2017 IEEE Conference on Control Technology and Applications (CCTA)
August 27-30, 2017. Kohala Coast, Hawai'i, USA

A Receding Horizon Routing Controller for Inventory Replenishment Agents in Bike Sharing Systems with Time-Varying Demand

Rebecca M. A. Swaszek† and Christos G. Cassandras‡

Abstract— **A Bike Sharing System (BSS) may be modeled as a graph with two node types: stations with finite bike inventory subject to time-varying demand and intersections to represent the underlying transportation network. Mobile agents (replenishment trucks) travel on the arcs of the graph to reset station inventories and make routing decisions at intersections. One-way rides create inventory imbalances across the system. Inventory control via rebalancing trucks has two main facets: selecting the number of bikes to load/unload between the truck and a station and routing decisions for the truck based on daily demand patterns. This paper focuses on the latter and introduces a Receding Horizon Controller which minimizes a user dissatisfaction metric defined as the expected number of users unable to rent or return a bike due to a station being empty or full, respectively. We model stations as M/M/1/K queues subject to time-varying birth and death rates based on the time of day. The controller proceeds in an event-driven manner and determines at each event the optimal routes over a finite planning horizon, with the control applied over a shorter action horizon. The proposed controller is applied to a simulated BSS with station and demand parameters extracted from the data sets of Hubway, the BSS in Boston MA.**

I. Introduction

Vehicle Sharing Systems, a form of mobility-on-demand systems, offer transportation for those whose vehicular needs are limited. Vehicles are parked in convenient locations and may be taken and returned by users at leisure. Bike Sharing Systems (BSS) are a subset of such systems and differ from their automotive peers in terms of fleet size, return location, and inventory management. Bike stations have a finite capacity of spaces, and inventory may be purposefully kept not full so users may rent or return a bike at will. Multiple bikes may be moved around the system on trucks, and BSS encourage users to not return to their origin station but to use the bikes to solve the "last mile" problem in commuting. These one-way rides cause inventory imbalances that lead to the need for multiple bikes to be moved from one station to another.

The rise in popularity of BSS in cities across the world has been followed by research focused on one or more of the following: truck routing problems, fleet size, station capacity, user dissatisfaction, and demand forecasting. In this paper we focus on the station inventory management side of a BSS– specifically the control of replenishment truck routes. Our objective is to minimize a user dissatisfaction metric– the expected number of users unable to rent (return) a bike at their desired station due to insufficient inventory of bikes (spaces). Other possible objectives in the related literature are to minimize the number of empty and full stations [1], to maximize the number of possible trips [2],[3],[4], and to keep station inventories between lower and upper bounds [5].

Inventory control via replenishment truck routing can be synthesized using methods such as linear programming [5],[4] or tree heuristics [2], and the route can be either static [5],[6] or dynamic and subject to change in response to stochastic inventory evolution [2],[7]. Modeling simplifications commonly include deterministic net demand [2] and constant demand rates [1].

A contribution of this paper is that the model introduced allows for *non-stationary* random demand processes. We adopt a Receding Horizon Control (RHC) approach, introduced in [8] and extended in [9], which operates in an event-driven fashion and is responsible for routing replenishment trucks as well as determining the number of replenishment bikes at a station. In this paper, we assume the latter is fixed and concentrate on the former. In particular, the RHC determines an optimal route over a given *planning horizon* on a graph topology of nodes (stations and intersection) and arcs (streets) which mimic the urban environment of BSS.

The paper is organized as follows: Section II introduces a discrete-event model, Section III details the Receding Horizon Controller, Section IV presents simulation results, and Section V concludes.

II. System Model

We consider a Bike Sharing System as a system of N nodes $\mathcal{N} = \{1, ..., N\}$ and A agents (trucks) $\mathcal{A} = \{1, ..., A\}$. Nodes are connected by arcs $(i, k) \in \mathcal{E}$ such that the system operates in the graph topology $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. Each arc (i, k) has an associated weight $W(i, k)$ which represents the time it takes for an agent to travel from node i to node k (note that this may include time spent by an agent at node i, e.g., waiting at a traffic light.) We extend the concept of weight to include arc segments (i, s) where $s \in \mathbb{R}^2$ may be any point on some arc $(i, k) \in \mathcal{E}$. Thus, $W(i, s) \leq W(i, k)$ for such points s. The set of nodes $\mathcal{N}$ is partitioned into two subsets: stations and intersections. We will refer to the set of station nodes as $\mathcal{S} = \{1, ..., S\}$, $\mathcal{S} \subset \mathcal{N}$. Station nodes have self loops of length $\mathcal{L}_i$ such that $W(i, i) = \mathcal{L}_i, i \in \{1, ..., S\}$ which represent the loading delay for replenishing a station.

* Supported in part by NSF under grants ECCS-1509084, CNS-1645681, and IIP-1430145, by AFOSR under grant FA9550-15-1-0471, by the MathWorks, by Bosch, and by the Clare Boothe Luce Foundation.

† Division of Systems Eng., Boston University, e-mail: swaszek@bu.edu
‡ Dept. of Electrical and Computer Eng. and Division of Systems Eng., Boston University, 8 St. Marys St., Boston, MA 02215, e-mail: cgc@bu.edu.

Every station i and agent j have finite integer inventory capacities C_i^S and C_j^A, respectively. Stations are subject to positive or negative demands for single units of inventory (i.e., spaces or bikes). The system runs for a finite period $[0,T]$ justified by the cyclical nature of BSS commuter demands. This finite period is divided into K intervals each of length $\mathcal{I}$ such that $T = K\mathcal{I}$. In interval $k \in \{1,...,K\}$ station i is subject to constant rates of *positive* demand (spaces for bikes) $\lambda_{i,k}$ and *negative* demand (bikes) $\mu_{i,k}$ [5]. The agents travel node-to-node to "rebalance" the stations from their own inventory with the goal of keeping station inventories out of the empty and full states.

State Space: Let $x_i(t) \in \{0,1,...,C_i^S\}$ be the inventory of station i, $y_j(t) \in \{0,1,...,C_j^A\}$ be the inventory of agent j, and $z_j(t) \in \mathcal{N}$ be the next (possibly current) node location of agent j at time t. Let $k(t) \in \{1,...,K\}$ be the interval that specifies the demand rates in effect at time t. Thus, the state of the system is $\mathscr{X}(t) = [x_1(t),...,x_S(t),y_1(t),...,y_A(t),z_1(t),...,z_A(t),k(t)]$.

Events: The system dynamics are event-driven such that a sample path over $[0,T]$ of a total of Q^T events may be described as a sequence of events $\mathbf{e} = \{e_1,...,e_{Q^T}\}$ and corresponding event times $\boldsymbol{\tau} = \{\tau_1,...,\tau_{Q^T}\}$. Only at these times τ_q may the state of the system change or control actions occur. We may write the state of the system in terms of time τ_q or event q: $\mathscr{X}(\tau_q) = \mathscr{X}_q$, $x_i(\tau_q) = x_{i,q}$, for all $i \in S$, $q \in \{1,\dots,Q^T\}$, etc. We define the set of all events E as the union of the uncontrollable and controllable events $E = E_U \cup E_C$. The following are the event types within E:

- δ_i^+ (δ_i^-) $\in E_U$ event: a positive demand event, i.e., demand for a space (respectively, negative demand event, i.e., demand for a bike) occurs at station $i \in S$.
- $\alpha_{i,j} \in E_C$ event: the arrival of agent j to node $i \in \mathcal{N}$.
- $\theta_{i,j} \in E_C$ event: agent j completes the inventory replenishment of station $i \in \mathcal{S}$.
- $\kappa_k \in E_U$ event: the start of the kth interval which prompts the change of demand rates $\lambda_{i,k}$ and $\mu_{i,k}$.

The distinction between events $\alpha_{i,j}$ and $\theta_{i,j}$ is needed to capture the fact that an agent arriving at a station may decide not to replenish it and, if it does, there is a finite amount of time $\mathcal{L}_i$ required for the replenishment.

Controls: Control actions are only possible after an arrival $\alpha_{i,j}$ or replenishment $\theta_{i,j}$ event.

- $v_{j,q}$, $j \in \mathcal{A}$ next node control: This is the next node selected for agent j to visit when located at $z_{j,q}$. This control is limited by the set of nodes which belong to the output set of node $z_{j,q}$ as defined by the graph; this set is denoted by $\mathcal{O}(z_{j,q})$. If $i \in \mathcal{S}$, then $i \in O(i)$, which represents the choice to replenish that station, which will be followed by an event $\theta_{i,j}$.
- $u_{i,j,q}$ inventory transfer control: This is the amount of inventory to transfer from agent j to station i and is only viable when $v_{j,q} = i$ following an $\alpha_{i,j}$ event, in which case it is followed by a $\theta_{i,j}$ replenishment completion event. This control is limited by the inventories and capacities of agent j and station i in that it is bounded from below by either the number of bikes at the station or spaces in the truck and bounded above by the number of bikes on the truck or spaces in the station:

$$u_{i,j,q} \in \{-\min\{x_{i,q\text{-}1}, C_j^A - y_{j,q\text{-}1}\},...,\min\{y_{j,q\text{-}1}, C_i^S - x_{i,q\text{-}1}\}\} \tag{1}$$

State Dynamics: The inventory evolution of station i depends on both the uncontrollable demand events δ_i^+ and δ_i^- and the controllable replenishment events $\theta_{i,j}$:

$$x_{i,q} = \begin{cases} \min\{x_{i,q-1}+1, C_i^S\} & e_q = \delta_i^+ \\ \max\{x_{i,q-1}-1, 0\} & e_q = \delta_i^- \\ x_{i,q-1} + u_{i,j,q} & e_q = \theta_{i,j}, \quad j \in \mathcal{A} \\ x_{i,q-1} & \text{otherwise} \end{cases} \tag{2}$$

Note that the $\min$ operation in (2) prevents the station inventory from exceeding capacity in the case that a positive demand occurs at a full station; likewise, the $\max$ operation prevents station inventory from falling below 0 in the case that a negative demand event occurs at an empty station.

The inventory evolution of agent j depends solely upon the controllable replenishment events $\theta_{i,j}$:

$$y_{j,q} = \begin{cases} y_{j,q-1} - u_{i,j,q} & e_q = \theta_{i,j},\ i \in \mathcal{S} \\ y_{i,q-1} & \text{otherwise} \end{cases} \tag{3}$$

Note that the transfer amount $u_{i,j,q}$, must stay within the upper and lower bounds as defined in (1), which keeps the station inventory in (2) feasible: $x_{i,q} \in \{0,1,\dots,C_i^S\}$, as well as agent inventory in (3) feasible: $y_{j,q} \in \{0,1,...,C_j^A\}$.

Agent j's next node location z_j remains the same until the agent arrives at that node following an event $\alpha_{z_j,j}$. The controller decides upon a route for agent j and the next node location changes to be the first node s' on that route:

$$v_{j,q} = \begin{cases} s' & e_q = \alpha_{i,j} \\ v_{j,q-1} & \text{otherwise} \end{cases} \tag{4}$$

Finally, at time instants $t = k\mathcal{I}$, $k \in \{1,\dots,K\}$ the interval k changes upon the occurrence of a κ_k event, thus causing the demand rates $\lambda_{i,k}$ and $\mu_{i,k}$ to change for every station $i \in \mathcal{S}$: $k_q = k_{q-1} + 1$.

Objective Function: The agents' purpose is to minimize the number of events in which a demand cannot be sated by station inventory across the entire system over $[0,T]$. We assign penalties p and h for each event when a negative and positive demand cannot be satisfied, respectively [6]. Let $\boldsymbol{\rho}_i^{[0,T]} = \{\rho_{i,1},...,\rho_{i,B_i^{[0,T]}}\}$ be the event times of all δ_i^- negative demand events at station i and $\boldsymbol{\sigma}_i^{[0,T]} = \{\sigma_{i,1},...,\sigma_{i,D_i^{[0,T]}}\}$ be the event times of all δ_i^+ positive demand events over $[0,T]$ for a total of $B_i^{[0,T]}$ positive and $D_i^{[0,T]}$ negative demands. We seek to minimize the expected cost over $[0,T]$ to capture the importance of every user's demand. Let $\mathscr{X}_0$ be the initial state of the system and $\mathbf{1}[\cdot]$ be the indicator function. $\mathbf{u}_q = \{u_{1,q},...,u_{A,q}\}$ enumerates the replenish amounts while $\mathbf{v}_q = \{v_{1,q},...,v_{A,q}\}$ describes the agent's routes. The optimization problem is:

$$J(\mathscr{X}_0) = \min_{\mathbf{u}_q, \mathbf{v}_q} E\Bigg[\sum_{i=1}^{S}\Bigg(\sum_{q=1}^{B_i^{[0,T]}} \big(\mathbf{1}[x_i(\rho_{i,q})=0]p + \sum_{q=1}^{D_i^{[0,T]}} \mathbf{1}[x_i(\sigma_{i,q})=C_i^S]h\big)\Bigg)\Bigg] \quad (5)$$

III. Receding Horizon Control

We motivate the RHC by discussing the complexity of the optimality equation corresponding to (5) and by exploiting the fact that penalties are incurred only when uncontrollable events occur. We define $V_i(\mathscr{X}, \mathbf{v}_q, \mathbf{u}_q)$ as the cost incurred when the qth event takes place at station i given a state $\mathscr{X}$:

$$V_i(\mathscr{X}, \mathbf{v}_q, \mathbf{u}_q) = \begin{cases} p & e_q = \delta_i^- \quad , \quad x_{i,q-1} = 0 \\ h & e_q = \delta_i^+ \quad , \quad x_{i,q-1} = C_i^S \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

The solution of the following optimality equation over $q = 1, \ldots, Q^T$ yields the optimal expected cost $J_q(\mathscr{X})$ when the state is $\mathscr{X}$ after the qth event:

$$J_q(\mathscr{X}) = \min_{\mathbf{u}_q, \mathbf{v}_q} E\Bigg[\sum_{i \,\in\, \mathcal{S}} V_i(\mathscr{X}, \mathbf{v}_q, \mathbf{u}_q) + J_{q+1}(\mathscr{X}')\Bigg] \quad (7)$$

where $V_i(\mathscr{X}, \mathbf{v}_q, \mathbf{u}_q)$ is the cost incurred when event e_q takes place and $J_{q+1}(\mathscr{X}')$ is the cost-to-go after the next event e_{q+1} when the state has changed to $\mathscr{X}'$. Evaluation of the cost-to-go $J_{q+1}(\mathscr{X}')$ depends not only on the controllable events $\alpha_{i,j}$ and $\theta_{i,j}$, but it is further complicated by the time-varying demand rates $\lambda_{i,k}$ and $\mu_{i,k}$ making such evaluation particularly complex under time-varying demand processes. This complexity is what normally motivates the use of RHC which restricts the selection of decisions to be optimal only over a finite *planning horizon*. These decisions are executed over a shorter *action horizon*, at which point a new optimization problem is solved [8],[9].

The complexity due to the time-varying demand processes is mitigated through a RHC based on the following observation. Note that in (6) penalties are only incurred upon uncontrollable demand events δ_i^- and δ_i^+, while controllable events incur no penalties. Since we can only apply controls $\mathbf{v}_q, \mathbf{u}_q$ at events $\alpha_{i,j}$ and $\theta_{i,j}$, by properly updating our RHC we can evaluate the expected cost in between these events and then select a control that minimizes it. Thus, we introduce a RHC which takes advantage of this behavior to determine optimal controls over an appropriately selected planning horizon $H-$ applied until the next control event when the RHC is re-invoked. Our RHC is event driven, similar to the approach in [10] for elevator systems featuring a similar graph topology. Our analysis is based on the following assumptions:

A1: All events are observable.

A2: Inter-demand times are exponentially distributed.

A3: Agents have an infinite inventory capacity.

Assumption ***A1*** is justified by the fact that modern BSS inventories are available in real time via apps and online [11].

Assumption ***A2***, supported by empirical data [12] and accepted in the literature [5],[6], allows us to treat station inventories as finite birth-death chains. Each station is an independent M/M/1/C_i^S queue subject to birth rate $\lambda_{i,k}$ and death rate $\mu_{i,k}$ [12],[5],[6]. In reality, if users are between stations they may choose the station with the higher inventory that they seek (bikes or spaces) [13], so that the demand rates may in fact be inventory dependent.

Using assumption ***A2*** we may evaluate *transient* probability distributions for the station states, which in turn allows us to make decisions that take into account the time-varying character of the demand processes. Let us define the following useful integral where $\pi_{i,n}(x_i(t), k, s)$ is the transient probability that the inventory of station i is $n \in \{0, ..., C_i^S\}$ at time $t+s$ when the initial state is $x_i(t)$ and $(k-1)\mathcal{I} < t < k\mathcal{I}$:

$$\Gamma_i(x_i(t), k, \Delta) = \int_0^{\Delta} \big[\pi_{i,0}(x_i(t), k, s)\mu_{i,k}p + \pi_{i,C_i^S}(x_i(t), k, s)\lambda_{i,k}h\big]ds$$
$$(k-1)\mathcal{I} < t < t+\Delta < k\mathcal{I} \quad (8)$$

Observe that this is the expected cost incurred at station i over an interval $[t, t+\Delta]$ fully contained within the kth interval $[(k-1)\mathcal{I}, k\mathcal{I}]$ of fixed demand rates. This cost depends on the initial inventory $x_i(t)$ and interval index $k(t)$ (we will omit the dependence on t for simplicity).

Let us now extend the expected cost evaluation in (8) to cases where $[t, t+\Delta]$ spans two intervals when demand rates $\lambda_{i,k}$ and $\mu_{i,k}$ change due to event κ_k at time $t = k\mathcal{I}$. In particular, let Δ be such that

$$(k-1)\mathcal{I} \le t \le k\mathcal{I} < t+\Delta \le (k+1)\mathcal{I} \quad (9)$$

This allows us to extend the evaluation of the expected cost incurred at station i into future times after the demand rates have changed, as illustrated in Fig. 1, i.e., $[t, t+\Delta]$ is split into $[t, k\mathcal{I}]$ and $(k\mathcal{I}, t+\Delta]$ corresponding to the times of constant demand rates in intervals k and $k+1$, respectively.

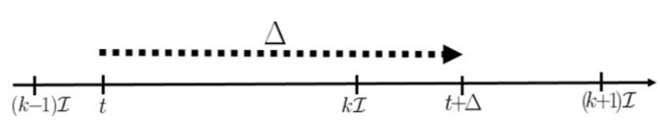

Fig. 1. The case when $[t, t+\Delta]$ spans intervals k and $k+1$ and the demand rates $\lambda_{i,k}$ and $\mu_{i,k}$ change at time $k\mathcal{I}$.

Let $B_i^{[t,t+\Delta]}$ and $D_i^{[t,t+\Delta]}$ be the sequences of negative and positive demand events in $[t, t+\Delta]$, respectively. Using (8) and conditioning on the state of the station at time $t = k\mathcal{I}$, the expected cost accrued over $[t, t+\Delta]$ is given by:

$$
\begin{aligned}
&\mathcal{J}_i(x_i(t), k, \Delta) \\
&= E\Bigg[\sum_{q=1}^{B_i^{[t,t+\Delta]}} \mathbf{1}[x_i(\rho_{i,q}) = 0]p + \sum_{q=1}^{D_i^{[t,t+\Delta]}} \mathbf{1}[x_i(\sigma_{i,q}) = C_i^S]h\Bigg] \\
&= \Gamma_i(x_i(t), k, k\mathcal{I} - t) \qquad + \\
&\sum_{n=0}^{C_i^S} \pi_{i,n}(x_i(t), k, k\mathcal{I} - t)\Gamma_i(n, k+1, t+\Delta - k\mathcal{I}) \quad (10)
\end{aligned}
$$

The first term of (10) refers to the time $(t, k\mathcal{I}]$ spent in the kth interval when the initial inventory $x_i(t)$ is known. The second term refers to the time $[k\mathcal{I}, t+\Delta]$ spent in the $(k+1)$th interval when the probability that the initial inventory is $x_i(k\mathcal{I}) = n$ is given by $\pi_{i,n}(x_i(t), k, k\mathcal{I} - t)$.

Over the interval $[t, t+\Delta]$, some stations will accrue more expected costs due to their initial inventory and demand rates as shown in Fig. 2. Some stations are therefore more critical than others and should be considered for an agent to visit so as to rebalance the inventory to mitigate future expected costs.

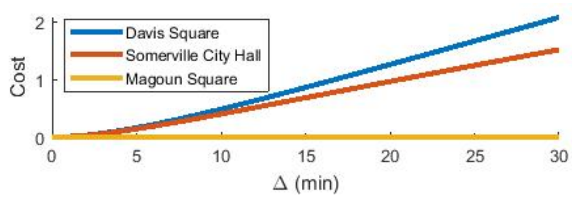

Fig. 2. Expected cost for stations in the Boston, MA BSS Hubway with demand rates μ_i and λ_i extracted from 2015 data weekdays 8-9 AM, penalties p and h assigned to 1, and initial inventories $\mathbf{x}(0) = \mathbf{C}^S - 1$ [11]. Note that the expected cost for Magoun Square is so low as to be negligible – therefore, not a good candidate to rebalance inventory when other stations such as Davis Square show vastly higher costs.

Assumption ***A3*** is made so as to let us focus on the routing facet of the BSS inventory management problem. If agents have infinitely many bikes or spaces, the inventory transfer control in (1) is no longer constrained by agent capacity and becomes: $u_{i,j,q} \in \{-x_{i,q}, ..., C_i^S - x_{i,q}\}$ and (3) is no longer necessary. Instead of deciding how much inventory to transfer, we predefine an inventory level to which the agent will rebalance the station to; in particular, we select the halfway point: $\bar{x}_i = \lfloor \frac{C_i^S}{2} \rfloor$. Effectively, the control $u_{i,j,q} = \lfloor \frac{C_i^S}{2} \rfloor - x_{i,q}$. The state dynamics in (2) change to account for this predefined inventory level:

$$
x_{i,q} = \begin{cases} \min\{x_{i,q}+1, C_i^N\} & e_q = \delta_i^+ \\ \max\{x_{i,q}-1, 0\} & e_q = \delta_i^- \\ \bar{x}_i & e_q = \theta_{i,j}, \quad j \in \mathcal{A} \\ x_{i,q} & \text{otherwise} \end{cases} \tag{11}
$$

Agent Routing Control: A route $R(l, s)$ starting at node $l \in \mathcal{N}$ and ending at any point s belonging to some arc ($s \notin \mathcal{N}$ in general) is defined as a sequence $\{l, r_1, \ldots, r_M, s\}$ where $r_m \in \mathcal{N}$, $m = 1, \ldots, M$ and s is reachable from l over the given graph $(\mathcal{N}, \mathcal{E})$. If $r_m = r_{m+1}$, this indicates that the agent applies control $v_{j,q} = r_m$ when the qth event is $\alpha_{r_m,j}$, hence it takes the self-loop arc and replenishes station r_m. The *distance* of such a route, denoted by $d(l, s)$ is given by: $d(l,s) = W(l, r_1) + \sum_{m=1}^{M-1} W(r_m, r_{m+1}) + W(r_M, s)$.

Let us now consider any route $R(l, s)$ such that $d(l, s) = H$ and define a set $\Omega_i(H)$ of *horizon points* ω_i as follows: $\Omega_l(H) = \{\omega_l : d(l, \omega_l) = H\}$. In simple terms, the set $\Omega_l(H)$ contains all points $s \in \mathbb{R}^2$ reachable from $l \in \mathcal{N}$ in H time units through the graph $(\mathcal{N}, \mathcal{E})$ as illustrated in Fig. 3. Thus, we consider routes $R(l, s)$ with $\omega_l \in \Omega_l(H)$ and denote the set of such routes by $\mathcal{R}_l^H$.

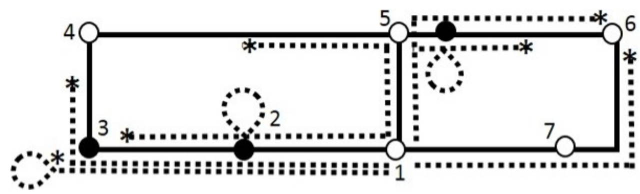

Fig. 3. All possible horizon points when an agent is located at intersection 1 where $*$ mark the horizon points and self loops represent replenishments. Intersections and stations are marked as white and black circles, respectively.

Next, assume that agent j is located at node $l \in \mathcal{N}$ at some time $t \in [0, T]$. We associate routes in $\mathcal{R}_l^H$ to agent j by writing $R_j^H \in \mathcal{R}_l^H$. We now consider events $\theta_{i,j}$ which may occur over the interval $[t, t+H]$ when agent j replenishes some station $i \in \mathcal{S}_l^H$. Let the time of such an event be $t + \xi_i^j$ (at most one event per station to avoid cyclical routes). Note that when the replenishment event $\theta_{i,j}$ occurs, the inventory of station i changes to a new value $x_i(t + \xi_i^j) = \bar{x}_i$ in accordance with (11). Finally, note that the value of ξ_i^j is simply given by: $\xi_i^j = d(l, i)$, the time it takes agent j to travel from node l to station $i \in S$ and replenish it.

Let us now select the planning horizon H to span two intervals as in (9) with $\Delta = H$. Note that this extends beyond the next event time when a control is feasible, so that the controller can effectively "see" the savings effects the possible controls will have on the system. We define a route vector $\mathbf{R}^H = [R_1^H, \ldots, R_A^H]$ as a combination of possible agent routes $R_j \in \mathcal{R}_l^H$. Let $\mathcal{S}_{\mathbf{R}}^H$ be the set (possibly empty) of stations to replenish included in any route vector $\mathbf{R}^H$. For $i \notin \mathcal{S}_{\mathbf{R}}^H$, the only relevant event is the demand rate change event occurring at $k\mathcal{I}$. For $i \in \mathcal{S}_{\mathbf{R}}^H$, the relevant events are the occurrence of the replenishment event(s) $\theta_{i,j}$, $j \in \mathcal{A}$ at time $t + \xi_i^j$ which may occur before or after the demand rate change event.

Next, we evaluate the expected cost $\mathcal{C}_i(\mathbf{R}^H)$ for station i over $[t, t+H]$. Let us, for simplicity, first start with the single-agent case $A = 1$. We can then omit the superscript j in the replenishment time ξ_i^j. There are three possible cases which all follow from (10).

Case 1: $i \notin \mathcal{S}_{\mathbf{R}}^H$ i.e., no replenishment events over $[t, t+H]$.

$$
\mathcal{C}_i(\mathbf{R}^H) = \mathcal{J}_i(x_i(t), k, H) \tag{12}
$$

Case 2: $i \in \mathcal{S}_{\mathbf{R}}^H$, $t + \xi_i \leq k\mathcal{I}$, i.e. a replenishment event occurs before the demand rates change.

$$\mathcal{C}_i(\mathbf{R}^H) = \Gamma_i(x_i(t), k, \xi_i) + \mathcal{J}_i(\bar{x}_i, k, H - \xi_i) \tag{13}$$

Case 3: $i \in \mathcal{S}^H_{\mathbf{R}}$, $t + \xi_i > k\mathcal{I}$, i.e., a replenishment event occurs after the demand rates change.

$$\mathcal{C}_i(\mathbf{R}^H) = \mathcal{J}_i(x_i(t), k, \xi_i) + \Gamma_i(\bar{x}_i, k+1, H - \xi_i) \tag{14}$$

For a multi-agent system ($A > 1$), the number of cases expands to include the number of replenishment events at a station before and after the demand rates change. Let $\{t + \xi_{i,1}, ..., t + \xi_{i,m}, k\mathcal{I}, t + \xi_{i,m+1}, ..., t + \xi_{i,M}\}$ be the sequence of relevant events for station i over $[t, t + H]$ with a total of M replenishment events by various agents and the demand rate change event at $t = k\mathcal{I}$. Let $t + \xi_{i,m}$ be the time of the last replenishment event before the demand rates change. Note that we omit again the subscript j in ξ^j_i associated with these events because, under ***A3***, all replenishments are equivalent in terms of inventory outcomes: $x_i(t + \xi^j_{i,r}) = \bar{x}_i$, $r \in \{1, ..., M\}$, $j \in \mathcal{A}$. There are two possible cases:

Case 1: $i \notin \mathcal{S}^H_{\mathbf{R}}$, i.e., no replenishment events over $[t, t + H]$. This case is identical to (12).

Case 2: $i \in \mathcal{S}^H_{\mathbf{R}}$, $t + \xi_{i,r} < k\mathcal{I}$, $r \in \{1, ..., m\}$, i.e., the first m replenishment events occur before the demand rates change. It follows from (10) that

$$\begin{aligned}
&\mathcal{C}_i(\mathbf{R}^H) = \Gamma_i(x_i(t), k, \xi_{i,1}) + \\
&\sum_{r=2}^{m} \Big[\Gamma_i(\bar{x}_i, k, \xi_{i,r} - \xi_{i,r-1}) \Big] + \mathcal{J}_i(\bar{x}_i, k, \xi_{i,m+1} - \xi_{i,m}) + \\
&\sum_{r=m+1}^{M-1} \Big[\Gamma_i(\bar{x}_i, k+1, \xi_{i,r+1} - \xi_{i,r}) \Big] + \Gamma_i(\bar{x}_i, k+1, H - \xi_{i,M})
\end{aligned} \tag{15}$$

Note that the special cases where $m = M$ or $m = 0$ are included in (15). If $m = M$, then all replenishment events occur before the demand rates change, so that the third term above vanishes and the sum in the second term applies with $m = M$ and with $\xi_{i,m+1}$ in $\mathcal{J}_i(\bar{x}_i, k, \xi_{i,m+1} - \xi_{i,m})$ replaced by H. Similarly, if $m = 0$, then all replenishment events occur after the demand rates change, so that the first and second term above vanish to be replaced by $\mathcal{J}_i(x_i(t), k, \xi_{i,1})$.

When the RHC is invoked at time t at which point an agent is located at some node $l \in \mathcal{N}$, the controller solves an optimization problem seeking to determine routes $R^H_j \in \mathcal{R}^H_l$ for all agents $j = 1, \ldots, A$ to form a route vector $\mathbf{R}^H$ minimizing the total cost based on (12) and (15). The RHC optimization problem is:

$$\mathbf{R}^{H*} = \arg \min_{R^H_j \in \mathcal{R}^H_l, j=1,...,A} \sum_{i=1}^{S} \mathcal{C}_i(\mathbf{R}^H) \tag{16}$$

The solution of this problem is straightforward to obtain since it entails comparing a finite number of values $\mathcal{C}_i(\mathbf{R}^H)$ over all possible route vectors $\mathbf{R}^H$. Once an optimal route for each agent j is determined, we set the next node control $v_j(t) = r^*_{j,1}$ as the first node on the optimal route $R^*_j = \{l, r^*_{j,1}, \ldots, r^*_{jR}, \omega_l\}$. The RHC is re-invoked when any agent reaches its immediate next node $v_j(t)$.

IV. Simulation Results using Hubway Data

Hubway, the BSS of Boston, MA, publishes data on its usage each quarter which includes the time, duration, origin, and destination stations for each trip as well as the latitude, longitude, and capacity of all stations. We extracted the mean demand rates for bikes and spaces for each hour of each day of the 2015 season. We conglomerated all weekdays from April to October and used the mean demand rates from 6 AM to 10 PM to simulate a normal day. Fig. 4 shows an example of a typical demand pattern.

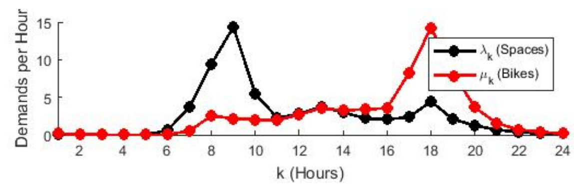

Fig. 4. Mean demand rates at the Boston Public Library station show high demand for spaces (bikes) in the morning (afternoon)– typical for a commercial area.

We simulated in MATLAB the Somerville neighborhood of Hubway with 22 nodes: 11 stations, 11 intersections, and arcs representing main thoroughfares. A BSS often places stations at intersections for user convenience, so effectively stations are special intersections. Fig. 5 shows the graph topology simulated. The arc times are calculated as the distance according to Google Maps divided by an average speed of 10 miles/hour to account for city traffic.

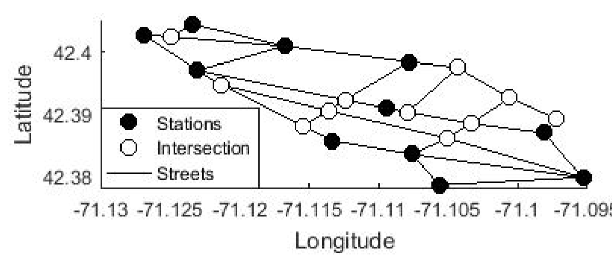

Fig. 5. The Somerville neighborhood system.

The simulation begins with each station's initial inventory $\mathbf{x}(0) = \lfloor \frac{\mathbf{C}^S}{4} \rfloor$. We set $\mathcal{I}$ to 1 hour, K to 16 intervals, penalties p and h both to 1, and simulate many constant lengths of planning horizon H. The case in which the planning horizon is contained within a single interval has a simpler cost structure which relies upon sums of $\Gamma(\cdot)$ terms as in (8) and (10). Fig. 6 shows for the one-agent case how the length of the planning horizon H affects the average performance of the RHC for 100 simulated sample paths of the full 16-hour day with various loading delays $\mathcal{L}$ (assumed fixed over all stations).

In order to better assess the performance of our RHC, we compare it to a "greedy" controller and a Traveling Salesman Problem (TSP) shortest path controller. The greedy controller routes the agents based on system inventory alone without regard for demand rates, capacity, or node decision points. This controller commits the agents to a shortest-path route

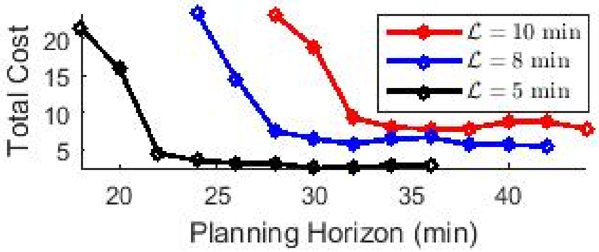

Fig. 6. The average total cost for 100 sample paths for the one-agent case with various loading delays $\mathcal{L}$ shows that the performance of the RHC improves when the constant planning horizon H is extended. However, past some threshold the performance remains about the same.

to the "neediest" station– the station whose inventory at time t is most extreme in terms of near empty or full:

$$v_j(t) = \arg\min_{i \in S} \left\{ \min\{x_i(t), C_i^S - x_i(t)\} \right\} \tag{17}$$

Table I shows results from 100 simulated sample paths for the one and two-agent cases under the RHC, greedy and TSP controllers for different loading delay parameters $\mathcal{L}$. Each RHC case was run for many different planning horizons H (as in Fig. 6), but we only display the results of the planning horizon H^* which resulted in the lowest total cost on average. Clearly, on average the RHC performs better than the greedy controller given a sufficiently long planning horizon. Moreover, as expected, the RHC also drastically outperforms the TSP controller. Note that the

TABLE I

COMPARING CONTROLLERS: AVERAGE TOTAL COSTS

		RHC			Greedy		TSP	
	$\mathscr{L}$ (min)	H^* (min)	AVG	SD	AVG	SD	AVG	SD
A=1	5	30	2.5	2.4	5.7	4.4	15.2	7.0
	8	42	5.4	4.6	9.3	5.8	26.7	8.1
	10	36	7.7	5.0	12.8	6.5	31.3	10.7
A=2	5	24	0.3	0.6	1.0	1.5	4.2	2.9

averages are quite low and the standard deviations rather high. For example, in the one-agent case with $\mathcal{L} = 5$ min under the RHC, most sample paths resulted in very low total costs with some high cost outliers: 42% of sample paths resulted in total costs ≤ 1.0 while 8% resulted in costs ≥ 6.0. This is echoed in the greedy controller to a lesser extent: 20% of sample paths resulted in total costs ≤ 1.0 and 8% resulted in costs ≥ 13.0.

V. CONCLUSION

We introduced an event-driven Receding Horizon Controller (RHC) for the inventory management of a Bike Sharing System (BSS) subject to time-varying demand rates which considers current inventory, route travel times, and current and future demand rates to find a sequence of optimal routes over a planning horizon. Routes are executed only until any agent reaches its first node in the route, at which point the RHC is re-invoked.

The BSS is modeled as a node and arc graph topology that mimics the streets and intersections of cities. We model stations and intersections as points where routing decisions are made, as these represent real decision points. We include simulation results for the one and two-agent cases with parameters extracted from the datasets of Hubway, the BSS in Boston, MA in order to illustrate how the RHC performs compared to two baseline simple alternatives.

Ongoing and future work includes finding the optimal inventory transfer control $u_{i,j}$, relaxing ***A3***, i.e., finite agent capacity, and simulations on a larger topology. We are investigating potential instabilities due to cyclic routes the means to avoid them as well exploring incentivizing users to locally rebalance by slightly altering their desired stations.

REFERENCES

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

[2] J. Pfrommer, J. Warrington, G. Schildbach, and M. Morari, "Dynamic vehicle redistribution and online price incentives in shared mobility systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1567–1578, 2014.

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

[4] N. Jian, D. Freund, H. M. Wiberg, and S. G. Henderson, "Simulation optimization for a large-scale bike-sharing system," in *Proceedings of the 2016 Winter Simulation Conference*. IEEE Press, 2016, pp. 602–613.

[5] J. Schuijbroek, R. Hampshire, and W.-J. van Hoeve, "Inventory rebalancing and vehicle routing in bike sharing systems," *European Journal of Operational Research*, vol. 257, no. 3, pp. 992–1004, 2017.

[6] T. Raviv and O. Kolka, "Optimal inventory management of a bike-sharing station," *IIE Transactions*, vol. 45, no. 10, pp. 1077–1093, 2013.

[7] I.-L. Wang and C.-W. Wang, "Analyzing bike repositioning strategies based on simulations for public bike sharing systems: Simulating bike repositioning strategies for bike sharing systems," in *Advanced Applied Informatics (IIAIAAI), 2013 IIAI International Conference on*. IEEE, 2013, pp. 306–311.

[8] W. Li and C. G. Cassandras, "A cooperative receding horizon controller for multivehicle uncertain environments," *IEEE Transactions on Automatic Control*, vol. 51, no. 2, pp. 242–257, 2006.

[9] Y. Khazaeni and C. G. Cassandras, "A new event-driven cooperative receding horizon controller for multi-agent systems in uncertain environments," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 2770–2775.

[10] K. Wesselowski and C. Cassandras, "The elevator dispatching problem: hybrid system modeling and receding horizon control," *IFAC Proceedings Volumes*, vol. 39, no. 5, pp. 136–141, 2006.

[11] Hubway, "Hubway: metro-Boston's bike share program," 2016. [Online]. Available: https://www.thehubway.com/

[12] N. Gast, G. Massonnet, D. Reijsbergen, and M. Tribastone, "Probabilistic forecasts of bike-sharing systems for journey planning," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015, pp. 703–712.

[13] A. Schlote, B. Chen, and R. Shorten, "On closed-loop bicycle availability prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, 2015.