly in the center (15).

Table 15. Homogenous – symmetric problem discretization

X_O	$F(X_O)$ - 15 min. disc.	$F(X_O)$ - 5 min. disc.	$F(X_O)$ - 1 min. disc.
0	9.6193	9.894	10.0289
1	9.076	9.1345	9.1358
2	8.3017	8.2836	8.2731
3	7.516	7.4884	7.4778
4	6.7873	6.7605	6.75
5	6.1258	6.0996	6.0891
6	5.5307	5.5048	5.4945
7	5.0011	4.9756	4.9654
8	4.5363	4.5111	4.501
9	4.1355	4.1105	4.1006
10	3.7977	3.7731	3.7632
11	3.5225	3.4981	3.4884
12	3.3092	3.2849	3.2753
13	3.1571	3.133	3.1234
14	3.0661	3.0421	3.0325
15	**3.0356***	**3.0118***	**3.0022***
16	3.0661	3.0421	3.0325
17	3.1571	3.133	3.1234
18	3.3092	3.2849	3.2753
19	3.5225	3.4981	3.4884
20	3.7977	3.7731	3.7632
21	4.1355	4.1105	4.1006
22	4.5363	4.5111	4.501
23	5.0011	4.9756	4.9654
24	5.5307	5.5048	5.4945
25	6.1258	6.0996	6.0891
26	6.7873	6.7605	6.75
27	7.516	7.4884	7.4778
28	8.3017	8.2836	8.2731
29	9.076	9.1345	9.1358
30	9.6193	9.894	10.0289

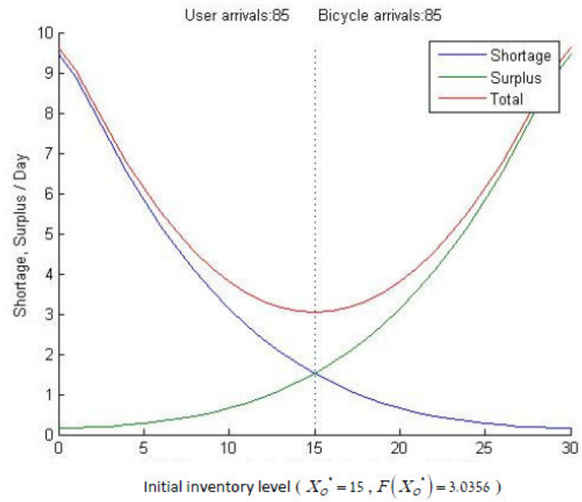

Fig 22. Homogenous – symmetric problem 15 minutes discretization

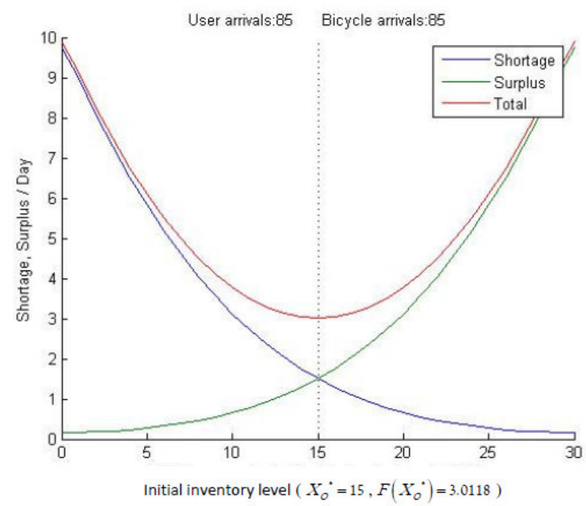

Fig 23. Homogenous – symmetric problem 5 minutes discretization

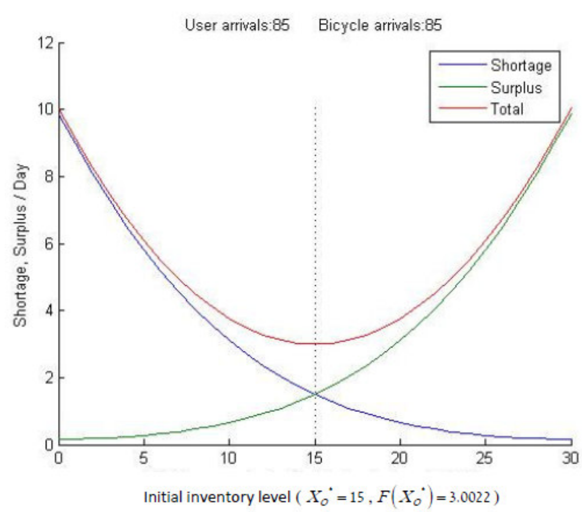

Fig 24. Homogenous – symmetric problem 1 minute discretization

Table 16. Homogenous – non symmetric problem discretization

X_o	$F(X_o)$ - 15 min. disc.	$F(X_o)$ - 5 min. disc.	$F(X_o)$ - 1 min. disc.
0	21.0749	21.3545	21.4982
1	20.5308	20.5689	20.5534
2	19.6721	19.6057	19.5728
3	18.719	18.6364	18.603
4	17.7617	17.6799	17.6469
5	16.8196	16.7389	16.706
6	15.8951	15.8151	15.7824
7	14.9899	14.9106	14.8782
8	14.1059	14.0274	13.9954
9	13.2453	13.1678	13.1362
10	12.4102	12.3339	12.3027
11	11.6032	11.5282	11.4975
12	10.8265	10.7528	10.7227
13	10.0824	10.0103	9.9809
14	9.3739	9.3035	9.2747
15	8.7034	8.6349	8.6069
16	8.074	8.0075	7.9803
17	7.489	7.4245	7.3981
18	6.9517	6.8895	6.864
19	6.4665	6.4065	6.382
20	6.038	5.9802	5.9566
21	5.6716	5.6162	5.5936
22	5.3739	5.3209	5.2993
23	5.1529	5.1022	5.0815
24	5.0181	4.9695	4.9497
25	**4.9812***	**4.9346***	**4.9157***
26	5.0568	5.0119	4.9938
27	5.2624	5.2188	5.2013
28	5.6096	5.5769	5.5601
29	6.0597	6.1007	6.0965
30	6.4341	6.6769	6.8057

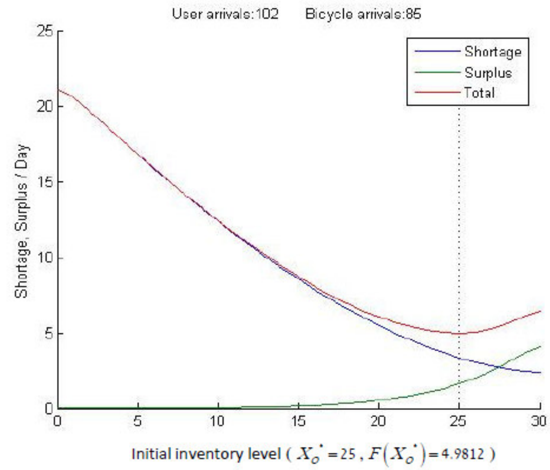

Fig 25. Homogenous – non symmetric problem 15 minutes discretization

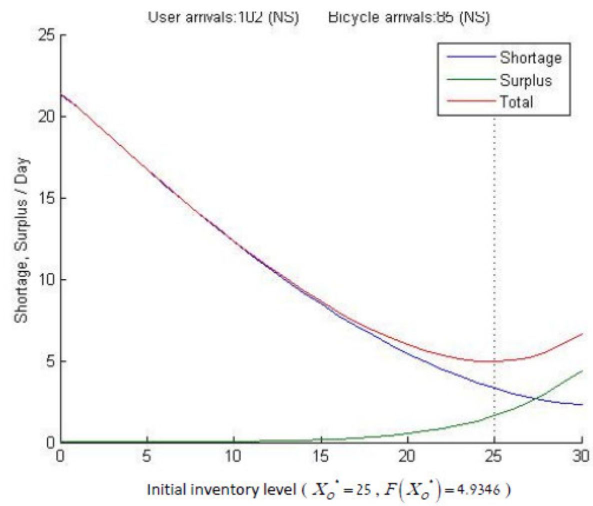

Fig 26. Homogenous – non symmetric problem 5 minutes discretization

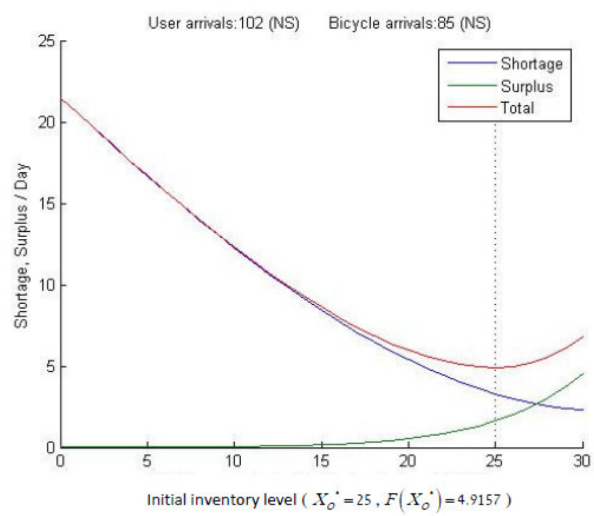

Fig 27. Homogenous – non symmetric problem 1 minute discretization

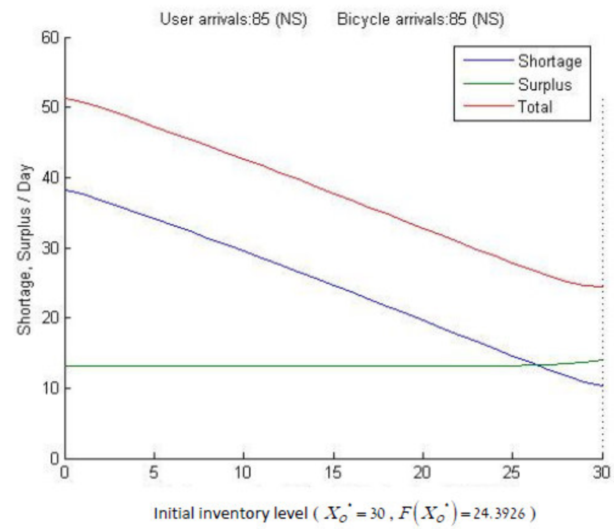

Fig 28. Peaks – symmetric problem 15 minutes discretization

Table 17. Peaks – symmetric problem discretization results

X_o	$F(X_o)$ - 15 min. disc.	$F(X_o)$ - 5 min. disc.	$F(X_o)$ - 1 min. disc.
0	51.296	50.9288	50.8154
1	50.7443	50.1092	49.8542
2	49.9454	49.1648	48.8641
3	49.0827	48.2072	47.8724
4	48.1959	47.2424	46.8791
5	47.2921	46.2719	45.8849
6	46.3749	45.2973	44.8897
7	45.4468	44.3192	43.894
8	44.5094	43.3382	42.8976
9	43.5639	42.3546	41.9008
10	42.6111	41.3687	40.9035
11	41.6514	40.3807	39.9059
12	40.6855	39.3908	38.908
13	39.7137	38.3992	37.91
14	38.7363	37.4061	36.912
15	37.7538	36.4117	35.9142
16	36.7664	35.4165	34.917
17	35.7749	34.4209	33.9208
18	34.7797	33.4256	32.9264
19	33.782	32.4317	31.9348
20	32.783	31.4405	30.9473
21	31.7847	30.4537	29.9658
22	30.7892	29.4739	28.9925
23	29.7997	28.5039	28.0304
24	28.8204	27.5477	27.0835
25	27.8569	26.6109	26.1571
26	26.919	25.7025	25.2602
27	26.0271	24.8423	24.4122
28	25.2329	24.0803	23.6626
29	24.6472	23.5417	23.138
30	**24.3926***	**23.4326***	**23.1051***

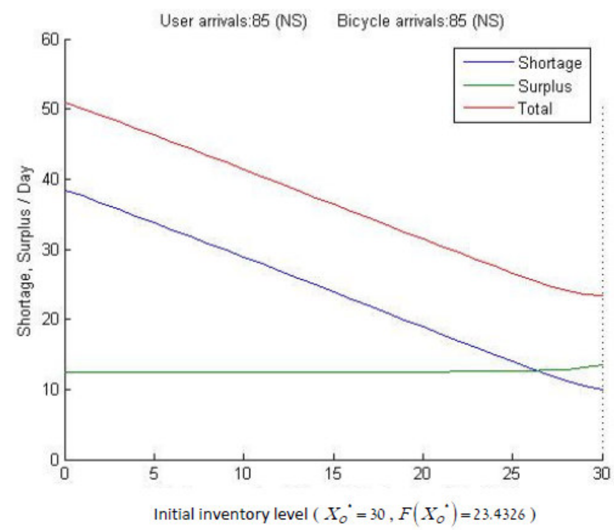

Fig 29. Peaks – symmetric problem 5 minutes discretization

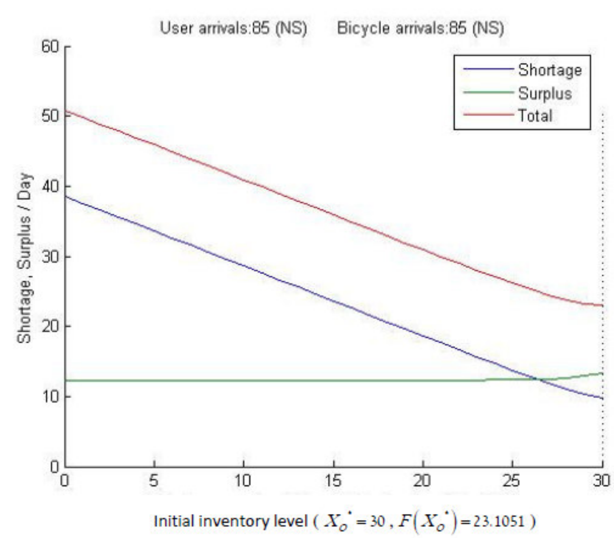

Fig 30. Peaks – symmetric problem 1 minute discretization

Table 18. Peaks – non symmetric problem discretization results

X_o	$F(X_o)$ - 15 min. disc.	$F(X_o)$ - 5 min. disc.	$F(X_o)$ - 1 min. disc.
0	108.5834	107.796	107.48758
1	107.9959	106.9676	106.52527
2	107.1746	106.0152	105.53402
3	106.29	105.0499	104.54178
4	105.3818	104.0787	103.54926
5	104.4583	103.1039	102.55735
6	103.5237	102.1275	101.5671
7	102.5812	101.1513	100.57977
8	101.6337	100.1771	99.596979
9	100.6842	99.2073	98.620843
10	99.7357	98.2447	97.654052
11	98.7922	97.2927	96.699979
12	97.8577	96.3557	95.762755
13	96.9375	95.4385	94.847323
14	96.0376	94.5469	93.959452
15	95.1647	93.6877	93.1057
16	94.3261	92.868	92.293332
17	93.5299	92.0958	91.530174
18	92.7847	91.3793	90.824416
19	92.0992	90.727	90.18437
20	91.4821	90.1471	89.618202
21	90.9416	89.6476	89.133684
22	90.4859	89.2359	88.738021
23	90.1224	88.9189	88.437884
24	89.8591	88.7034	88.239847
25	89.7054	**88.5975***	**88.15152***
26	**89.6753***	88.6125	88.183719
27	89.7903	88.7667	88.353648
28	90.0732	89.0877	88.687908
29	90.4931	89.6003	89.22101
30	90.8529	90.1908	89.948251

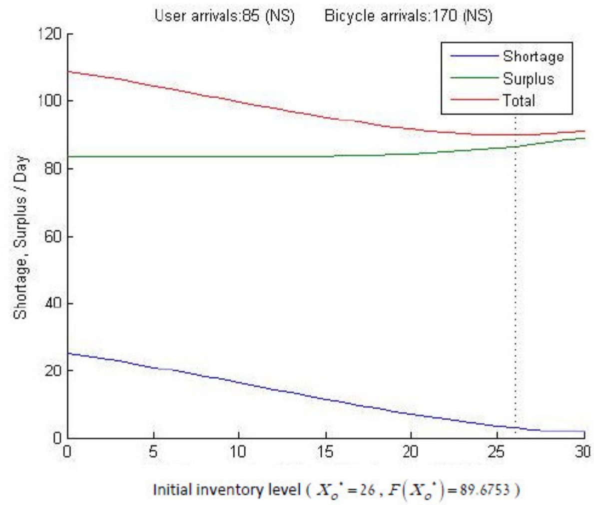

Fig 31. Peaks – non symmetric problem 15 minutes discretization

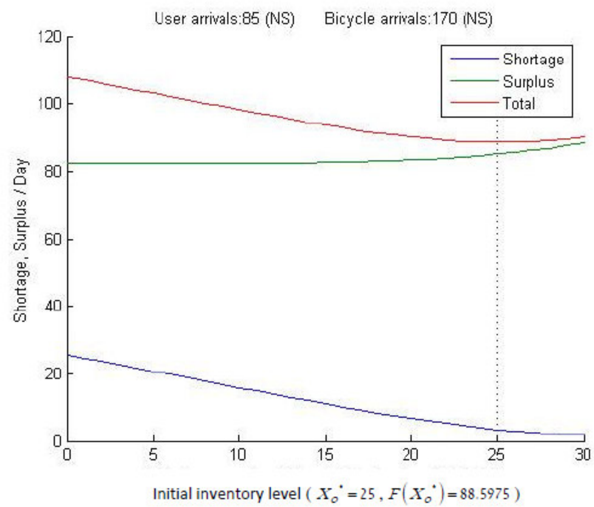

Fig 32. Peaks – non symmetric problem 5 minutes discretization

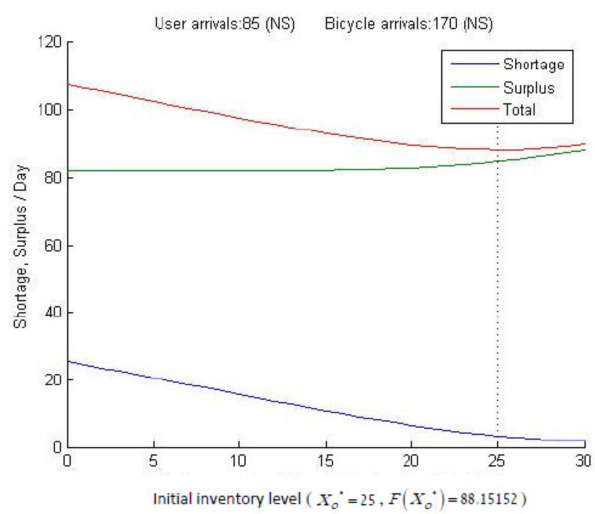

Fig 33. Peaks – non symmetric problem 1 minute discretization

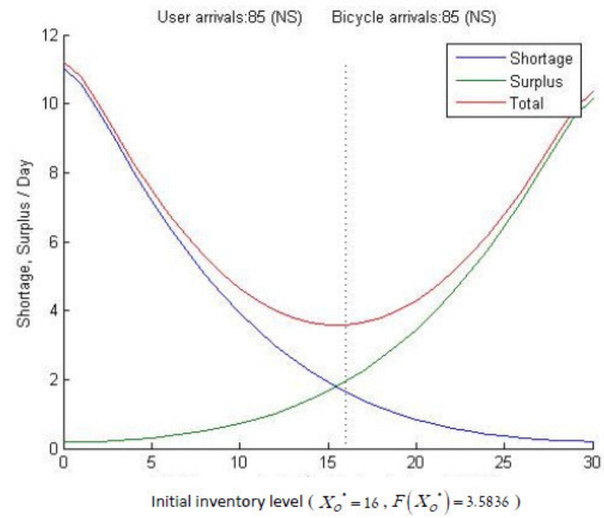

Fig 34. Random-symmetric problem 15 minutes discretization

Table 19. Random - symmetric problem discretization results

X_O	$F(X_o)$ - 15 min. disc.	$F(X_o)$ - 5 min. disc.	$F(X_o)$ - 1 min. disc.
0	11.2246	10.8396	10.7496
1	10.7701	10.1563	9.8695
2	10.0041	9.2759	8.9804
3	9.1301	8.4196	8.152
4	8.2783	7.629	7.3877
5	7.495	6.9066	6.6875
6	6.7858	6.2508	6.0516
7	6.148	5.6608	5.48
8	5.5799	5.1366	4.9726
9	5.0817	4.6782	4.5296
10	4.654	4.2859	4.1507
11	4.2974	3.9594	3.8359
12	4.012	3.6989	3.5849
13	3.7978	3.5042	3.3976
14	3.655	3.3752	3.2739
15	**3.5836***	**3.312***	**3.2138***
16	**3.5836***	3.3146	3.2175
17	3.6551	3.3833	3.2852
18	3.7982	3.5183	3.4173
19	4.0131	3.72	3.6141
20	4.2995	3.9886	3.8763
21	4.6573	4.3247	4.2044
22	5.0862	4.7283	4.5989
23	5.5856	5.1998	5.0603
24	6.1555	5.7392	5.589
25	6.7967	6.3468	6.1854
26	7.5104	7.0233	6.8497
27	8.2932	7.7696	7.582
28	9.1149	8.584	8.382
29	9.8746	9.4375	9.2472
30	10.3701	10.1381	10.1207

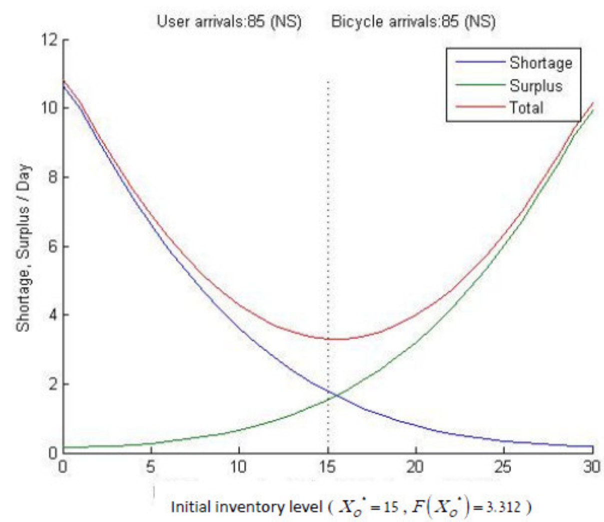

Fig 35. Random-symmetric problem 5 minutes discretization

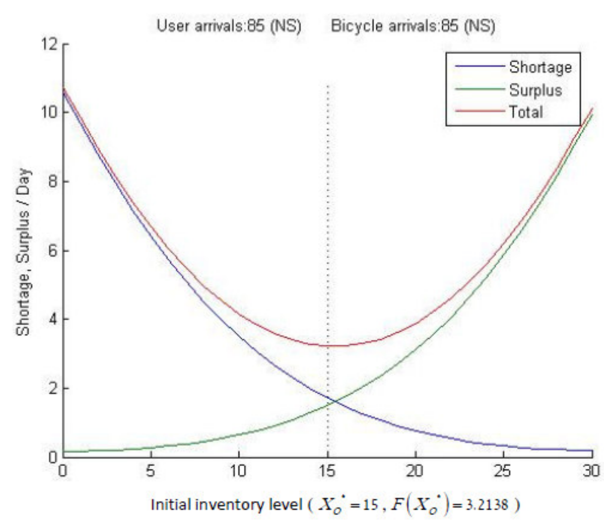

Fig 36. Random-symmetric problem 1 minute discretization

3.3.2.8 Comparison between the transient state and steady state analyses

A standard approach in the modeling of service systems is to assume that the system is operated in its steady state. If the stochastic processes that govern the system are changing over time, the analysis is conducted separately for each short period for which in the processes are more or less homogenous. This approach is very fruitful for systems with high demand volume, such as call centers. This is due to the fact that when the rate of events (e.g., arrival of customers) is high, the system approaches its steady state behavior quickly, consequently the transient behavior can be neglected. However, steady state analysis is not useful for inventory systems such as a bicycle rental station. These systems are governed by manipulating their state (e.g., changing the inventory level) every certain time and it is not likely, or desirable, to have the system approach its steady state before the next control operation.

In order to illustrate the problem of using steady state analysis for the Replenishment problem discussed in §3.1, we compare our approximation procedure results for three test problems to the results received while using steady state considerations.

From the discussion in §3.1, assuming a Poisson stochastic arrival process of renters and returners with demand rates of $\mu(t)$ and $\lambda(t)$ respectively, we can think of the Replenishment problem as a M/M/1/C birth and death process. The appropriate formula for $\tilde{P}_n(t)$ (the probability of being at state *n* in the steady state of time interval *t*) is (Ross 1997):

$$\tilde{P}_n(t) = \begin{cases} \dfrac{\left(\lambda(t)/\mu(t)\right)^n \left(1-\lambda(t)/\mu(t)\right)}{1-\left(\lambda(t)/\mu(t)\right)^{C+1}} & ,\lambda(t) \neq \mu(t) \\[2ex] \dfrac{1}{C+1} & ,\lambda(t) = \mu(t) \end{cases} \qquad (32)$$

Let us denote the expected total penalty function using steady state considerations by $F\left(p^A, h^A, \mu(t), \lambda(t), C, T\right)$. Note that due to the fact that the system is in steady state, the function is not affected by the initial state (X_O).

Now,

$$F\left(p^A, h^A, \mu(t), \lambda(t), C, T\right) = p^A \cdot G_-\left(p^A, \mu(t), C, T\right) + h^A \cdot G_+\left(h^A, \lambda(t), C, T\right) \quad (33)$$

where $G_-\left(p^A, \mu(t), C, T\right)$ and $G_+\left(h^A, \lambda(t), C, T\right)$ are the expected number of shortage and surplus events respectively, during the interval $\left[0,T\right]$.

Now,

$$G_-\left(p^A, \mu(t), C, T\right) = \sum_{t=0}^{T} \mu(t) \cdot \tilde{P}_0(t) \quad (34)$$

$$G_+\left(h^A, \lambda(t), C, T\right) = \sum_{t=0}^{T} \lambda(t) \cdot \tilde{P}_C(t) \quad (35)$$

and hence

$$F\left(p^A, h^A, \mu(t), \lambda(t), C, T\right) = \sum_{t=0}^{T} \left[p^A \cdot \mu(t) \cdot \tilde{P}_0(t) + h^A \cdot \lambda(t) \cdot \tilde{P}_C(t) \right] \quad (36)$$

Note that when the arrival processes are homogenous ($\mu(t) = \mu,\ \lambda(t) = \lambda$), then $\tilde{P}_n(t) = \tilde{P}_n \quad , \forall t$ is constant and hence

$$F\left(p^A, h^A, \mu, \lambda, C, T\right) = T \cdot \left[p^A \cdot \mu \cdot \tilde{P}_0 \quad , h^A \cdot \lambda \cdot \tilde{P}_C \right] \quad (37)$$

First, we consider the two *homogenous* problems. In the *homogenous – symmetric* problem we have 85 renters arrivals and 85 returners arrivals. In the *homogenous - non symmetric* problem we have 102 renters arrivals and 85 returners arrivals. The planning horizon is 18 hours. In the two first rows of Table 20, a comparison of the expected total penalty is made between the steady state analysis and our approximation procedure's results for the optimal initial inventory level with 1 minute discretization level (as presented in §3.3.2.7), considering a 18 hours planning horizon. It is apparent from the table that the steady state analysis yields different results, compared to the results of the approximation procedure that have been validated by the simulation study in §3.3.2.5. From these comparisons, we show that the transient state analysis is the appropriate way for solving the Replenishment problem by illustrating that indeed the initial inventory level has great influence on the expected total penalty.

Second, we consider the *'peak – symmetric'* problem. Assuming an extremely long planning horizon of 500 days (a total of 42,500 renters arrivals and 42,500 returners arrivals), we compare the result of our approximation procedure (1 minute discretization level) with the result using steady state analysis. The complete results of the approximation procedure (for all initial inventory levels) and the steady state calculation (for every time interval) are presented in Appendix E. From the comparison presented in the third row of Table 20, we can see that the expected total penalty received using the approximation procedure is significantly different compared to the expected total penalty received using steady state analysis. In order to validate the result calculated by the approximation procedure, we conducted a simulation experiment (using the same model presented in §3.3.2.5), but this time with a planning horizon of 500 days and assuming no replenishment conducted during this period. A 95% confidence interval for the expected total penalty during the period was calculated based on 50 replications. The interval is $[12{,}561 \ , \ 12{,}666.6]$ and thus our approximation procedure's result is valid. This last experiment demonstrates the fact that even without intervention in the state of the system by replenishment, the transient analysis proposed in this study is appropriate while the steady state analysis is not. A possible explanation for the significant difference of the results achieved by the two methods is that the steady state analysis predicts high shortage (of bicycles or lockers) during peak hours, while in reality the renters and returners peaks may balance each other.

In conclusion, we illustrated that steady state considerations are not the appropriate method for the problem. This may be an important lesson for the analyzer of service systems in any environment where the demand rate is low relative to the length of the period during which the rate can be assumed to be constant. Indeed, steady state analysis is useful for systems with relatively high rate events, such as large scale call centers.

It is also important to mention that while one could suggest that steady state considerations can be used for a capacity decision (due to the fact that this analysis does not consider the initial state of the process), it is apparent from the aforementioned results that this is not the case.

Table 20. Comparison of steady state analysis to the approximation procedure

Problem	Planning horizon (days)	Expected total penalty - steady state analysis	Expected total penalty – Approximation procedure optimal result (1 min. disc.)
Homogenous - symmetric	18	5.484	3.0022 $\left(X_O^* = 15\right)$
Homogenous – non symmetric	18	17.354	4.9159 $\left(X_O^* = 25\right)$
Peaks - symmetric	500	35,018.54	12,645.1 $\left(X_O^* = 30\right)$

3.3.2.9 Sensitivity analysis

After testing the model validity, an important experiment is the sensitivity of the model. In such a dynamic environment as a Bike-Sharing system, it is clear that changes of the expected demand rates will occur. Inaccurate estimations of the demand rates can cause changes in the optimal initial inventory levels. Hence, an analysis for the original solution's quality is needed, i.e. to verify that the optimal X_O for the expected rates will be good enough also for rates which contain these inaccuracies. It is important to understand that this analysis is performed in terms of resistance for inaccuracies in the estimations. However, when significant changes in the rates appear, there is a need for new estimations that are proper to the current state that is available in the online control system.

3.3.2.9.1 Random inaccuracies

Inaccuracies in estimations are very common in dynamic systems. Very often these inaccuracies are random, i.e. the estimates are not biased in a specific direction (positive or negative). For the purpose of testing the quality of the solution for random inaccuracies, two problems were tested so that inaccuracies that are uniformly distributed between [-10%, 10%] were made in each one of the expected renters and returners rates per time unit. The results of these problems were compared to the problems with the expected rates. The problems that were tested are the *homogenous – non symmetric* problem and the *Peaks – symmetric* problem with 1 minute discretization level.

Figures 37 and 39 show the *Homogenous – non symmetric* problem and the *Peaks – symmetric* problem demand rates respectively after the inaccuracies were made. Tables 21-22 and Figures 38 and 40 show the results (the estimated demand patterns appear in figures 7,8 and the estimated results appear in figures 27,30).

The difference in the expected total penalty is less than 1% for all possible X_O in both cases and the optimal X_O agrees with the original in both cases also.

From these experiments we conclude that the model's solutions are not sensitive for random changes in the expected demand rates, so it could fit to real use from this point of view also.

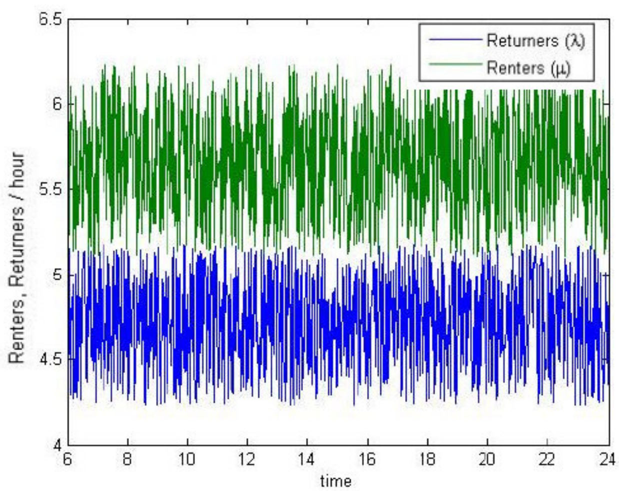

Fig 37. Homogenous – non symmetric problem with random inaccuracies

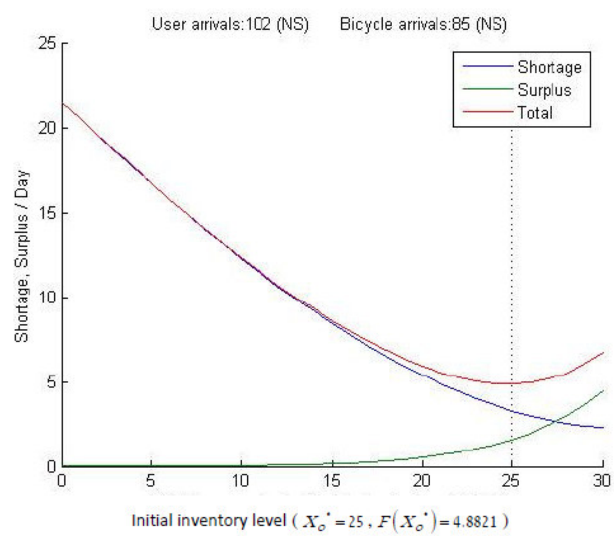

Fig 38. Homogenous – non symmetric problem with random inaccuracies results

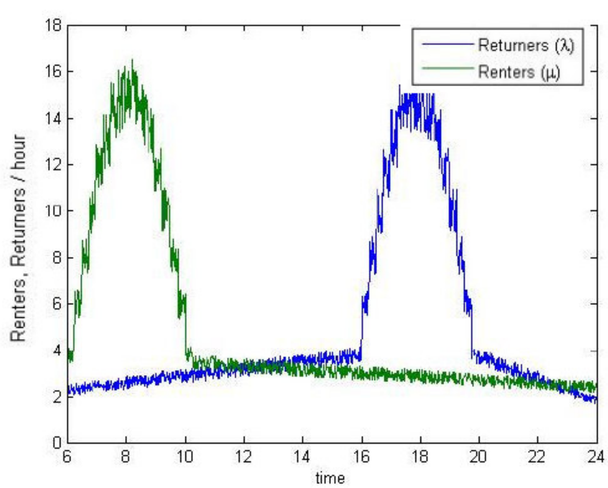

Fig 39. Peaks –symmetric problem with random inaccuracies

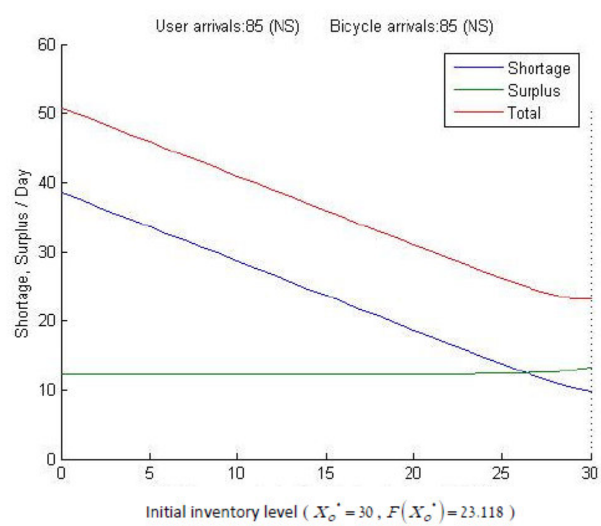

Fig 40. Peaks –symmetric problem with random inaccuracies results

Table 21. Homogenous – non symmetric problem with random inaccuracies results

X_o	$F(X_o)$ - with noise	Difference (%) from original
0	21.5016	0.015815278
1	20.5571	0.018001888
2	19.5754	0.013283741
3	18.605	0.010750954
4	17.6481	0.006800061
5	16.7066	0.003591524
6	15.7822	0.001267234
7	14.8771	0.007393367
8	13.9934	0.01429041
9	13.1332	0.022837655
10	12.2988	0.031700358
11	11.4926	0.04261796
12	10.7169	0.054090854
13	9.9742	0.067128215
14	9.2671	0.081943351
15	8.5982	0.10108169
16	7.9706	0.121549315
17	7.3872	0.147335127
18	6.8517	0.179195804
19	6.368	0.21936697
20	5.9407	0.266930799
21	5.5751	0.330735126
22	5.2777	0.407601004
23	5.0564	0.493948637
24	4.9206	0.587914419
25	4.8821	0.683524218
26	4.9553	0.770955985
27	5.1578	0.836329379
28	5.5111	0.881279114
29	6.041	0.910358402
30	6.7448	0.89483815

Table 22. Peaks –symmetric problem with random inaccuracies results

X_o	$F(X_o)$ - with noise	Difference (%) from original
0	50.8514	0.070844665
1	49.8901	0.072009981
2	48.9002	0.073878369
3	47.9083	0.074991018
4	46.915	0.076579968
5	45.9207	0.07802131
6	44.9256	0.079973802
7	43.9298	0.081560122
8	42.9335	0.083687666
9	41.9366	0.085439896
10	40.9394	0.08776755
11	39.9418	0.089961635
12	38.9439	0.092268942
13	37.9459	0.094697969
14	36.9478	0.09698743
15	35.95	0.09968202
16	34.9527	0.102242461
17	33.9565	0.105245159
18	32.962	0.108119928
19	31.9702	0.110850859
20	30.9825	0.113741748
21	30.0007	0.116466105
22	29.027	0.118996292
23	28.0644	0.121296878
24	27.1168	0.12295309
25	26.1895	0.123866942
26	25.2914	0.123514461
27	24.4413	0.119202694
28	23.6879	0.106919781
29	23.1569	0.08168381
30	23.118	0.055831829

3.3.2.9.2 Renters and returners arrival volume inaccuracies

Another common phenomenon which occurs in service systems is the difficulty to estimate the arrival volume for the planning horizon (Steckley et al. 2009). Thus, the shrinkage or the inflation of the arrival volume in practice, compared to the estimations, can occur. We assume that due to the fact that all of the rates are equally biased, the optimal solution would remain approximately in the same position as the optimal solution for the expected rates. This is because the renters and returners demands balance each other. In this subsection two problems (the *homogenous – non symmetric* problem and the *Peaks – non symmetric* problem) were tested so that the renters and returners demand rates in practice were shrunk or inflated up to 20% (in 5% intervals) compared to the estimations. Figures 41 and 43 describe the optimal X_O of the *homogenous – non symmetric* problem and the *Peaks – non symmetric* problem respectively, for every change of rates, including when the rates are as expected. The graphs presented in Figures 42 and 44 show the difference between the optimal expected total penalty calculated by the procedure on the basis of the estimated demand, and the optimal expected total penalty in practice. On the horizontal axis we see the rates increase / decrease. The vertical axis presents the difference between the expected total penalty in practice (that is based on the optimal X_O that was calculated for the expected rates) and the actual optimal expected total penalty (that could have been achieved if the rates were estimated correctly).

In Figure 41 we see that for the *homogenous – non symmetric* problem the optimal X_O is in most cases the same, as can be expected in the case of homogenous rates. In the -10%, -15% and -20% shrinkage cases, the change is of only one locker (24 instead of 25). This can be explained by the decrease in the intensity of the renters demand rate (which is 1.2 times bigger than the returners demand rate) allowing the placement of less bicycles in the station. In Figure 42 an important meaning to the differences mentioned above is seen. In the cases in which the optimal X_O remains the same, there is naturally no difference between the optimal expected total penalty in practice and the optimal expected total penalty calculated by the procedure. In the -10%, -15% and -20% shrinkage cases, we see that the differences are minor and are always less then 2%. In Table 23, the largest possible change in the expected total penalty caused by an improper decision (compared to the optimal one) is presented for each one of the inflation and shrinkage cases. It is shown that extreme changes could be made by a wrong decision. This strengthens the conclusion that the expected optimal solution is not sensitive with inaccuracies while other policies might be very sensitive.

In Figure 43 we see that for the *Peaks – symmetric* problem the optimal X_O remains the same as the rates are inflated, and decreases by one locker for all cases as the rates decrease. This can be explained by the behavior of the morning peak of renters demand. It was previously mentioned that the morning peak is the dominant factor in the calculation of the optimal X_O, due to the need of placing the proper amount of bicycles for the users that arrive at the station in the morning. When the peak's intensity decreases, the afternoon peak has bigger influence on the optimal X_O calculation, and indeed we see that the optimal X_O decreases. However, Figure 44 shows that the differences of the expected total penalty are minor (less then 1.8 % for all cases).

Exactly as for the *homogenous – non symmetric* problem case, it is seen in Table 24 that extreme changes can be made by a wrong decision. This outcome also supports the quality of the expected optimal solution.

This second set of experiments brings us to the conclusion that the approximation procedure produces qualitative results and it can be a reliable tool, also in the sense of shrinkage or inflation of the expected rates.

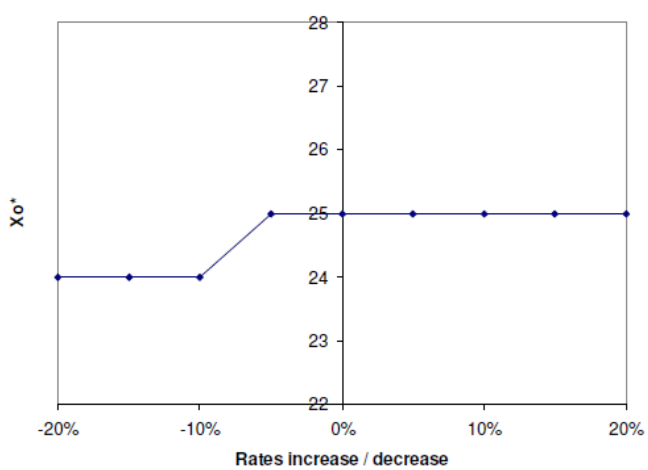

Fig 41. Homogenous – non symmetric problem X_O^* as function of the rates increase / decrease

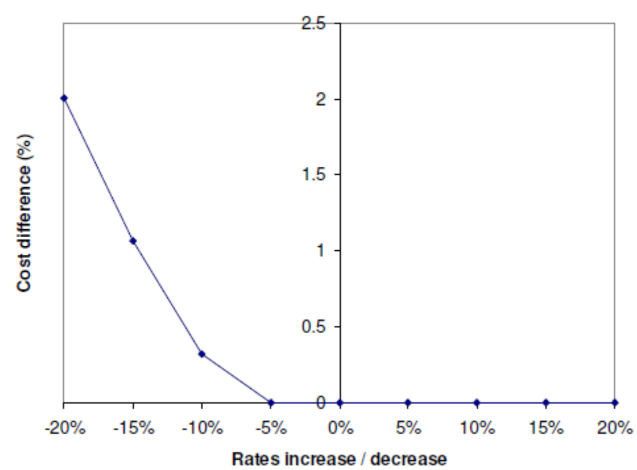

Fig 42. Homogenous – non symmetric problem $F(X_O)$ difference cause by the use of X_O^* calculated from the expected rates

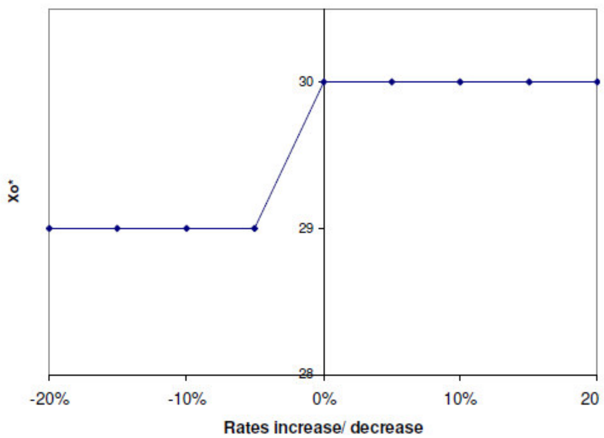

Fig 43. Peaks – symmetric problem X_o^* as function of the rates increase / decrease

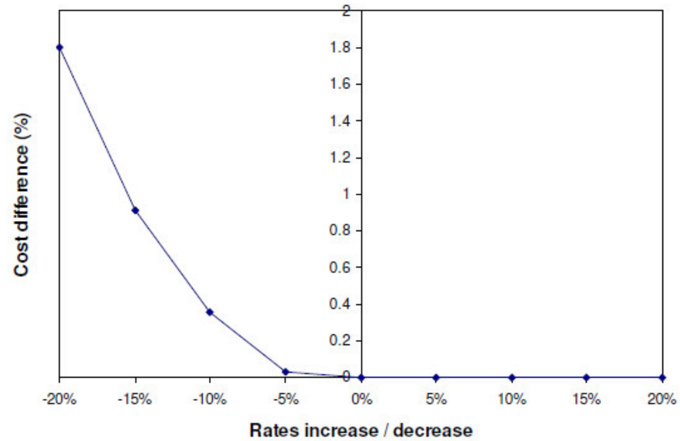

Fig 44. Peaks – symmetric problem $F(X_o)$ difference cause by the use of X_o^* calculated from the expected rates

Table 23. Homogenous – non symmetric problem largest changes of $F(X_o)$ due to system's inflation / shrinkage

Rates increase/decrease	Largest gap of $F(X_o)$ (%)
-20%	484.9878
-15%	439.2875
-10%	399.4682
-5%	365.8997
5%	311.8750
10%	289.1251
15%	268.7815
20%	250.5521

Table 24. Peaks – symmetric problem largest changes of $F(X_o)$ due to system's inflation / shrinkage

Rates increase/decrease	Largest gap of $F(X_o)$ (%)
-20%	215.7681
-15%	184.9706
-10%	158.8427
-5%	137.2515
5%	119.9315
10%	94.0225
15%	84.3482
20%	76.2963

3.4 Practical implication from the expected total penalty function's convexity

In this section we point out some implications for the expected total penalty function's convexity.

First, when we are interested in the optimization of a single station and the arrival processes are non Markovian, our approximation procedure does not reflect the true nature of the expected total penalty function. In these cases, the penalty function would probably have to be calculated (or estimated) by simulation, allegedly for every possible initial inventory level, which is a very time consuming mission as was previously shown in §3.3.2.6. Nevertheless, due to the convexity of the expected total penalty function, convex optimization methods such as the Bisection or the

Dichotomous Search can be operated. That way, the optimization process would be much less time consuming.

Second, the convexity property is very meaningful when we are interested in the optimization of the entire system under a bicycles quantity constraint. In this case, using the decreasing marginal benefit of adding a bicycle, we add each time the next pair of bicycles to the station that would benefit most, and repeat this procedure the number of times that is equal to the number of bicycles in the system. That way, we find the number of bicycles that should be replenished at each station.

Moreover, the convexity property of the penalty function allows optimization of the inventory level of a station in the system subject to much more complex constraints. For example, Raviv et al. (2010) used a preliminary result from this study to solve a complex inventory routing problem that aims to reposition bicycles among stations in the system, during the night, so as to minimize the expected total penalty during the next day.

4. The Capacity problem

Recall that the bicycle rental station *Capacity problem* is associated with a long term decision regarding the optimal number of lockers in each station of a Bike-Sharing system. This decision is made by the operators of the system based on relevant data as the expected renters and returners demand rates at any specific station, the value of the service provided by the system and the infrastructure cost.

Throughout this chapter we assume Poisson arrival processes and users with no patience. We use the same method as in §3.3.2 in order to estimate the expected penalty of each possible station size and select a size that minimizes the total cost. For that purpose we look at the values of the expected penalty associated with the optimal initial inventory level at each size.

4.1 Formulation of the Capacity problem objective function

Consistently with the assumption in §2.1 that the replenishment of bicycles is done every fixed period while the system is idle, say every eighteen hours (during the night), the capacity problem is formally stated as follows:

$$\underset{X_O,C}{\text{Min}}\left\{\int_{t=0}^{T}\left[p^A\cdot\mu_t\cdot\pi_{X_o,0}(t)+h^A\cdot\lambda_t\cdot\pi_{X_o,C}(t)\right]dt+IC(C)\right\} \qquad (38)$$

This definition is very similar to the definition of the Replenishment problem, except that here the capacity (C) is a decision variable and the infrastructure cost for the relevant capacity is taken into consideration, in addition to the expected total penalty.

4.2 Calculation tool

The calculation of the objective function is based on the approximation procedure presented in Chapter 3. Given a capacity C = c, the expected total penalty calculation based on the expected shortage penalty and the expected surplus penalty is actually the same as in Chapter 3. The approximation procedure is performed for each value of C in a required range specified by the operator. As a result, we obtain an efficiency frontier which describes the optimal expected total penalty for every given capacity in the range that was required. We conjecture that the marginal benefit of each locker in the station is decreasing, i.e. it decreases slower as C increases. The decreasing expected total penalty is due to the fact that the bigger the station's capacity is, the better (i.e., the smaller) the expected total penalty is. The reason for the Decreasing marginal benefit is due to the decrease in the probability of using an extra locker when the station capacity is big enough. For example, a station with a capacity of 45 lockers with a total of 85 bicycle requests and 85 bicycles returns a day, would not improve the expected total penalty significantly if the capacity increases to 46 lockers. The difference is even smaller in the transition to a capacity of 47 lockers and so on.

The optimum capacity on the specified range can be revealed if the operator has the information regarding the infrastructure cost (IC). In that case, the expected total penalty function and the IC function are summed together so that the expected total cost function is received. It is important to notice that the IC and the expected total penalty functions are both scaled per one day.

It is worth mentioning that the discretization level for this objective function calculation is less important due to the minor changes in the values of the optimal expected total penalties and the optimal X_O for a given capacity C = c. Hence, it would not make significant changes in the results.

In addition to the importance of the IC data, it is very important that the operator will have a reliable forecast of the renters and returners demand patterns. While the total number of renters and returners per day can be identical in two stations, if their patterns are different, the result can be significantly different. To illustrate the last observation, the expected total penalties as a function of the station capacity (assuming optimal initial inventory levels) were calculated for the *homogenous – symmetric* and the *peaks – symmetric* problems. The discretization level was set to 1 minute and the IC was set to 1 per locker per day, that is IC(C) = C. In both of these problems, the daily total number of renters and returners are equal to 85 each, but the demand patterns are significantly different, as shown in Figures 6 and 8.

Figure 45 shows the result for the *homogenous –symmetric* problem – as predicted the expected total penalty function has a decreasing marginal benefit. It is seen that the optimum capacity is 13 with 7 as the optimal X_O. Consequently, it is not profitable to build a station with a capacity larger than 13. It is worth mentioning that the optimal X_O is 7, so that the optimal replenishment is when the station is approximately half full, as in the 30 lockers case.

As for the *peaks – symmetric* problem – we see in Figure 46 that as in the *homogenous –symmetric* problem, the expected total penalty function has a decreasing marginal benefit, although the decrease is slower than the *homogenous – symmetric* case. This is due to the dominant morning peak – there is a better chance that an extra locker will be in use. The optimum capacity is 38 with 37 as optimal X_O, i.e. almost full as in the 30 lockers case.

As we explained above, we see that the optimum capacities for the two test problems are significantly different, although the daily total number of renters and returners are identical. Hence, this tool is important for the use of the system's operators before planning the station capacity and when all the data mentioned in this chapter is available.

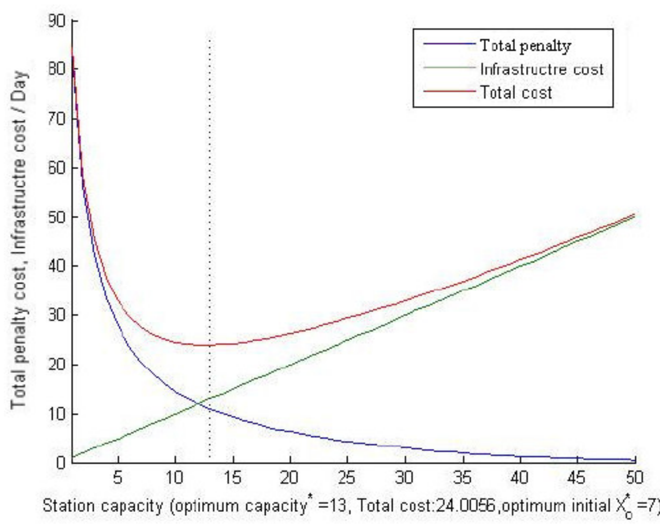

Fig 45. Homogenous – symmetric problem – solution of the Capacity problem

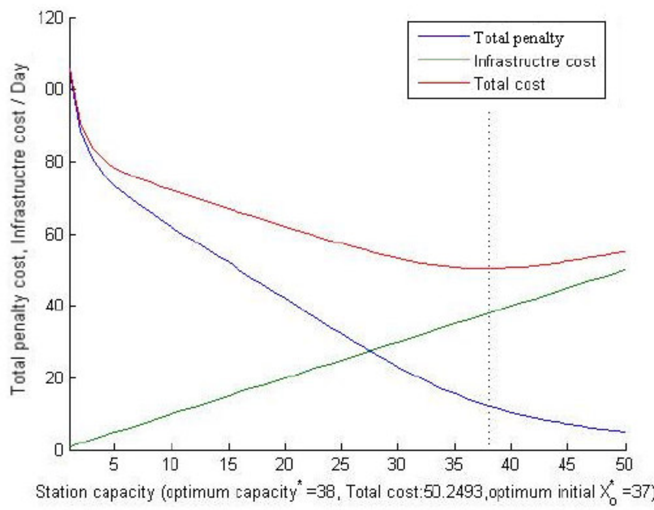

Fig 46. Peaks – symmetric problem – solution of the Capacity problem

Note that it is also possible to extend this model to consider demand cycles that are longer than the review cycles. For example, a weekly demand cycle in a system that is reviewed every night with different demand patterns during the weekend.

5. Conclusions and future research

5.1 Conclusions

Bike-Sharing systems are an emerging industry that grew rapidly over the last few years. The number of cities that already implemented Bike-Sharing systems, or plans to do so, increased significantly and success was proven in major cities throughout the world. As the interest in Bike-Sharing systems increased, companies offering complementary services as equipment or consulting became more involved in the industry. Usage of Bike-Sharing systems is useful to reduce congestion and parking problems in city centers, to reduce air pollution and noise, to encourage healthy life style and to offer an economical transportation alternative. It is also a good complement to other modes of the mass transit systems.

It appears that one of the main problems of the system voiced by users and operators is the lack of bicycles or of lockers in the stations. Consequently, a crucial factor for the success of the system is its ability to meet opposite forces of fluctuating demand for bicycles and for vacant lockers at each station. Therefore, an optimal replenishment is required so that users inconvenience will be minimized. In this study we focused on the inventory part of the problem, i.e. our goal was to find an optimal initial inventory level to be set at each station.

From our review of the Reverse Logistics literature, we found that the proposed models do not consider the dynamics within the planning horizon, which is of interest in Bike-Sharing systems. Furthermore, we found that one-way car sharing systems differ by nature from Bike-Sharing systems so that proposed car sharing models cannot be adopted.

Based on the aforementioned review conclusions, we developed a new method to prescribe the optimal level of the initial inventory. We modeled the system as a series of continuous time Markov chains, each representing a time interval during which the demand rates for bicycle and lockers was assumed to be constant. The probabilities of a station being empty or full at each time interval were calculated, namely, the transient state of the system was of interest. Based on these probabilities, an expected total penalty based on the number of unsatisfied customers was calculated.

A set of experiments was designed with focus on the impatient user behavior model. Through these experiments we wanted to validate our model. Since our model is a discrete approximation of continuous time reality, we considered three discretization levels – 1, 5, and 15 minutes. We then compared the model's result for a set of test problems to a simulation study with 95% confidence level. From the results, we have concluded that our model is valid due to identical graph trends and identical optimal policies (except for one case with a gap of one locker). Moreover, the results of the 1 and 5 discretization levels were almost always in the confidence level.

The computation time required to resolve the model using the proposed method is a very small fraction of the time required for estimation using simulation with an adequate number of replications. For example, for a 30 lockers station with discretization level of 1 minute, the running time of the proposed method is 10,000 times smaller than simulation.

Additionally, we performed a sensitivity analysis to examine the performance of the optimal policy under errors in the forecasted demand for bicycles and lockers. We considered random and biased errors in the forecast, and in both cases we found that the optimal solution is robust under small errors.

In addition, the convexity of the Replenishment problem's objective function was proven for impatient and infinitely-patient user behavior models.

The model that was formulated in this study can contribute to researchers that deal with the routing problem of the replenishment operation, which consists of the decision on the routes that the dedicated fleet of trucks should follow. Indeed, Raviv et al. (2010) construct their objective function based on results of this study.

In addition to the immediate contribution to the domain of Bike-Sharing systems management, we believe that some of the concepts developed in this study are relevant to other inventory management environments. This method may be applicable to other inventory management problems where the dynamics within the planning horizon are of interest and the demand is stochastic with low volume. Furthermore, we showed that steady state considerations are not advisable in these environments and hence transient analysis is required.

Finally, we offer a solution for the long term Capacity problem, i.e. the number of lockers in each station. The solution is based on our model and we calculate an efficiency frontier for the expected total penalty versus the station capacity. This flexible calculation tool contributes to operators that face this problem and can estimate the demand and the infrastructure cost in each area.

5.2 Future research

Since our work is the first to address this problem, it can be extended in many different directions. One is to examine our model on real demand data. As the field of Bike-Sharing is emerging, we see that interest of operators for collaboration is rising so that practical contribution can be achieved. A natural extension in this direction is to consider different demand patterns in different days of the week. This would yield a practical contribution so that a weekly replenishment plan could be outlined and also the capacity of each station could be determined.

Practical contribution can be also achieved by examining a variety of demand patterns from real data. Operational insights obtained by this examination can be especially useful when designing a new system and the demand volume and pattern in each area can only be roughly estimated. Accordingly, stations capacity and preliminary replenishment plans can be outlined.

During the study we saw that finer discretization yields more accurate results at the expense of the computation time. Another possible direction is to reduce the gap between the results achieved in different discretization levels. If the results of gross discretization levels would be very close to those of finer ones, significant calculation time can be saved when considering the optimal initial inventory levels of the stations in the entire system. In our model we use the transition probability to the end of each discretized period. If we would use the transition probability to the midpoint of each discretized period, we would expect the results of different discretization levels to be closer and more accurate. this is done in the following way:

$$F\left(p^A, h^A, \mu(t), \lambda(t), C, T, X_O\right) \approx \sum_{t=0}^{T/d-1} \left[p^A \cdot \mu(t) \cdot \pi_{X_o,0}(t+0.5) + h^A \cdot \lambda(t) \cdot \pi_{X_o,C}(t+0.5) \right] \quad (39)$$

Still on the length of the transition, it is possible to build bounds for the expected total penalty cost. A lower bound can be constructed as

$$\sum_{t=0}^{T/d-1} \left[p^A \cdot \mu(t) \cdot \min\left(\pi_{X_o,0}(t), \pi_{X_o,0}(t+1)\right) + h^A \cdot \lambda(t) \cdot \min\left(\pi_{X_o,C}(t), \pi_{X_o,C}(t+1)\right) \right] \quad (40)$$

and an upper bound can be constructed as

$$\sum_{t=0}^{T/d-1}\left[p^A \cdot \mu(t) \cdot \max\left(\pi_{X_o,0}(t), \pi_{X_o,0}(t+1)\right) + h^A \cdot \lambda(t) \cdot \max\left(\pi_{X_o,C}(t), \pi_{X_o,C}(t+1)\right)\right] \quad (41)$$

The bounds are valid due the fact that during each period the mean rate of renters (resp. returners) abandonments accumulation is either non-increasing or non-decreasing, depending on the relation between $\mu(t)$ and $\lambda(t)$. For example, if the renters arrival rate is greater than the returners arrival rate, $\mu(t) > \lambda(t)$, the probability of the station being empty, $\pi_{X_o,0}$ increases during the period, while the probability of the station being full, $\pi_{X_o,C}$ decreases. Calculating the number of abandonments based on the lower probability between the results is a lower bound, while using the higher probabilities results is an upper bound.

In addition, a more relevant extension when considering real data is to experiment the model when users have limited patience, i.e. a user that is not serviced immediately would not definitely leave as he may wait for service. We formulated this problem in §2.3. The objective function of this extended model is

$$\begin{aligned}\sum_{t=0}^{T/d-1}\Bigg(&\sum_{j=-\infty}^{0}\pi_{X_o,j}(t+0.5)\left[p^A \cdot \mu(t) \cdot \left(1-\beta(I_t)\right) - j \cdot p^W \cdot d\right] \\ &+\sum_{j=C}^{\infty}\pi_{X_o,j}(t+0.5)\left[h^A \cdot \lambda(t) \cdot \left(1-\sigma(I_t)\right) + \left(j-C\right) \cdot h^W \cdot d\right]\Bigg)\end{aligned} \quad (42)$$

The first summation term represents the expected user inconvenience accumulated when the station is empty and, possibly, there is a queue of renters. During such time, abandonments occurred at a rate of $\mu(t) \cdot \left(1-\beta(I_t)\right)$ and each bears a penalty of p^A. In addition, the total waiting time of renters is accumulated in a rate that is equal to the queue length, -j. The second summation term represents the expected user discomfort accumulated when there are no vacant lockers at the station and, possibly, there is a queue of returners. During such time, abandonments occurred at a rate of $\lambda(t) \cdot \left(1-\sigma(I_t)\right)$ and each bears a penalty of h^A. In addition, the total waiting time of returners is accumulated in a rate that is equal to the queue length, C-j. As a preparation for future research, we wrote a proper MATLAB procedure considering the three aforementioned extensions. We assumed that users patience (referred to as *UP* in the sequel) is exponentially distributed and then transforms the infinite chain (Figure 2) to a finite one by truncating states that are not likely to occur. We defined a renter's and a returner's probability to join a queue as follows:

$$\beta(I_t)=\begin{cases}1 & ,I_t>0\\ \text{Prob.}\left(UP>\dfrac{-I_t+1}{\lambda(t)}\right) & ,\text{Else}\end{cases} \tag{43}$$

$$\sigma(I_t)=\begin{cases}1 & ,I_t<C\\ \text{Prob.}\left(UP>\dfrac{I_t-C+1}{\mu(t)}\right) & ,\text{Else}\end{cases} \tag{44}$$

Namely, a user will join a queue if his patience is greater than the expected waiting time in the queue. Note that a more accurate calculation could be based on the rate of the returns and rentals, not only during the current period, but also during future periods. Additionally, from the left of the chain (queue of renters) we decided to truncate all the states that admit $\max_t\left\{\dfrac{\mu(t)\cdot\beta(I_t)}{\lambda(t)+\mu(t)\cdot\beta(I_t)}\le 0.001\right\}$. That is, we truncate all states where the approximated probability that another renter who is willing to join the queue will appear before a returner will appear is negligible (this statement is exact if the arrival rates are constant). Similarly, from the right of the chain (queue of returners) we truncate all the states that admit $\max_t\left\{\dfrac{\lambda(t)\cdot\sigma(I_t)}{\mu(t)+\lambda(t)\cdot\sigma(I_t)}\le 0.001\right\}$. The transition rate from the left most remaining state to the state immediately to its right was adjusted to $\lambda(t)-\mu(t)\cdot\beta(I_t)$. Similarly, the transition rate at the opposite side of the chain was modified to $\mu(t)-\lambda(t)\cdot\sigma(I_t)$. After the truncation the approximated objective function is

$$\sum_{t=1}^{T/d}\left(\sum_{j=-L}^{0}\pi_{X_o,j}(t)\left[p^A\cdot\mu(t)\cdot\left(1-\beta(I_t)\right)-j\cdot p^W\cdot d\right]\right.$$
$$\left.+\sum_{j=C}^{C+U}\pi_{X_o,j}(t)\left[h^A\cdot\lambda(t)\cdot\left(1-\sigma(I_t)\right)+\left(j-C\right)\cdot h^W\cdot d\right]\right) \tag{45}$$

where L (resp., U) denotes an upper bound on the length of the queue of renters (resp., returners). For the procedure, see Appendix F. We also calculated lower and upper bounds in a similar way as in (40) and (41) (for the procedure see Appendix G). Currently we have preliminary results based on one of our test problems (*peaks-symmetric*) and under the assumption that renters and returners patience is exponentially distributed with a mean of 10 minutes. From the results (shown in Figure 47 and Appendix H) we see that the expected total penalty cost for the 1 and 15 discretization levels is almost the same (the largest difference is 0.3233%). Future research should thoroughly examine this extended model's results on real data.

While this study focuses on a single station, another extension can be the consideration of a network of stations, i.e. to decide upon the optimal initial inventory level considering the influence of stations in close proximity to each other. For solving this problem, the total amount of lockers in the network must be considered and proper assumptions on user's transition rate between the stations must be made.

Finally, another direction is to solve the Replenishment problem under a dynamic repositioning assumption. This policy is good when facing extreme peaks of demand that cannot be efficiently handled by static repositioning (for example when the amount of renters is much different from the amount of returners during various hours of the day). For that purpose optimal visiting times must be decided upon.

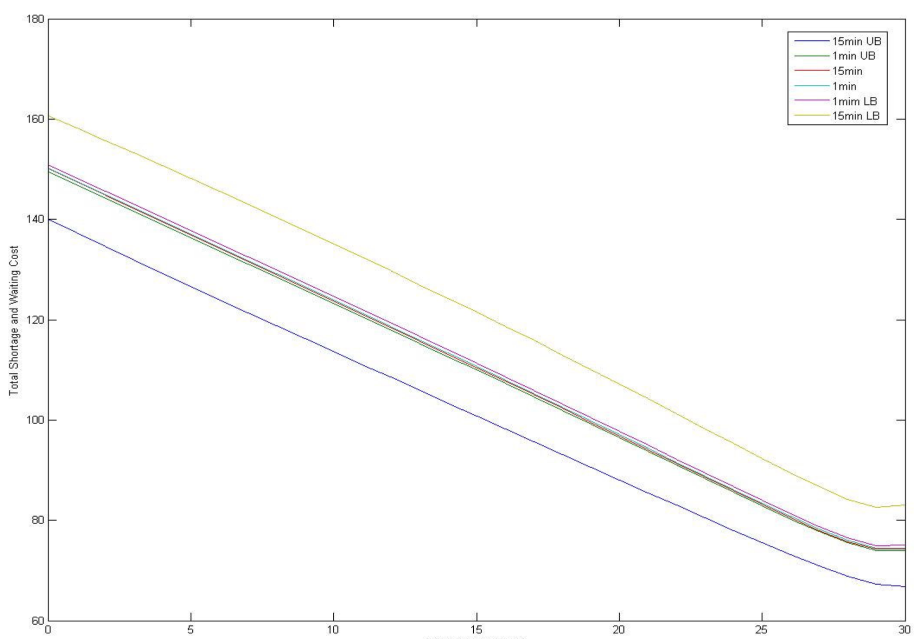

Fig. 47. *Peaks – symmetric* problem - users with limited patience

References

[1] Bar-Eli, A. (2008) "From Amsterdam to Tel Aviv: Israel's 2nd city promotes bicycling" , *Haaretz*, webpage, cited December 2008, http://www.haaretz.com/hasen/spages/943554.html.

[2] Barth M, Todd M, Xue L. " User-Based Vehicle Relocation Techniques for Multiple-Station Shared-Use Vehicle Systems", *The 80th Annual Meeting of the Transportation Research Board*, January 2004.

[3] Becker, B. (2008) " Bicycle-Sharing Program to Be First of Kind in U.S. ", *The New York Times*, webpage, cited December 2008, http://www.nytimes.com/2008/04/27/us/27bikes.html?_r=1&oref=slogin .

[4] Birk M ,Geller R. "Bridging the Gaps: How the Quality and Quantity of a Connected Bikeway Network Correlates with Increasing Bicycle Use", *TRB 2006 Annual Meeting*, p. 26, 2005.

[5] Chemla D, Meunier F, Calvo R. "Bike hiring system: solving the rebalancing problem in the static case", Submitted to Transportation Science, 2010.

[6] Collins, J. (2005) " Bike Rental Program Peddles Smart Cards", *RFID journal*, webpage, cited December 2008, http://www.rfidjournal.com/article/articleview/2013/1/1/ .

[7] Davis J, Massey A, Whitt W. "Sensitivity to the Service-time Distribution in the Nonstationary Erlang Loss Model ", Management Science, Vol. 41, No.6 pp. 1107-1116, 1996.

[8] Demaio, P. "Will Smart Bikes Succeed as Public Transportation in the United States?", Journal of Public Transportation, Vol. 7, No.2, pp. 1-16, 2004.

[9] Demaio, P. "Will Smart Bikes Succeed as Public Transportation in the United States?", *Carbusters*, Vol. 36, pp. 12, 2008.

[10] Demaio, P. (2010), " 2010 Year-end Wrap-up", *bike sharing blog*, webpage, cited March 2011, http://bike-sharing.blogspot.com/2010_12_01_archive.html.

[11] Diaz, O. (2008) "Buenos Aires to Implement a Vélib-style Bicycle Transit System", Institute for Transportation & Policy Development, webpage, cited December 2008, http://www.itdp.org/index.php/news/detail/buenos_aires_to_implement_a_velib_style_bicycle_transit_system/

[12] "Epic Bike-Sharing Post" (2008), *washcycle* , webpage, cited December 2008, http://washcycle.typepad.com/home/2007/11/epic-bike-shari.html .

[13] Fleischmann M, Bloemhof-Ruwaard J,Dekker R, van der Laan E, van Nunen J, Van Wassenhove L. "Quantitative models for reverse logistics: A review", European Journal of Operational Research vol. 103, pp. 1-17, 1997.

[14] Fleischmann M, Kuik R, Dekker R. "Controlling inventories with stochastic item returns: A basic model", European Journal of Operational Research vol. 138, pp. 63-75, 2002.

[15] Fleischmann M, Kuik R. "On optimal inventory control with independent stochastic
item returns", European Journal of Operational Research vol. 151, pp. 25-37, 2003.

[16] Froehlich J, Neumann J, Oliver N. "Sensing and Predicting the Pulse of the City through Shared Bicycling", *IJCAI'09 Proceedings of the 21st international jont conference on Artifical intelligence, San Francisc, USA*, pp. 1420-1426, 2009.

[17] Gardiner, A. (2008) " Public bicycles for inner-city trips", *Herald Sun*, webpage, cited December 2008, http://www.news.com.au/heraldsun/story/0,21985,24772441-2862,00.html .

[18] Halfin S, Whitt W. "Heavy-Traffic Limits for Queues with Many Exponential Servers", Operations Research, Vol. 29, No. 3 pp. 567-588, 1981.

[19] Kek A, Cheu R, Meng Q, Fung C. "A Decision Support Tool for Vehicle Relocation Operations in Carsharing Systems", *The 86th Annual Meeting of the Transportation Research Board*, November 2006.

[20] Mahadevan B, Pyke D, Fleischmann M. "Periodic review, push inventory policies for remanufacturing", European Journal of Operational Research, vol. 151, pp. 536-551, 2003.

[21] Massey A, Whitt W. "Stationary-Process Approximations for the Nonstationary Erlang Loss Model", Operations Research, Vol. 44, No. 6 pp. 976-983, 1996.

[22] Minnesota transportation department. "Mn/DOT Bicycle Modal Plan", 2005.

[23] Mukai N, Watanabe T. "Dynamic Location Management for On-Demand Car Sharing System", Knowledge-Based Intelligent Information and Engineering Systems, *9th International Conference, KES 2005, Melbourne, Australia*, pp. 768-774, September 14-16, 2005.

[24] Nair R, Miller-Hooks E. "Fleet Management for Vehicle Sharing Operations", Transportation Science, pp. 1 – 17, December 2010.
[25] Niches (2008), "New Seamless Mobility Services: Public Bicycles", *the European Commission* , available at : http://www.scribd.com/doc/392213/Public-Bicycles-Policy-Notes.

[26] Oregon Department of Transportation. "Oregon Bicycle and Pedestrian Plan: An Element of the Oregon Transportation Plan" , 1995.

[27] Posner, A. (2008) "Bike-Sharing Goes Global: 5 Programs You Should Know About", *Treehugger*, webpage, cited December 2008, http://www.treehugger.com/files/2008/09/bike-sharing-goes-global-5-bike-sharing-programs-to-know-about.php .
[28] Price, G. (2008) " VéloLiberté - Gordon Price in Paris", *price tags*, webpage, cited December 2008, http://regardingplace.com/?p=546 .

[29] "Public bike systems" (2008), *TMD Encyclopedia*, webpage, cited December 2008, http://www.vtpi.org/tdm/tdm126.htm .

[30] Raviv T, Tzur M, Forma I. "Static Repositioning in a Bike-Sharing System: Models and Solution Approaches", Working paper , 2010.
[31] Richard, M. (2008) "D.C. Bike-Sharing Program Launches Today, First in the USA", *Treehugger*, webpage, cited December 2008, http://www.treehugger.com/files/2008/08/dc-bike-sharing-smart-bikes.php .
[32] Rosental, E. (2008) "European Support for Bicycles Promotes Sharing of the Wheels", *The New York Times*, webpage, cited December 2008, http://www.nytimes.com/2008/11/10/world/europe/10bike.html?_r=2&ref=world&oref=login&oref=slogin .
A version of this article appeared in print on November 10, 2008, on page A10 of the New York edition.
[33] Ross S. "Introduction to probability models, Sixth edition", Academic Press, 1997.
[34] Steckley S, Henderson S, Mehrotra V. "Forecast errors in service systems", Probability in the Engineering and Informational Sciences, Vol. 23, pp. 305-332, 2009.

[35] Tucker, E. (2008) "Bikes make the Democratic National Convention wheely Green", *examiner national*, webpage, cited December 2008, http://www.examiner.com/x-297-Denver-People-Examiner~y2008m8d27-Bikes-make-the-Democratic-National-Convention-wheely-fun .

[36] Uesugi K, Mukai N, Watanabe T. "Optimization of Vehicle Assignment for Car Sharing System", Lecture Notes in Computer Science Vol. 4693, pp. 1105-1111, 2007.

[37] Whitt W. "Heavy-Traffic approximations for service systems with blocking", AT&T Bell Labratories Journal, Vol. 63, No. 5 pp. 689-708, 1984.

[38] Woodland, R. (2008) "50,000 rental bikes for 2008 Beijing Olympics", *bikeradar*, webpage, cited December 2008, http://www.bikeradar.com/fitness/article/50000-rental-bikes-for-2008-beijing-olympics-12164 .

Appendix A: The approximation procedure for the Replenishment problem (written in MATLAB)

Main procedure

```
function [Shortage, Surplus] = Calc_Day_no(n,lambda, mu,Cost_rate);
% This function plot the expected total shortage and surplus over a period with non stationary demand

% n - Station capacity + 1
% lambda - Arrival rate of returners at each period
% mu - Arrival rate of renters at each period
% lambda and mu - Row vectors

M = 300; %The control parameter from the approximation formula
Shortage = zeros(1,n); %Column vector for the expected total shortage penalty for every initial Xo
Surplus =  zeros(1,n); %Column vector for the expected total surplus  penalty for every initial Xo
Pi = eye(n); %Initialization of 'The transition probability matrix', Pi(t), from subsection 3.3.1

T_d = length(lambda);
%Total time periods set by the lenght of lambda and mu (this is parallel to T/d that was mentioned at subsection 3.3.2.1)

%Next is the Calculation of 'The single period transition probability matrix' , P(t) (from susection 3.3.2) and of Pi(t), for
%any given t in [1,T] from the approximation formula
for t = 1: T_d
  R = create_R(n, mu(t), lambda(t)); %Matrix R that was defined in the approximation formula (susection 3.3.2)
  Pi = Pi * calc_P(R, 1,M); %Pi(t+1) is calculated as a markov chain as explained in subsection 3.3.2.  calc_P(R, 1,M) is P(t).
  Shortage = Shortage + Pi(:,1)' * mu(t); %Epected shortage penalty is calcualted from the first column in every Pi(t)
  Surplus = Surplus + Pi(:,n)' * lambda(t)*Cost_rate; %Expected Shortage penalty is calcualted from the Last column in every Pi(t)
end
```

Procedure for calculating matrix *R* from the approximation formula

```
function R = create_R(n, mu, lambda)
%This function builds matrix R from the approximation formula for the current demand rates (mu and lambda)
function R = create_R(n, mu, lambda)
%This function builds matrix R from the approximation formula for the current demand rates (mu and lambda)

R = eye(n)* -(mu+lambda); %Initialzing the main daiagonal with the transition rate (*-1) from each state
for i = 2: n-1 %Setting the transition rate between every two different states
  R(i, i-1) = mu;
  R(i, i+1) = lambda;
end

R(1,1) = -lambda;  %fixing the transition rate from state 0 (*-1)
R(1,2) = lambda; %Inserting the transition rate from state 0 to state 1 (this cell was not includedion the loop)
R(n,n) = -mu; %fixing the transition rate from state C (*-1)
R(n,n-1) = mu; %Inserting the transition rate from state C-1 to state C (this cell was also not included in the loop)
```

Procedure for calculating the *single period transition probability matrix*, P(t)

```
function P = calc_P(R, t,npower)
% This function builds 'The single period transition probability matrix, P(t)
n = size(R,1); %n gets the  dimension size of matrix R
P = (eye(n) + R*(t/npower))^npower;  %Using the approximation formula
```

Appendix B: Renters and returners demand rates for the test problems

Table 25. *Homogenous – symmetric* problem demand rates

Time	Renters rate	Returners rate	Time	Renters rate	Returners rate
6:00-6:15	1.1806	1.1806	15:00-15:15	1.1806	1.1806
6:15-6:30	1.1806	1.1806	15:15-15:30	1.1806	1.1806
6:30-6:45	1.1806	1.1806	15:30-15:45	1.1806	1.1806
6:45-7:00	1.1806	1.1806	15:45-16:00	1.1806	1.1806
7:00-7:15	1.1806	1.1806	16:00-16:15	1.1806	1.1806
7:15-7:30	1.1806	1.1806	16:15-16:30	1.1806	1.1806
7:30-7:45	1.1806	1.1806	16:30-16:45	1.1806	1.1806
7:45-8:00	1.1806	1.1806	16:45-17:00	1.1806	1.1806
8:00-8:15	1.1806	1.1806	17:00-17:15	1.1806	1.1806
8:15-8:30	1.1806	1.1806	17:15-17:30	1.1806	1.1806
8:30-8:45	1.1806	1.1806	17:30-17:45	1.1806	1.1806
8:45-9:00	1.1806	1.1806	17:45-18:00	1.1806	1.1806
9:00-9:15	1.1806	1.1806	18:00-18:15	1.1806	1.1806
9:15-9:30	1.1806	1.1806	18:15-18:30	1.1806	1.1806
9:30-9:45	1.1806	1.1806	18:30-18:45	1.1806	1.1806
9:45-10:00	1.1806	1.1806	18:45-19:00	1.1806	1.1806
10:00-10:15	1.1806	1.1806	19:00-19:15	1.1806	1.1806
10:15-10:30	1.1806	1.1806	19:15-19:30	1.1806	1.1806
10:30-10:45	1.1806	1.1806	19:30-19:45	1.1806	1.1806
10:45-11:00	1.1806	1.1806	19:45-20:00	1.1806	1.1806
11:00-11:15	1.1806	1.1806	20:00-20:15	1.1806	1.1806
11:15-11:30	1.1806	1.1806	20:15-20:30	1.1806	1.1806
11:30-11:45	1.1806	1.1806	20:30-20:45	1.1806	1.1806
11:45-12:00	1.1806	1.1806	20:45-21:00	1.1806	1.1806
12:00-12:15	1.1806	1.1806	21:00-21:15	1.1806	1.1806
12:15-12:30	1.1806	1.1806	21:15-21:30	1.1806	1.1806
12:30-12:45	1.1806	1.1806	21:30-21:45	1.1806	1.1806
12:45-13:00	1.1806	1.1806	21:45-22:00	1.1806	1.1806
13:00-13:15	1.1806	1.1806	22:00-22:15	1.1806	1.1806
13:15-13:30	1.1806	1.1806	22:15-22:30	1.1806	1.1806
13:30-13:45	1.1806	1.1806	22:30-22:45	1.1806	1.1806
13:45-14:00	1.1806	1.1806	22:45-23:00	1.1806	1.1806
14:00-14:15	1.1806	1.1806	23:00-23:15	1.1806	1.1806
14:15-14:30	1.1806	1.1806	23:15-23:30	1.1806	1.1806
14:30-14:45	1.1806	1.1806	23:30-23:45	1.1806	1.1806
14:45-15:00	1.1806	1.1806	23:45-00:00	1.1806	1.1806

Table 26. *Homogenous – non symmetric* problem demand rates

Time	Renters rate	Returners rate	Time	Renters rate	Returners rate
6:00-6:15	1.4167	1.1806	15:00-15:15	1.4167	1.1806
6:15-6:30	1.4167	1.1806	15:15-15:30	1.4167	1.1806
6:30-6:45	1.4167	1.1806	15:30-15:45	1.4167	1.1806
6:45-7:00	1.4167	1.1806	15:45-16:00	1.4167	1.1806
7:00-7:15	1.4167	1.1806	16:00-16:15	1.4167	1.1806
7:15-7:30	1.4167	1.1806	16:15-16:30	1.4167	1.1806
7:30-7:45	1.4167	1.1806	16:30-16:45	1.4167	1.1806
7:45-8:00	1.4167	1.1806	16:45-17:00	1.4167	1.1806
8:00-8:15	1.4167	1.1806	17:00-17:15	1.4167	1.1806
8:15-8:30	1.4167	1.1806	17:15-17:30	1.4167	1.1806
8:30-8:45	1.4167	1.1806	17:30-17:45	1.4167	1.1806
8:45-9:00	1.4167	1.1806	17:45-18:00	1.4167	1.1806
9:00-9:15	1.4167	1.1806	18:00-18:15	1.4167	1.1806
9:15-9:30	1.4167	1.1806	18:15-18:30	1.4167	1.1806
9:30-9:45	1.4167	1.1806	18:30-18:45	1.4167	1.1806
9:45-10:00	1.4167	1.1806	18:45-19:00	1.4167	1.1806
10:00-10:15	1.4167	1.1806	19:00-19:15	1.4167	1.1806
10:15-10:30	1.4167	1.1806	19:15-19:30	1.4167	1.1806
10:30-10:45	1.4167	1.1806	19:30-19:45	1.4167	1.1806
10:45-11:00	1.4167	1.1806	19:45-20:00	1.4167	1.1806
11:00-11:15	1.4167	1.1806	20:00-20:15	1.4167	1.1806
11:15-11:30	1.4167	1.1806	20:15-20:30	1.4167	1.1806
11:30-11:45	1.4167	1.1806	20:30-20:45	1.4167	1.1806
11:45-12:00	1.4167	1.1806	20:45-21:00	1.4167	1.1806
12:00-12:15	1.4167	1.1806	21:00-21:15	1.4167	1.1806
12:15-12:30	1.4167	1.1806	21:15-21:30	1.4167	1.1806
12:30-12:45	1.4167	1.1806	21:30-21:45	1.4167	1.1806
12:45-13:00	1.4167	1.1806	21:45-22:00	1.4167	1.1806
13:00-13:15	1.4167	1.1806	22:00-22:15	1.4167	1.1806
13:15-13:30	1.4167	1.1806	22:15-22:30	1.4167	1.1806
13:30-13:45	1.4167	1.1806	22:30-22:45	1.4167	1.1806
13:45-14:00	1.4167	1.1806	22:45-23:00	1.4167	1.1806
14:00-14:15	1.4167	1.1806	23:00-23:15	1.4167	1.1806
14:15-14:30	1.4167	1.1806	23:15-23:30	1.4167	1.1806
14:30-14:45	1.4167	1.1806	23:30-23:45	1.4167	1.1806
14:45-15:00	1.4167	1.1806	23:45-00:00	1.4167	1.1806

Table 27. *Peaks – symmetric* problem demand rates

Time	Renters rate	Returners rate	Time	Renters rate	Returners rate
6:00-6:15	0.9389	0.5735	15:00-15:15	0.769	0.912
6:15-6:30	1.4959	0.5829	15:15-15:30	0.7643	0.9214
6:30-6:45	2.0269	0.5923	15:30-15:45	0.7596	0.9308
6:45-7:00	2.5163	0.6017	15:45-16:00	0.7549	0.9402
7:00-7:15	2.9453	0.6111	16:00-16:15	0.7502	1.4904
7:15-7:30	3.2974	0.6205	16:15-16:30	0.7455	2.0195
7:30-7:45	3.559	0.6299	16:30-16:45	0.7407	2.5072
7:45-8:00	3.72	0.6393	16:45-17:00	0.736	2.9346
8:00-8:15	3.7744	0.6487	17:00-17:15	0.7313	3.2853
8:15-8:30	3.72	0.6581	17:15-17:30	0.7266	3.546
8:30-8:45	3.559	0.6675	17:30-17:45	0.7219	3.7065
8:45-9:00	3.2974	0.6769	17:45-18:00	0.7171	3.7607
9:00-9:15	2.9453	0.6863	18:00-18:15	0.7124	3.7065
9:15-9:30	2.5163	0.6957	18:15-18:30	0.7077	3.546
9:30-9:45	2.0269	0.7051	18:30-18:45	0.703	3.2853
9:45-10:00	1.4959	0.7145	18:45-19:00	0.6983	2.9346
10:00-10:15	0.9436	0.7239	19:00-19:15	0.6936	2.5072
10:15-10:30	0.8587	0.7333	19:15-19:30	0.6888	2.0195
10:30-10:45	0.854	0.7427	19:30-19:45	0.6841	1.4904
10:45-11:00	0.8492	0.7521	19:45-20:00	0.6794	0.9402
11:00-11:15	0.8445	0.7615	20:00-20:15	0.6747	0.9108
11:15-11:30	0.8398	0.7709	20:15-20:30	0.67	0.8814
11:30-11:45	0.8351	0.7803	20:30-20:45	0.6652	0.852
11:45-12:00	0.8304	0.7897	20:45-21:00	0.6605	0.8226
12:00-12:15	0.8257	0.7991	21:00-21:15	0.6558	0.7933
12:15-12:30	0.8209	0.8085	21:15-21:30	0.6511	0.7639
12:30-12:45	0.8162	0.8179	21:30-21:45	0.6464	0.7345
12:45-13:00	0.8115	0.8273	21:45-22:00	0.6417	0.7051
13:00-13:15	0.8068	0.8367	22:00-22:15	0.6369	0.6757
13:15-13:30	0.8021	0.8461	22:15-22:30	0.6322	0.6464
13:30-13:45	0.7974	0.8556	22:30-22:45	0.6275	0.617
13:45-14:00	0.7926	0.865	22:45-23:00	0.6228	0.5876
14:00-14:15	0.7879	0.8744	23:00-23:15	0.6181	0.5582
14:15-14:30	0.7832	0.8838	23:15-23:30	0.6133	0.5288
14:30-14:45	0.7785	0.8932	23:30-23:45	0.6086	0.4995
14:45-15:00	0.7738	0.9026	23:45-00:00	0.6039	0.4701

Table 28. *Peaks – non symmetric* problem demand rates

Time	Renters rate	Returners rate	Time	Renters rate	Returners rate
6:00-6:15	0.9389	1.147	15:00-15:15	0.769	1.8239
6:15-6:30	1.4959	1.1658	15:15-15:30	0.7643	1.8427
6:30-6:45	2.0269	1.1846	15:30-15:45	0.7596	1.8615
6:45-7:00	2.5163	1.2034	15:45-16:00	0.7549	1.8803
7:00-7:15	2.9453	1.2222	16:00-16:15	0.7502	2.9808
7:15-7:30	3.2974	1.241	16:15-16:30	0.7455	4.0391
7:30-7:45	3.559	1.2598	16:30-16:45	0.7407	5.0143
7:45-8:00	3.72	1.2786	16:45-17:00	0.736	5.8691
8:00-8:15	3.7744	1.2974	17:00-17:15	0.7313	6.5707
8:15-8:30	3.72	1.3162	17:15-17:30	0.7266	7.0919
8:30-8:45	3.559	1.335	17:30-17:45	0.7219	7.4129
8:45-9:00	3.2974	1.3538	17:45-18:00	0.7171	7.5213
9:00-9:15	2.9453	1.3726	18:00-18:15	0.7124	7.4129
9:15-9:30	2.5163	1.3914	18:15-18:30	0.7077	7.0919
9:30-9:45	2.0269	1.4102	18:30-18:45	0.703	6.5707
9:45-10:00	1.4959	1.4291	18:45-19:00	0.6983	5.8691
10:00-10:15	0.9436	1.4479	19:00-19:15	0.6936	5.0143
10:15-10:30	0.8587	1.4667	19:15-19:30	0.6888	4.0391
10:30-10:45	0.854	1.4855	19:30-19:45	0.6841	2.9808
10:45-11:00	0.8492	1.5043	19:45-20:00	0.6794	1.8803
11:00-11:15	0.8445	1.5231	20:00-20:15	0.6747	1.8216
11:15-11:30	0.8398	1.5419	20:15-20:30	0.67	1.7628
11:30-11:45	0.8351	1.5607	20:30-20:45	0.6652	1.7041
11:45-12:00	0.8304	1.5795	20:45-21:00	0.6605	1.6453
12:00-12:15	0.8257	1.5983	21:00-21:15	0.6558	1.5865
12:15-12:30	0.8209	1.6171	21:15-21:30	0.6511	1.5278
12:30-12:45	0.8162	1.6359	21:30-21:45	0.6464	1.469
12:45-13:00	0.8115	1.6547	21:45-22:00	0.6417	1.4102
13:00-13:15	0.8068	1.6735	22:00-22:15	0.6369	1.3515
13:15-13:30	0.8021	1.6923	22:15-22:30	0.6322	1.2927
13:30-13:45	0.7974	1.7111	22:30-22:45	0.6275	1.234
13:45-14:00	0.7926	1.7299	22:45-23:00	0.6228	1.1752
14:00-14:15	0.7879	1.7487	23:00-23:15	0.6181	1.1164
14:15-14:30	0.7832	1.7675	23:15-23:30	0.6133	1.0577
14:30-14:45	0.7785	1.7863	23:30-23:45	0.6086	0.9989
14:45-15:00	0.7738	1.8051	23:45-00:00	0.6039	0.9402

Table 29. *Random – symmetric* problem demand rates

Time	Renters rate	Returners rate	Time	Renters rate	Returners rate
6:00-6:15	2.1469	1.8302	15:00-15:15	1.9121	1.1564
6:15-6:30	0.5223	1.2399	15:15-15:30	1.1866	1.3981
6:30-6:45	1.3712	0.8085	15:30-15:45	0.4579	0.4563
6:45-7:00	1.0981	1.5338	15:45-16:00	1.5187	0.829
7:00-7:15	2.014	1.193	16:00-16:15	1.8938	1.7097
7:15-7:30	1.722	0.971	16:15-16:30	0.0444	1.4861
7:30-7:45	1.0314	1.516	16:30-16:45	1.5394	1.0064
7:45-8:00	0.0418	1.3561	16:45-17:00	0.8575	1.2394
8:00-8:15	1.856	1.7348	17:00-17:15	1.8795	1.7335
8:15-8:30	1.0048	2.0885	17:15-17:30	1.1361	0.1292
8:30-8:45	1.3906	1.1406	17:30-17:45	1.6031	1.3159
8:45-9:00	1.7894	1.9211	17:45-18:00	0.9691	0.1097
9:00-9:15	2.0829	0.3775	18:00-18:15	0.6883	0.9066
9:15-9:30	1.668	2.1385	18:15-18:30	0.4285	0.6657
9:30-9:45	0.3983	0.5925	18:30-18:45	0.4371	1.9085
9:45-10:00	0.9167	0.5508	18:45-19:00	1.5415	0.0328
10:00-10:15	2.1138	1.9115	19:00-19:15	0.6841	1.6762
10:15-10:30	2.0718	1.6093	19:15-19:30	1.224	2.119
10:30-10:45	0.927	0.298	19:30-19:45	0.3409	2.161
10:45-11:00	2.0193	0.0257	19:45-20:00	1.577	1.7218
11:00-11:15	0.1308	1.9511	20:00-20:15	0.855	0.9574
11:15-11:30	0.7973	0.4347	20:15-20:30	1.9433	1.0876
11:30-11:45	1.8374	0.652	20:30-20:45	1.9289	0.467
11:45-12:00	0.0223	1.4437	20:45-21:00	1.3412	1.4045
12:00-12:15	0.3138	0.6208	21:00-21:15	1.122	0.6985
12:15-12:30	0.4582	1.0242	21:15-21:30	2.0331	2.0956
12:30-12:45	0.449	0.1414	21:30-21:45	1.8565	1.586
12:45-13:00	1.3643	2.1572	21:45-22:00	1.4572	0.8992
13:00-13:15	0.615	1.272	22:00-22:15	1.8483	1.6251
13:15-13:30	0.4492	0.9244	22:15-22:30	1.4918	0.5848
13:30-13:45	0.0345	1.1252	22:30-22:45	0.7727	0.9602
13:45-14:00	1.6874	0.7289	22:45-23:00	0.6547	2.0373
14:00-14:15	1.0057	0.9449	23:00-23:15	0.771	1.4915
14:15-14:30	2.1055	0.4932	23:15-23:30	1.2068	0.4639
14:30-14:45	1.053	1.2655	23:30-23:45	1.643	1.8318
14:45-15:00	0.946	1.6596	23:45-00:00	0.6989	1.3724

Appendix C: Renters and returners demand rates discretization converting procedure

** This procedure is used to convert the demand data to the required discretization level

```
function ret = expand_demand(V, f)
%V is the vector to expand
%f is (data_interval/d) where 'd' is the discretization level and 'data_interval' is the time interval in which the demand data is given in

n = length(V); %n gets the lenght of the vector
ret = zeros(1,n*f); % ret is the expended vector which lenght is determined by the multipication of n and f
for i = 0:(n-1)
  for j = 1:f
    ret(i*f+j) = V(i+1); %A proper amount of cells (f cells) from the expended vector, ret, are allocated for each cell of the original vector, V.
  end
end

ret = ret / sum(ret); %the value  of each cell of the expanded vector, ret, is normalized so the sum of the vector will be 1.
```

Appendix D: Complete results of the simulation study compared to the results of the approximation procedure

Table 30. *Homogenous –symmetric* problem validation

X_O	$F(X_O)$ - CI (simulation)	$F(X_O)$ - (simulation)	$F(X_O)$ + CI (simulation)	Standard Deviation	15 min. disc. compatibility	5 min. disc. compatibility	1 min. disc. compatibility
0	9.6274	10.101	10.5746	7.6411		*	*
1	8.6874	9.157	9.6266	7.5766	*	*	*
2	7.8258	8.287	8.7482	7.4410	*	*	*
3	7.0312	7.481	7.9308	7.2571	*	*	*
4	6.2977	6.734	7.1703	7.0393	*	*	*
5	5.6348	6.056	6.4772	6.7957	*	*	*
6	5.0468	5.451	5.8552	6.5214	*	*	*
7	4.5268	4.913	5.2992	6.2310	*	*	*
8	4.0784	4.446	4.8136	5.9309	*	*	*
9	3.678	4.028	4.378	5.6469	*	*	*
10	3.3357	3.669	4.0023	5.3775	*	*	*
11	3.0548	3.373	3.6912	5.1339	*	*	*
12	2.831	3.141	3.451	5.0016	*	*	*
13	2.675	2.975	3.275	4.8402	*	*	*
14	2.579	2.869	3.159	4.6789	*	*	*
15	**2.525**	**2.815***	**3.105**	**4.6789**	*	*	*
16	2.548	2.838	3.128	4.6789	*	*	*
17	2.64	2.93	3.22	4.6789	*	*	*
18	2.794	3.094	3.394	4.8402	*	*	*
19	3.005	3.315	3.625	5.0016	*	*	*
20	3.252	3.582	3.912	5.3242	*	*	*
21	3.569	3.909	4.249	5.4856	*	*	*
22	3.938	4.298	4.658	5.8083	*	*	*
23	4.369	4.749	5.129	6.1309	*	*	*
24	4.8785	5.277	5.6755	6.4294	*	*	*
25	5.4562	5.872	6.2878	6.7085	*	*	*
26	6.1116	6.543	6.9744	6.9602	*	*	*
27	6.8243	7.27	7.7157	7.1910	*	*	*
28	7.6108	8.068	8.5252	7.3765	*	*	*
29	8.4786	8.944	9.4094	7.5088	*	*	*
30	9.4226	9.892	10.3614	7.5733	*	*	*

*	- The result is in the confidence interval
	- The result is not in the confidence

CI = confidence interval

Table 31. *Homogenous – non symmetric* problem validation

X_O	$F(X_O)$ - CI (simulation)	$F(X_O)$ - (simulation)	$F(X_O)$ + CI (simulation)	Standard Deviation	15 min. disc. compatibility	5 min. disc. compatibility	1 min. disc. compatibility
0	20.8009	21.51	22.2191	11.4406	*	*	*
1	19.8087	20.517	21.2253	11.4277	*	*	*
2	18.8298	19.536	20.2422	11.3938	*	*	*
3	17.8695	18.572	19.2745	11.3341	*	*	*
4	16.9212	17.619	18.3168	11.2583	*	*	*
5	15.9879	16.68	17.3721	11.1663	*	*	*
6	15.0685	15.754	16.4395	11.0599	*	*	*
7	14.1748	14.852	15.5292	10.92	*	*	*
8	13.3023	13.97	14.6377	10.772	*	*	*
9	12.4583	13.115	13.7717	10.595	*	*	*
10	11.6513	12.295	12.9387	10.385	*	*	*
11	10.8757	11.505	12.1343	10.153	*	*	*
12	10.128	10.742	11.356	9.9063	*	*	*
13	9.4188	10.016	10.6132	9.6352	*	*	*
14	8.7476	9.327	9.9064	9.348	*	*	*
15	8.1142	8.675	9.2358	9.0479	*	*	*
16	7.511	8.053	8.595	8.74466	*	*	*
17	6.9493	7.472	7.9947	8.43327	*	*	*
18	6.437	6.94	7.443	8.1154	*	*	*
19	5.9714	6.455	6.9386	7.8024	*	*	*
20	5.5689	6.033	6.4971	7.4878	*	*	*
21	5.2337	5.679	6.1243	7.1845	*	*	*
22	4.9652	5.393	5.8208	6.9021	*	*	*
23	4.771	5.183	5.595	6.6472	*	*	*
24	4.6606	5.059	5.4574	6.42781	*	*	*
25	**4.636**	**5.026***	**5.416**	**6.29228**	*	*	*
26	4.712	5.092	5.472	6.13094	*	*	*
27	4.91	5.28	5.65	5.9696	*	*	*
28	5.263	5.633	6.003	5.9696	*	*	*
29	5.797	6.177	6.557	6.13094	*	*	*
30	6.537	6.917	7.297	6.13094		*	*

*	- The result is in the confidence interval
	- The result is not in the confidence

CI = confidence interval

Table 32. *Peaks –symmetric* problem validation

X_O	**$F(X_O)$ - CI (simulation)**	**$F(X_O)$ - (simulation)**	**$F(X_O)$ + CI (simulation)**	**Standard Deviation**	**15 min. disc. compatibility**	**5 min. disc. compatibility**	**1 min. disc. compatibility**
0	50.6785	51.354	52.0295	10.8986	*	*	*
1	49.6785	50.354	51.0295	10.8986	*	*	*
2	48.6785	49.354	50.0295	10.8986	*	*	*
3	47.6785	48.354	49.0295	10.8986		*	*
4	46.6785	47.354	48.0295	10.8986		*	*
5	45.6785	46.354	47.0295	10.8986		*	*
6	44.6785	45.354	46.0295	10.8986		*	*
7	43.6785	44.354	45.0295	10.8986		*	*
8	42.6785	43.354	44.0295	10.8986		*	*
9	41.6785	42.354	43.0295	10.8986		*	*
10	40.6785	41.354	42.0295	10.8986		*	*
11	39.6785	40.354	41.0295	10.8986		*	*
12	38.6785	39.354	40.0295	10.8986		*	*
13	37.6785	38.354	39.0295	10.8986		*	*
14	36.6785	37.354	38.0295	10.8986		*	*
15	35.6785	36.354	37.0295	10.8986		*	*
16	34.6785	35.354	36.0295	10.8986		*	*
17	33.6785	34.354	35.0295	10.8986		*	*
18	32.6807	33.356	34.0313	10.8953		*	*
19	31.6917	32.366	33.0403	10.8792		*	*
20	30.7137	31.386	32.0583	10.8469		*	*
21	29.7376	30.408	31.0784	10.8163		*	*
22	28.7677	29.436	30.1043	10.7824		*	*
23	27.8037	28.47	29.1363	10.7501		*	*
24	26.8498	27.514	28.1782	10.7162		*	*
25	25.9169	26.578	27.2391	10.6662		*	*
26	24.9987	25.656	26.3133	10.6049		*	*
27	24.1241	24.777	25.4299	10.5339		*	*
28	23.3424	23.989	24.6356	10.4323		*	*
29	22.774	23.414	24.054	10.3258		*	*
30	**22.6732**	**23.309***	**23.9448**	**10.2580**		*	*

*	- The result is in the confidence interval
	- The result is not in the confidence

CI = confidence interval

Table 33. *Peaks –non symmetric* problem validation

X_O	$F(X_O)$ - CI (simulation)	$F(X_O)$ - (simulation)	$F(X_O)$ + CI (simulation)	Standard Deviation	15 min. disc. compatibility	5 min. disc. compatibility	1 min. disc. compatibility
0	107.2215	107.769	108.3165	15.2999		*	*
1	106.2229	106.7703	107.3177	15.2971		*	*
2	105.2243	105.7717	106.3191	15.2971		*	*
3	104.2271	104.7743	105.3215	15.2915		*	*
4	103.2319	103.779	104.3261	15.2887		*	*
5	102.2402	102.787	103.3338	15.2803		*	*
6	101.2498	101.7963	102.3428	15.2720		*	*
7	100.2607	100.807	101.3533	15.2664		*	*
8	99.275	99.821	100.367	15.2580		*	*
9	98.2976	98.843	99.3884	15.2412		*	*
10	97.3312	97.8757	98.4202	15.2161		*	*
11	96.3777	96.921	97.4643	15.1825		*	*
12	95.4379	95.9797	96.5215	15.1406		*	*
13	94.5237	95.0637	95.6037	15.0903		*	*
14	93.6487	94.1863	94.7239	15.0232		*	*
15	92.8013	93.3363	93.8713	14.9506		*	*
16	91.9954	92.5277	93.06	14.8751		*	*
17	91.2501	91.779	92.3079	14.7801		*	*
18	90.5643	91.0897	91.6151	14.6823		*	*
19	89.9365	90.4583	90.9801	14.5817		*	*
20	89.3804	89.899	90.4176	14.4923		*	*
21	88.9097	89.4257	89.9417	14.4196		*	*
22	88.5453	89.059	89.5727	14.3554		*	*
23	88.2701	88.7823	89.2945	14.3134		*	*
24	88.1046	88.6157	89.1268	14.2827		*	*
25	**88.0485**	**88.5597***	**89.0709**	**14.2855**		*	*
26	88.1007	88.613	89.1253	14.3162		*	*
27	88.2898	88.803	89.3162	14.3414		*	*
28	88.6529	89.1677	89.6825	14.3861		*	*
29	89.2044	89.721	90.2376	14.4364		*	*
30	89.9853	90.503	91.0207	14.4671	*	*	

*	- The result is in the confidence interval
	- The result is not in the confidence

CI = confidence interval

Table 34. *Random –symmetric* problem validation

X_O	**$F(X_O)$ - CI (simulation)**	**$F(X_O)$ - (simulation)**	**$F(X_O)$ + CI (simulation)**	**Standard Deviation**	**15 min. disc. compatibility**	**5 min. disc. compatibility**	**1 min. disc. compatibility**
0	10.1381	10.648	11.1579	8.2268		*	*
1	9.1756	9.683	10.1904	8.1864		*	*
2	8.2767	8.778	9.2793	8.0880		*	*
3	7.445	7.937	8.429	7.9380		*	*
4	6.6911	7.171	7.6509	7.7427		*	*
5	6.0188	6.484	6.9492	7.5056		*	*
6	5.4066	5.856	6.3054	7.2507		*	*
7	4.8709	5.303	5.7351	6.9715		*	*
8	4.4193	4.833	5.2467	6.6747		*	*
9	4.0181	4.414	4.8099	6.3875		*	*
10	3.6937	4.072	4.4503	6.1035		*	*
11	3.4301	3.792	4.1539	5.8389		*	*
12	3.2259	3.573	3.9201	5.6001		*	*
13	3.0814	3.416	3.7506	5.3985		*	*
14	2.9786	3.304	3.6294	5.2500		*	*
15	**2.9292**	**3.249***	**3.5688**	5.1597		*	*
16	2.9436	3.261	3.5784	5.1210		*	*
17	3.0072	3.326	3.6448	5.1435		*	*
18	3.1323	3.456	3.7797	5.2226		*	*
19	3.3339	3.665	3.9961	5.3420		*	*
20	3.5802	3.922	4.2638	5.5146		*	*
21	3.877	4.232	4.587	5.7276		*	*
22	4.2477	4.617	4.9863	5.9583		*	*
23	4.6917	5.076	5.4603	6.2003		*	*
24	5.1984	5.598	5.9976	6.4472		*	*
25	5.7503	6.166	6.5817	6.7069		*	*
26	6.3753	6.806	7.2367	6.9489		*	*
27	7.0667	7.511	7.9553	7.1684		*	*
28	7.8321	8.288	8.7439	7.3555		*	*
29	8.6942	9.157	9.6198	7.4668		*	*
30	9.6125	10.08	10.5475	7.5427	*	*	*

*	- The result is in the confidence interval
	- The result is not in the confidence

CI = confidence interval

Appendix E: complete results of the comparison between the steady state analysis and the approximation procedure for the *'peak – symmetric'* problem

Table 35. Approximation procedure results

X_O	$G_-(X_O)$	$G_+(X_O)$	$F(X_O)$
0	6349.1	6323.7	12672.8
1	6348.1	6323.7	12671.8
2	6347.1	6323.7	12670.8
3	6346.1	6323.7	12669.8
4	6345.1	6323.7	12668.8
5	6344.1	6323.7	12667.8
6	6343.2	6323.7	12666.9
7	6342.2	6323.7	12665.9
8	6341.2	6323.7	12664.9
9	6340.2	6323.7	12663.9
10	6339.2	6323.7	12662.9
11	6338.2	6323.7	12661.9
12	6337.2	6323.7	12660.9
13	6336.2	6323.7	12659.9
14	6335.2	6323.7	12658.9
15	6334.2	6323.7	12657.9
16	6333.2	6323.7	12656.9
17	6332.2	6323.7	12655.9
18	6331.2	6323.7	12654.9
19	6330.2	6323.8	12654
20	6329.2	6323.8	12653
21	6328.2	6323.8	12652
22	6327.2	6323.8	12651
23	6326.2	6323.8	12650
24	6325.3	6323.8	12649.1
25	6324.3	6323.9	12648.2
26	6323.3	6323.9	12647.2
27	6322.4	6324	12646.4
28	6321.5	6324.1	12645.6
29	6320.8	6324.4	12645.2
30	6320.3	6324.8	**12645.1***

Table 36. Steady state results

Time (per one day)	Renters rate	Returners rate	Expected total penalty per one day	Expected total penalty for 500 days
6:00-6:15	0.9389	0.5735	0.3654	182.7001
6:15-6:30	1.4959	0.5829	0.913	456.5
6:30-6:45	2.0269	0.5923	1.4346	717.3
6:45-7:00	2.5163	0.6017	1.9146	957.3
7:00-7:15	2.9453	0.6111	2.3342	1167.1
7:15-7:30	3.2974	0.6205	2.6769	1338.45
7:30-7:45	3.559	0.6299	2.9291	1464.55
7:45-8:00	3.72	0.6393	3.0807	1540.35
8:00-8:15	3.7744	0.6487	3.1257	1562.85
8:15-8:30	3.72	0.6581	3.0619	1530.95
8:30-8:45	3.559	0.6675	2.8915	1445.75
8:45-9:00	3.2974	0.6769	2.6205	1310.25
9:00-9:15	2.9453	0.6863	2.259	1129.5
9:15-9:30	2.5163	0.6957	1.8206	910.3
9:30-9:45	2.0269	0.7051	1.3218	660.9
9:45-10:00	1.4959	0.7145	0.7814	390.7
10:00-10:15	0.9436	0.7239	0.219819	109.9094
10:15-10:30	0.8587	0.7333	0.127293	63.6467
10:30-10:45	0.854	0.7427	0.114274	57.1369
10:45-11:00	0.8492	0.7521	0.10171	50.85484
11:00-11:15	0.8445	0.7615	0.090002	45.00106
11:15-11:30	0.8398	0.7709	0.079334	39.6668
11:30-11:45	0.8351	0.7803	0.070023	35.01172
11:45-12:00	0.8304	0.7897	0.062414	31.20689
12:00-12:15	0.8257	0.7991	0.056833	28.41655
12:15-12:30	0.8209	0.8085	0.053532	26.76591
12:30-12:45	0.8162	0.8179	0.052731	26.36558
12:45-13:00	0.8115	0.8273	0.054428	27.21386
13:00-13:15	0.8068	0.8367	0.058516	29.25808
13:15-13:30	0.8021	0.8461	0.064775	32.38749
13:30-13:45	0.7974	0.8556	0.072971	36.48544
13:45-14:00	0.7926	0.865	0.082724	41.3621
14:00-14:15	0.7879	0.8744	0.093631	46.81574
14:15-14:30	0.7832	0.8838	0.105465	52.7325
14:30-14:45	0.7785	0.8932	0.117984	58.99189
14:45-15:00	0.7738	0.9026	0.130997	65.49835

Time (per one day)	Renters rate	Returners rate	Expected total penalty per one day	Expected total penalty for 500 days
15:00-15:15	0.769	0.912	0.144454	72.2268
15:15-15:30	0.7643	0.9214	0.158059	79.0295
15:30-15:45	0.7596	0.9308	0.171829	85.91469
15:45-16:00	0.7549	0.9402	0.185711	92.85564
16:00-16:15	0.7502	1.4904	0.7402	370.1
16:15-16:30	0.7455	2.0195	1.274	637
16:30-16:45	0.7407	2.5072	1.7665	883.25
16:45-17:00	0.736	2.9346	2.1986	1099.3
17:00-17:15	0.7313	3.2853	2.554	1277
17:15-17:30	0.7266	3.546	2.8194	1409.7
17:30-17:45	0.7219	3.7065	2.9846	1492.3
17:45-18:00	0.7171	3.7607	3.0436	1521.8
18:00-18:15	0.7124	3.7065	2.9941	1497.05
18:15-18:30	0.7077	3.546	2.8383	1419.15
18:30-18:45	0.703	3.2853	2.5823	1291.15
18:45-19:00	0.6983	2.9346	2.2363	1118.15
19:00-19:15	0.6936	2.5072	1.8136	906.8
19:15-19:30	0.6888	2.0195	1.3307	665.35
19:30-19:45	0.6841	1.4904	0.8063	403.15
19:45-20:00	0.6794	0.9402	0.260822	130.411
20:00-20:15	0.6747	0.9108	0.236143	118.0716
20:15-20:30	0.67	0.8814	0.211486	105.743
20:30-20:45	0.6652	0.852	0.186974	93.48699
20:45-21:00	0.6605	0.8226	0.16246	81.23008
21:00-21:15	0.6558	0.7933	0.138255	69.12744
21:15-21:30	0.6511	0.7639	0.114405	57.20229
21:30-21:45	0.6464	0.7345	0.091521	45.7605
21:45-22:00	0.6417	0.7051	0.070623	35.31135
22:00-22:15	0.6369	0.6757	0.05357	26.78475
22:15-22:30	0.6322	0.6464	0.04286	21.4302
22:30-22:45	0.6275	0.617	0.041056	20.52776
22:45-23:00	0.6228	0.5876	0.049082	24.54122
23:00-23:15	0.6181	0.5582	0.065208	32.60411
23:15-23:30	0.6133	0.5288	0.086224	43.11186
23:30-23:45	0.6086	0.4995	0.109579	54.78938
23:45-00:00	0.6039	0.4701	0.133914	66.95684
Total			**70.03709**	**35018.54**

Appendix F: Users with limited patience procedure

Main procedure

```
function [ShortageA, SurplusA,ShortageW, SurplusW] = Calc_Day(C,lambda, mu,disc,mean)
% Clculate the expected number of shorage and
% surpless events and the expected waiting time of renteres and returners
% (under the assumption that penalties for abandonment and the penalties for
% one unit of waiting time equale 1).
% This version uses MIC
%
% Input
% C - Station capacity
% lambda - Arrival rate of returners at each period (returners/ period)
% mu - Arrival of renters at each period (renters / period)
% lambda and mu - Row vectors
% disc - length in minutes of each period (minutes)
% mean - expected patience in minutes

%
% Output
% ShortageA - expected number of renteres that abandon the system
% SurplusA - expected number of returners that abandon the system
% ShortageW - expected waiting time of renters (minutes)
% SurplusW - expected waiting time of returners (minutes)

M = 300; %The control parameter from the approximation formula
mean = mean/disc;
U=upper_trunc(lambda,mu,mean); %setting the left finite state (part of the infinite chain truncation)
L=lower_trunc(lambda,mu,mean); %setting the right finite state (part of the infinite chain truncation)
n=C+1+L+U; %total number of states including state 0
ShortageA = zeros(1,n); %Column vector for the total shortage cost for every initial Xo
SurplusA = zeros(1,n); %Column vector for the total surplus cost for every initial Xo
ShortageW = zeros(1,n);
SurplusW = zeros(1,n);
P = eye(n); %The matrix P(t)
T = length(lambda); %Total time periods set by the lenght of lambda / mu

for t = 1: T %Calculation of P(t) for any given t in [1,T] from the approximation formula
  R = create_R_new_vec(t,lambda,mu,C,U,L,mean); %Matrix that was defined in the approximation formula

  % Proceede by half a period
  P = P * calc_P(R, 1/2,M); %P(t+1) is calculated as a markov chain as explained in section 4.2
  for j=1: L+1 %considering queue for bikes => states -L to 0
    B=beta_i(L+1-j,mean,lambda(t)); %calculating the probability that a renter will join the queue
    ShortageA = ShortageA + P(:,j)'* mu(t)*(1-B); %Shortage cost is calcualted for every state from -L to 0 for every P(t). L+1-j is the queue length
    ShortageW = ShortageW + P(:,j)'*disc* (L+1-j);
  end
  for j=L+1+C: n %considering queue for stalls => states C to C+U
    S=sigma_i(j-(L+1+C),mean,mu(t)); %calculating the probability that a returner will join the queue
    SurplusA = SurplusA + P(:,j)' * (lambda(t)*(1-S)); %Surplus cost is calcualted for every state from C to C+U for every P(t). j-(L+1+C) is the queue length
    SurplusW = SurplusW + P(:,j)' *disc*(j-(L+1+C));
  end
  % Proceede by another half of a period
  P = P * calc_P(R, 1/2,M);
end

ShortageA=ShortageA(L+1: L+1+C); % Showing only feasible initial inventory levels
SurplusA=SurplusA(L+1: L+1+C);
ShortageW=ShortageW(L+1: L+1+C);
SurplusW=SurplusW(L+1: L+1+C);
```

Upper truncation procedure

```
function U = upper_trunc(lambda,mu,mean)
%Truncating states from the right side of the chain
U=0;
%Start from state -1000 and decrease
for i = 1000:-1:1

  Si =sigma_i(i,mean,lambda); %Calculating the probability that a returner will join the queue

  if max(( lambda .* Si) ./(mu + lambda .* Si)) >0.001 %If the chance for a returner to join the queue is bigger then 0.001 then U=i;
    U=i;
    break
  end

end
```

Lower truncation procedure

```
function L = lower_trunc(lambda,mu,mean)
%Truncating states from the right side of the chain
L=0;
%start from state C+1000 and decrease
for i = 1000:-1:1

  Bi =beta_i(i,mean,lambda); %Calculating the probability that a renter will join the queue

  if max(( mu .* Bi) ./(lambda + mu .* Bi)) >0.001 %If the chance for a renter to join the queue is bigger then 0.001 then U=i;
    L=i;
    break
  end

end
```

Renter's probability to join the queue procedure

```
function Bi = beta_i(i,mean,lambda)
%Calculating the probability that a renter will join the queue
% i - system state (absolute value)
% mean - mean patience
% lambda - rate of bicycles arrival (vector, one element for each time slot)
  Bi=zeros(1,length(lambda));
if mean>0
  Ei=(i+1) ./ lambda;
  Bi = 1-expcdf(Ei,mean);
end
```

Returner's probability to join the queue procedure

```
function Si = sigma_i(i,mean,mu)
%Calculating the probability that a returner will join the queue
% i - system state (absolute value)
% mean - mean patience
% mu - rate of renters arrival (vector, one element for each time slot)
Si=zeros(1,length(mu));
if (mean>0)
  Ei=(i+1) ./ mu;
  Si=1-expcdf(Ei,mean);
end
```

Appendix G: Users with limited patience – upper and lower bounds procedure

```
function [ShortageA_LB, SurplusA_LB,ShortageW_LB, SurplusW_LB, ShortageA_UB, SurplusA_UB,ShortageW_UB, SurplusW_UB] = Calc_Day_conservative(C,lambda, mu,disc,mean)
% Clculate lower and upper bounds for the expected number of shorage and
% surpless events; the epected waiting time of renteres and returners

% C - Station capacity
% lambda - Arrival rate of returners at each period (returners/ period)
% mu - Arrival of renters at each period (renters / period)
% lambda and mu - Row vectors
% disc - length in minutes of each period
% mean - expected patience  in minutes


mean = mean / disc;
M = 300; %The control parameter from the approximation formula
U=upper_trunc(lambda,mu,mean); %setting the left finite state
L=lower_trunc(lambda,mu,mean); %setting the right finite state
n=C+1+L+U; %total number of states including state 0

% Lower bounds  (one entry for each possible initial state X0)
ShortageA_LB = zeros(1,n); % Expected number of renteres abandoments
SurplusA_LB =  zeros(1,n); % Expected number of returners abandoments
ShortageW_LB = zeros(1,n); % Epected waiting time of renteres
SurplusW_LB =  zeros(1,n); % Expected waiting times of returners

% Upper bounds
ShortageA_UB = zeros(1,n);
SurplusA_UB =  zeros(1,n);
ShortageW_UB = zeros(1,n);
SurplusW_UB =  zeros(1,n);

P = eye(n);  % The matrix P(t)
T = length(lambda); % Total number time periods set by the lenght of lambda & mu


for t = 1: T  % Calculation of P(t) for any given t in [1,T] from the approximation formula
  R = create_R_new_vec(t,lambda,mu,C,U,L,mean); %Matrix that was defined in the approximation formula
  old_P = P; %this is the trasition probability matrix using BIC method
  P = P * calc_P(R, 1,M); %this is the trasition probability matrix using EIC method
  for j=1: L+1 %considering queue for bikes => states -L to 0
    B=beta_i(L+1-j,mean,lambda(t));
    Pj_LB = min(P(:,j),old_P(:,j)'); %Lower bound for shoratge - finding the minimal probability for each initial inventory level by comparimg the BIC and EIC matrices.
    Pj_UB = max(P(:,j),old_P(:,j)'); %Upper bound for sjortage- finding the maximal probability for each initial inventory level by comparimg the BIC and EIC matrices.
    ShortageA_LB = ShortageA_LB + Pj_LB * (mu(t)*(1-B)); %Shortage cost is calcualted from the first L+1 column in every P(t). L+1-j is the queue length
    ShortageW_LB = ShortageW_LB + Pj_LB *disc* (L+1-j);
    ShortageA_UB = ShortageA_UB + Pj_UB * (mu(t)*(1-B)); %Shortage cost is calcualted for every state from -L to 0 for every P(t). L+1-j is the queue length
    ShortageW_UB = ShortageW_UB + Pj_UB *disc* (L+1-j);
  end
  for j=L+1+C: n %considering queue for stalls => states C to C+U
    S=sigma_i(j-(L+1+C),mean,mu(t));
    Pj_LB = min(P(:,j),old_P(:,j)'); %Lower bound for surplus - finding the minimal probability for each initial inventory level by comparimg the BIC and EIC matrices.
    Pj_UB = max(P(:,j),old_P(:,j)'); %Upper bound for sjortage- finding the maximal probability for each initial inventory level by comparimg the BIC and EIC matrices.
    SurplusA_LB = SurplusA_LB + Pj_LB * lambda(t)*(1-S); %sURPLUS cost is calcualted from the Last column in every P(t)
    SurplusW_LB = SurplusW_LB + Pj_LB *disc*(j-(L+1+C));
    SurplusA_UB = SurplusA_UB + Pj_UB * lambda(t)*(1-S); %Surplus cost is calcualted for every state from C to C+U for every P(t). j-(L+1+C) is the queue length
    SurplusW_UB = SurplusW_UB + Pj_UB *disc*(j-(L+1+C));
  end

end

ShortageA_LB=ShortageA_LB(L+1: L+1+C); % Showing only feasible initial inventory levels
SurplusA_LB=SurplusA_LB(L+1: L+1+C);
ShortageW_LB=ShortageW_LB(L+1: L+1+C);
SurplusW_LB=SurplusW_LB(L+1: L+1+C);
ShortageA_UB=ShortageA_UB(L+1: L+1+C);
SurplusA_UB=SurplusA_UB(L+1: L+1+C);
ShortageW_UB=ShortageW_UB(L+1: L+1+C);
SurplusW_UB=SurplusW_UB(L+1: L+1+C);
```

Appendix H: Users with limited patience – results from the *Homogenous – symmetric* problem

Table 37. *Peaks – symmetric* problem - users with limited patience results

X_O	Total expected penalty cost - 15 min. disc.	Total expected penalty cost - 1 min. disc.	Difference (%)
0	150.1149	150.2102	0.0635
1	147.4636	147.5422	0.0533
2	144.7823	144.8995	0.0809
3	142.1346	142.2702	0.0954
4	139.4983	139.649	0.1080
5	136.8682	137.0323	0.1199
6	134.2414	134.4175	0.1312
7	131.6154	131.8023	0.1420
8	128.9881	129.1849	0.1526
9	126.3577	126.5634	0.1628
10	123.7224	123.9364	0.1730
11	121.0807	121.3021	0.1829
12	118.431	118.6593	0.1928
13	115.772	116.0064	0.2025
14	113.1023	113.3423	0.2122
15	110.4209	110.6658	0.2218
16	107.727	107.9762	0.2313
17	105.0201	105.2729	0.2407
18	102.3004	102.5561	0.2500
19	99.5688	99.8268	0.2591
20	96.8274	97.0869	0.2680
21	94.0796	94.3399	0.2767
22	91.3305	91.5908	0.2850
23	88.5874	88.8469	0.2929
24	85.8605	86.1184	0.3004
25	83.165	83.4207	0.3075
26	80.5287	80.7815	0.3139
27	78.0137	78.2631	0.3197
28	75.7808	76.0258	0.3233
29	74.2446	74.4726	0.3071
30	74.3062	74.5001	0.2609