C2SMART
CONNECTED CITIES WITH
SMART TRANSPORTATION

Non-myopic Relocation of Idle Mobility-on-Demand Vehicles as a Dynamic Location-Allocation-Queueing Problem

Hamid R. Sayarshad, Joseph Y.J. Chow

This is the author's version of a work that has been accepted for publication in Transportation Research Part E: Logistics and Transportation Review, Volume 106, October 2017, pages 60-77. The final version can be found at doi.org/10.1016/j.tre.2017.08.003.

Non-myopic relocation of idle mobility-on-demand vehicles as a dynamic location-allocation-queueing problem

Hamid R. Sayarshad*
School of Civil and Environmental Engineering, Cornell University, Ithaca, NY 14853, USA
*Corresponding author's email: hs628@cornell.edu

Joseph Y.J. Chow
Department of Civil & Urban Engineering, New York University, New York, NY, USA

Abstract

Operation of on-demand services like taxis, dynamic ridesharing services, or vehicle sharing depends significantly on the positioning of idle vehicles to anticipate future demand and operational states. A new queueing-based formulation is proposed for the problem of relocating idle vehicles in an on-demand mobility service. The approach serves as a decision support tool for future studies in urban transport informatics and design of types of urban mobility like ridesharing, and smart taxis. A Lagrangian Decomposition heuristic is developed and compared with a relaxed lower bound solution. Using New York taxicab data, the non-myopic allocation problem reduces the cost in 27%, and 25%, as compared to the myopic case.

Keywords: p-median problem; preposition of idle vehicles; relocation costs, Lagrangian decomposition

Accepted for publication in Transportation Research Part E: Logistics and Transportation Review

1. Introduction

When dealing with the dynamic operations of a demand-responsive transit or mobility-on-demand (MOD) service, the location of an idle vehicle has a strong influence on the quality of service. Vehicles must be able to reach customers within a reasonable time. As an example, taxicab drivers in New York City (NYC) may spend 39% of their total mileage cruising for passengers (Schaller Consulting, 2006; Zhang et al., 2015). Similar issues exist with car sharing, where idle vehicles may need rebalancing to other locations in order to better serve the dynamic demand. The vehicle sharing service, Car2Go, had to abandon the market in San Diego due to high costs of rebalancing the fleet to fit the demand (Garrick, 2016).

Because of these steep costs in MOD services, and with the emerging availability of large scale real-time data (see Sayarshad, 2015; Sayarshad and Chow, 2016), a number of studies have examined ways to use real-time data to improve decision support. These include strategies to preposition idle vehicles in anticipation of new passenger arrivals and future locations of vehicles, or to rebalance vehicles in the system. Recent examples include Yuan et al. (2011), Powell et al. (2011), and Li et al. (2011) for taxis; and Nourinejad and Roorda (2014), Raviv et al. (2013), Zhang et al. (2016), Contardo et al. (2012), and Waserhole and Jost (2016) for vehicle sharing systems. A review of various dynamic ridesharing systems is available from Agatz et al. (2012). A strategy consists of an online relocation policy that directs vehicles, whether it is rebalancing or prepositioning idle vehicles, based on real-time demand data.

Recent forays into developing online relocation policies acknowledge the benefits of using information to look ahead, such as Thomas and White's (2004) modeling of vehicle waiting strategies (i.e. stay or move on) with look ahead, Chow and Regan's (2011a) chance-constraint relocation model with a rolling horizon, and Chow and Sayarshad's (2015) adaptive facility location model. These studies differ from the ones mentioned earlier as they feature look-ahead, and are classified as "non-myopic" online relocation policies. In other words, network information is combined with statistical learning to anticipate such future states as demand or vehicle availability.

Decisions that look ahead into the future based on stochastic information are called Markov decision processes (MDP), and are typically employed to find optimal non-myopic online policies. An MDP under discrete time intervals is modeled using a Bellman equation (Powell, 2011) as shown in Eq (1).

$$V_t(R_t) = \min_{x_t}(C_t(R_t, x_t) + \gamma E[V_{t+1}(R_{t+1})|(R_t, x_t)]) \tag{1}$$

where $\mathrm{V_t}$ is the value of the optimal dynamic policy, $\mathrm{C_t}$ is the immediate payoff of the decision $\mathrm{x_t}$ under state $\mathrm{R_t}$ (which is also typically driven by information on exogenous stochastic variables, and varies in size based on the underlying distribution of the variable(s)), E is an expectation, and γ is a discount factor. As noted in the network-based studies, anticipation of future states in a network context is highly intractable without significant simplifications in the state or network structure, due to the curse of dimensionality when trying to evaluate the conditional expectation $E[V_{t+1}(R_{t+1})|(R_t, x_t)]$ in Eq (1).

In the infinite horizon version of the problem, the optimal value function is a fixed point that is time-invariant. Researchers recently considered the use of queue delay as a simple approximation of this fixed point. This approximation stems naturally from the historical use of queueing models to capture steady state service delays. For example, Daganzo (1978) modeled a many-to-many static dial-a-ride service as a spatial queueing system, and Waserhole and Jost

(2016) analyzed a vehicle sharing system with a steady state queueing network. In the context of dynamic decisions, however, the steady state queueing delay is used to capture the conditional expected cost of the steady state due to a current decision instead.

Despite being an approximation, the queue-based policy has been shown in several simulation studies to be more effective than myopic policies in several applications: dynamic dispatch of MOD services (Hyytiä et al., 2012), dynamic pricing of MOD services (Sayarshad and Chow, 2015), and dynamic vehicle rebalancing of MOD services (Zhang and Pavone, 2016). In particular, the latter study is most similar to our current study in dealing with relocating vehicles using queues to approximate the conditional expected costs. However, Zhang and Pavone (2016) use a multi-server queueing network as the underlying model, which only considers the cost of matching with a customer in the same zone, as opposed to facility location problems, which consider spatial coverage costs of serving nearby zones.

We propose an online, non-myopic idle vehicle relocation policy based on using queue delay as an approximation of the conditional expected cost. Our policy considers spatial coverage effects by relying on an underlying server relocation model from facility location theory to have queueing components similar to how Marianov and ReVelle (1994) modified static facility location models to include queueing.

The significance of the contribution is most apparent in the area of shared autonomous vehicle fleet operations. Zhang and Pavone (2016), as well as other researchers (Brownell and Kornhauser, 2014; Fagnant and Kockelman, 2015; Liang et al., 2016; Mendes et al., 2017), have sought to combine ride-sharing with autonomous vehicle fleets. The National Highway Transportation Safety Administration (2013) defined five levels of AV functionality ranging from no automation (level 0) to full automation without driver controls (level 5). The technology to enable automation, particularly at levels 3 to 5, is extremely sophisticated and needs to apply high-performance computational hardware, state-of-the-art online models, decision-making algorithms, and real-time information. Our methodology could be of great use to automated vehicle (AV) and self-driving cars within on-demand mobility systems.

The literature is reviewed in Section 2; the dynamic policy is formulated in Section 3; a solution algorithm based on Lagrangian decomposition is proposed and tested in Section 4; and the effectiveness of the proposed non-myopic policy is validated against a myopic relocation policy using a simulation test derived from real NYC taxi data in Section 5.

2. Literature review

2.1. Idle vehicle rebalancing operations

Due to the availability of taxi and shared vehicle/bike data, much of the recent progress in positioning or rebalancing idle service vehicles has been applied to those services (compared to truck deliveries, dial a ride services, and emergency vehicle operations). In most taxi systems, customer arrivals are highly irregular and known to the operator only shortly before serving the customer. Yuan et al. (2011) proposed a probabilistic model based on a parking place detection algorithm to recommend locations to taxi drivers and people where they can easily find vacant taxis. Powell et al. (2011) introduced an approach based on historical data (GPS dataset) to direct taxi drivers in order to reduce the number of cruising miles (taxi without passenger) while increasing the number of live miles (taxi with passenger). This approach was tested on 600 taxis in Shanghai. The proposed framework was based on historical GPS data to score the potential profitability of locations given the current location and time of a taxi driver. Li et al. (2011)

proposed a time-location model with GPS data for passenger-finding strategies. This approach was tested on 5,350 taxis in China.

Yang et al. (2010) modeled urban taxi services in a network context to reduce the number of cruising miles while increasing the number of live miles by defining the relationship between customer and taxi waiting times. Another approach focused on customer queueing at taxi stands and taxis switching between serving stands and looking for passenger (see Cheng et al. 2009). Phithakkitnukoon et al. (2011) developed a predictive model for the number of vacant taxis in a given area based on time and weather conditions. The approach was tested on 150 taxis in Lisbon, Portugal. Gan et al. (2010) considered travel time variability on driver route choices in Shanghai taxi service. Fernandez et al. (2006) were concerned with the economics of taxi services under various types of regulation such as entry restriction and price control. Li et al. (2009) and Sirisoma et al. (2009) introduced a variety of issues from demand versus supply to pricing issues. Implementation of cyber technologies to assist cruising taxis in finding passengers was proposed by Hou et al. (2013).

Similar studies have been conducted for shared vehicle rebalancing, either in a static (Sayarshad et al., 2012; Raviv et al., 2013) or dynamic (Contardo et al., 2012; Nourinejad and Roorda, 2014; Zhang et al., 2016) context. However, these studies do not explicitly consider future actions as Markov decision processes. These studies also do not consider the reality of spatial coverage: that unmet demand may relocate to another node or be served by a vehicle positioned at a different node, which the class of facility location problems addresses.

2.2. Server relocation

Positioning of idle vehicles is a type of resource allocation problem categorized within the field of location science, where the location decision impacts the level of service that a vehicle provides to demand in neighboring nodes. Optimization problems in this area are called facility location problems, and like in routing, they can fall into different variations and subclasses depending on the objective and constraints desired. For example, the p-median problem is one type of location problem in which the objective is to minimize the average distance between all demand nodes and their closest facilities. Due to the large number of studies in location theory, only some relevant studies in relocation are highlighted. Interested readers can refer to Brotcorne et al. (2003) and Owen and Daskin (1998) for an overview.

Some of the earliest relocation research dealt with emergency services in urban fires (Kolesar and Walker, 1974). Berman and Odoni (1982) demonstrated the computational challenges of locating mobile servers in a stochastic network. Solutions to cope with this computational challenge involved three approaches. The first, exemplified by Berman et al. (1985), sought to understand the problem using stylized analytical expressions in simplified network settings. The second approach, exemplified by Gendreau et al. (2006), treated the dynamic decision as a near-myopic two-stage reoptimization problem. Other similar applications include Nair and Miller-Hooks (2009), where trade-offs are made between immediate relocation with higher transportation costs against no relocation and subsequent changes in coverage cost. The third approach is based on approximation of the conditional expected future performance over more than one future period. One example is that of Chow and Regan (2011a), who designed a chance constraint approach under a multi-period rolling horizon for the server relocation problem. None of these latter two approaches, however, extended the dynamic decision beyond one *a posteriori* decision stage. Chow and Sayarshad (2015) showed how to quantify the value of a dynamic relocation policy that allowed for timing, which has been shown by Chow and Regan (2011b) to always be a better solution than a myopic policy. Such a policy, however, appears currently unattainable for practical problems.

One scalable approximation approach for Markov decision processes is to approximate the future conditional expected value function with a queue delay expression. Hyytiä et al. (2012), Sayarshad and Chow (2015), and Zhang and Pavone (2016) demonstrated the effectiveness of this approximation for dynamic routing and assignment. Queueing has been used in location models before, but as a way of measuring the additional queue delay that may exist for certain services such as emergency vehicle fleets that may be busy when a call comes in, or electric charging infrastructure (e.g. Jung et al., 2014). Berman et al. (1987) presented a generalization of the p-median queue, concerned with waiting cost and service time. Batta et al. (1988) and Batta (1989) proposed a one-server model for the optimal location. These problems were solved by greedy heuristic algorithms to find the location of the server.

Marianov and ReVelle (1994, 1996) proposed queueing maximal availability location problems. These studies introduced several probabilistic maximal covering location-allocation models with linear constrained waiting time for queue length to consider service congestion. Marianov and Serra (1998) extended the maximum covering by taking the fact that the number of request services behaves according to some distribution function. Marianov and Serra (2002) modified these models, seeking a probabilistic location-allocation model to cover all the population. Berman and Drezner (2007) proposed a multi-server allocation model that accounted for queues at the server nodes, assuming that the demand is assigned to the closest node. Jung et al. (2014) proposed a simulation-based optimization approach using Berman and Drezner's (2007) approach in the upper level to locate electric charging stations that accounted for queueing delay and allocating to taxi tours instead of demand nodes.

2.3. Solution algorithms

In terms of solution procedures, authors have considered exact, heuristic, and metaheuristic algorithms to solve facility location problems. Mladenovíc et al. (2007) provide a classification of constructive heuristics, local search and mathematical programming algorithms for solving the p-median problem. Pasandideh and Niaki (2012) solved a multi-objective facility location problem based on queuing theory by genetic algorithms. They used a desirability function by defining a geometric mean to minimize waiting time, and idle probability at an open facility. Ant colony optimization (Levanova and Loresh, 2004), simulated annealing (Levanova and Loresh, 2004), and scatter search (García-López et al., 2003) have also been proposed in recent years. Marianov and Serra (2002) proposed a heuristic for solving the maximal covering, multiple-server model. The heuristic is based on the Heuristic Concentration method (Rosing et al., 1997). In addition, some heuristic algorithms have been proposed in order to find the optima (Cooper, 1964; Resende and Werneck, 2007).

The branch and bound (B&B) algorithm and subsequent variations are exact solution methods to location problems formulated as integer programming problems. However, the increase in number of variables and constraints makes a problem especially troublesome to solve when probabilistic constraints have the form of capacity constraints. These constraints tend to dramatically increase the branching in the B&B algorithm (ReVelle, 1993). One strategy to deal with large-scale location problems is Lagrangian relaxation (LR), which consists of relaxing some of the constraints to obtain a lower bound. Typically, in this approach the capacity constraints are relaxed and penalized in the objective function (Geoffrion, 1974).

The application of Lagrangian relaxation method with subgradient optimization with exact solutions can be found in Neebe (1978), Boccia et al. (2008), Ceselli (2003), and Ceselli and Righini (2005). They proposed three exact solutions: branch and bound, branch and cut, and branch and price methods with Lagrangian relaxation and subgradient optimization to solve the p-median problem. The Lagrangian relaxation method has also been used with heuristics to solve capacitated problems. Pirkul and Schilling (1991) and Current and Storbeck (1988) proposed such heuristics for capacitated maximal covering location models, and Davis and Ray (1969), and Cornuejols et al. (1991) introduced algorithms for capacitated plant location problems. For a review of LR methods and applications, see Guignard (2003).

3. Idle vehicle prepositioning problem

3.1. System description and model formulation

We consider the following generic data-driven ("smart") MOD system as shown in **Figure 1** from Sayarshad (2015). The framework is shown in **Figure 1**, which lists the various technologies and data sources needed to ensure a viable "smart" transit system that can be used in real-world operators such as New York taxis and Toronto Transit Commission (TTC). For example, communications equipment between the vehicle and the dispatching center is needed. Intelligent Vehicle-Highways Systems (IVHS) is an application of information and communications technologies (ICT) in order to better control the flow of vehicles. Electronic Data Interchange (EDI) is the electronic transfer from computer to computer to increase speed of communication and control all aspects using message data. Customers can send travel requests to dispatch centers via the communication equipment. Mobile device communication systems are one example of a technology capable of providing this information.

This system can serve as any dynamic mobility system where trip demand enters the system dynamically (see Djavadian and Chow 2016 for a discussion of the complexities of evaluating such systems). In this system, there are three key functions: dispatch/routing/assignment of vehicles to passengers, prediction of customer arrivals, and positioning or rebalancing of idle vehicles. We focus on the last function.

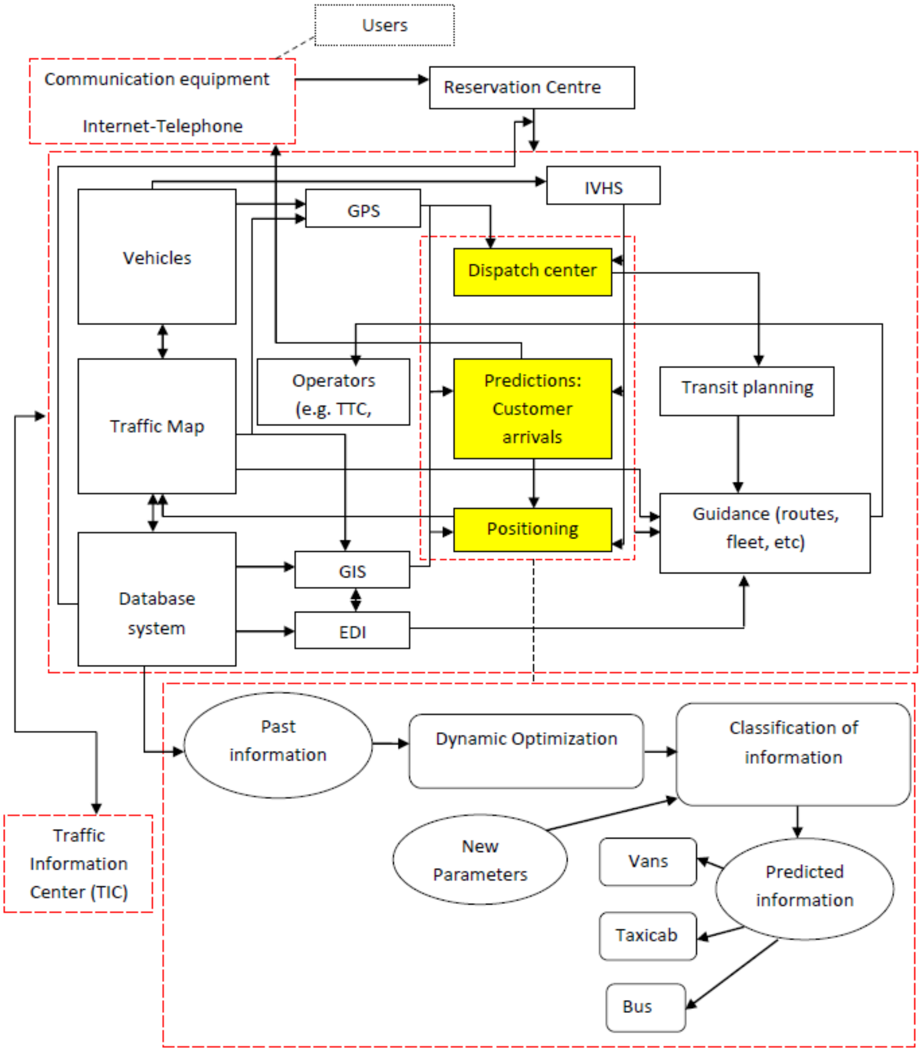
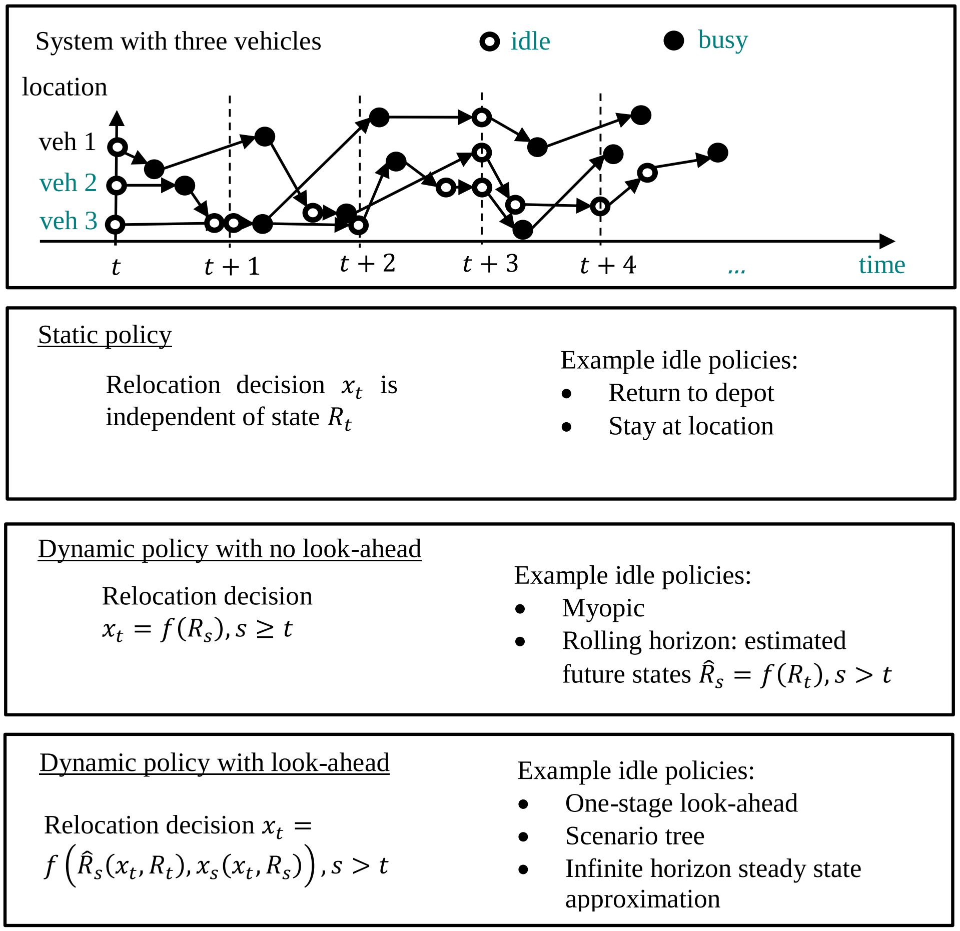


Fig. 1. **General framework for a "smart" mobility-on-demand service (source: Sayarshad, 2015).**

The problem setting is defined generically as follows (and can be adapted for taxi prepositioning or rebalancing shared vehicles):

- A centralized online operator policy queries the set of all idle vehicles and current demand levels at the start of every time interval T
- The policy returns a set of relocation recommendations to all the idle vehicles
- Requests arrive according to a Poisson process and service times are assumed independent and exponentially distributed
- In the context of the dynamic policy described by Eq (1), the $C_t(R_t, x_t)$ is the immediate relocation cost, while $E[V_{t+1}(R_{t+1})|(R_t, x_t)]$ is the expected future cost (including relocation and service cost) conditional on the new locations of the idle vehicles

We illustrate this problem setting using **Figure 2**.



Static policy

Relocation decision x_t is independent of state R_t

Example idle policies:

- Return to depot
- Stay at location

Dynamic policy with no look-ahead

Relocation decision $x_t = f(R_s), s \geq t$

Example idle policies:

- Myopic
- Rolling horizon: estimated future states $\hat{R}_s = f(R_t), s > t$

Dynamic policy with look-ahead

Relocation decision $x_t = f\left(\hat{R}_s(x_t, R_t), x_s(x_t, R_s)\right), s > t$

Example idle policies:

- One-stage look-ahead
- Scenario tree
- Infinite horizon steady state approximation

Fig. 2. Illustration of different idle vehicle relocation strategies using a 3-vehicle example.

The first panel in **Figure 2** describes the problem setting. Vehicles are either idle or busy at any time in the system. In our system, decisions are made for all idle vehicles at the start of each time interval ($t, t+1, t+2, \ldots$). A newly idle vehicle at some interval in time stays in location until the new interval is reached (e.g. veh 3 between $t+2$ and $t+3$) or until a new request is matched (veh 2 between t and $t+1$). Multiple vehicles may be posted in the same zone (e.g. veh 2 and veh 3 at $t+1$). A vehicle may serve multiple customers over multiple locations before becoming idle (e.g. veh 1 between t and $t+2$). Only idle vehicles are considered for relocation at the start of each time interval. For example, at t the three vehicles are all allocated (veh 2 and veh 3 assigned to the same location, while veh 1 assigned to relocate elsewhere), at $t+1$ veh 2 and veh 3 are assigned to the same location (likely because there is high demand noted there), at $t+2$ only veh 3 is idle and assigned to relocate elsewhere. Once the relocation policy is executed at the start of a time interval, the system lets the vehicles run their course (via other dispatch/routing algorithms as needed) until the start of the next time interval.

Different policies may be used for relocation, as highlighted by the bottom three panels in **Figure 2**. A static policy only considers decisions that are not updated based on new information (e.g. Nair and Miller-Hooks, 2012; Sayarshad et al., 2012; Raviv et al., 2013; Schuijbroek et al., 2017). A dynamic policy without look-ahead (e.g. Chow and Regan, 2011a; Contardo et al., 2012; Nourinejad and Roorda, 2014; Zhang et al., 2016) takes into account the updated information, and may even use that information to make predictions of future state conditions (these are rolling horizon problems). However, these policies do not explicitly account for the dependency of future states on future decisions, which are in turn dependent on current state and decision. These are the ones modeled as Markov decision processes using Bellman equations as shown in Eq (1). As mentioned in the introduction, MDP policies can be solved in multiple ways. One- or two-stage look ahead considers all possible state changes combined with all possible relocation trajectories of vehicles. This problem is difficult to solve for real instances due to curse of dimensionality. Finite horizon approximate dynamic programming (with either value or function approximation) may be conducted by simulating scenario trees that branch off for different possible decisions and states. A third approach is to look at this problem from an infinite horizon perspective; that for a given state trend, we forecast the steady state condition and find a solution to Eq (1) that is stable. Several studies (e.g. Hyytiä et al. (2012), Sayarshad and Chow (2015), and Zhang and Pavone (2016)) have employed this last method for modeling MDPs by modeling the whole system of vehicles as a multi-server spatial queue such that the queue delay approximates the $E[V(R,x)]$ of the fixed point of Eq (1).

We follow the same approach as these studies, where the policy is to locate idle vehicles at the start of each time interval T based on current state of passenger requests arrivals, to minimize the cost of relocation and immediate coverage $C_t(R_t,x_t)$ as well as the fixed point value $E[V(R,x)]$ of this spatial queueing system.

Parameters:

N: the set of nodes in system

λ_{iT}: arrival rate at node $i \in N$, which may vary for each time interval T (the T index is dropped for convenience since only the arrival rate at the current time interval T is used when the model is run)

μ_j: average rate at which vehicles at node $j \in N$ match with and serve a customer, where the service time is assumed to follow an exponential distribution.

t_{ij}: travel time of matching a vehicle at node j to a customer at node i

r_{ij}: the cost of relocating an idle vehicle from node i to node j

θ: a scalar conversion of the relocation cost to the value of improved deployment time

C_j: maximum possible number of vehicles that are cruising or standing at node j

B: fleet of idle vehicles

y_j^T: total number of idle vehicles located at node j at the start of time interval T

Decision Variables: (note that the index T is dropped for convenience since each variable is solved anew at each time interval, independent of the variables in other time intervals)

X_{ij}: customer arrivals in node i matched to vehicles in node j if set to 1

Y_{jm}: the m^{th} vehicle from node j to serve customers if set to 1 (note that multiple vehicles can be assigned to any given node by notation m)

W_{ij}: flow of idle vehicles relocating from node i to node j

S_i: dummy variable for the surplus of vehicles based on current idle vehicle configurations y_j^T

D_j: dummy variable for the demand of vehicles based on current idle vehicle configurations y_j^T

Contrary to the queues in other facility location problems, the queueing in this model is used to measure the future opportunities of a vehicle that is positioned at a particular node. In that context, the service rate is dependent on demand patterns. For example, a taxi that locates in an area in which it is matched to many short trips nearby may have a very high service rate (and effective queue capacity). Conversely, a car sharing service that relocates a car to an area whose users tend to reserve the vehicle for very long periods may have a very low service rate.

P1: Non-linear relocation problem

$$Z_1 = min \sum_{i \in N} \sum_{j \in N} \lambda_i t_{ij} X_{ij} + \theta \sum_{i \in N} \sum_{j \in N} r_{ij} W_{ij} + \sum_{j \in N} \frac{\sum_i \lambda_i X_{ij}}{\mu_j \sum_m Y_{jm} - \sum_i \lambda_i X_{ij}} \tag{2}$$

Subject to:

$$\sum_{j \in N} X_{ij} = 1, \qquad \forall i \in N \tag{3}$$

$$Y_{jm} \leq Y_{j,m-1}, \qquad \forall j, m = 2,3, \dots, C_j \tag{4}$$

$$X_{ij} \leq Y_{j1}, \qquad \forall i, j \in N \tag{5}$$

$$\sum_{j \in N} \sum_{m=1}^{C_j} Y_{jm} = B \tag{6}$$

$$\sum_{j \in N} W_{ij} = S_i\,, \qquad \forall i \in N \tag{7}$$

$$\sum_{i \in N} W_{ij} = D_j\,, \qquad \forall j \in N \tag{8}$$

$$-D_j - y_j^T + \sum_{m=1}^{C_j} Y_{jm} \leq 0, \qquad \forall j \in N \tag{9}$$

$$-S_j + y_j^T - \sum_{m=1}^{C_j} Y_{jm} \leq 0, \qquad \forall j \in N \tag{10}$$

$$X_{ij}, Y_{jm} \in \{0,1\} \tag{11}$$

$$D_j, S_j, W_{ij} \geq 0 \tag{12}$$

Objective function (2) is to minimize the travel time between demand points and server locations, the sum of relocation costs to get vehicles from current locations to the new locations, and the conditional expected cost approximated as an average M/M/m queue delay. Customer demands at each node i appear according to a Poisson process with intensity λ_i. Constraint (3) allocates each customer demand node to one and only one idle vehicle. Constraint (4) ensures the $(m-1)^{th}$ idle vehicle is allocated before allocating the m^{th} idle vehicle. Constraint (5) forces the allocation of a demand node only to an idle vehicle. Constraint (6) sets the number of available idle vehicles. Constraints (7) – (10) form the transportation problem constraints used to determine the relocation flows. Constraints (7) and (8) assign the differences in vehicle locations to supply and demand at each node using dummy variables. By updating y_j^T to Y_{jm} in each time interval, we are able to realize the total number of idle vehicles located at each node to serve multiple demand nodes. Constraints (9) and (10) are flow conservation constraints. The remaining constraints are non-negativity and binary constraints.

P1 is a non-linear optimization problem, since the average M/M/m queue delay is calculated by a nonlinear $\sum_{\mathrm{j\in N}} \frac{\sum_{\mathrm{i}} \lambda_{\mathrm{i}} \mathrm{X}_{\mathrm{ij}}}{\mu_{\mathrm{j}} \sum_{\mathrm{m}} \mathrm{Y}_{\mathrm{jm}} - \sum_{\mathrm{i}} \lambda_{\mathrm{i}} \mathrm{X}_{\mathrm{ij}}}$. Following Marianov and Serra (2002), we convert it to **P2**, which is an equivalent linear optimization. The contribution lies in how the constraint is designed to be dependent on the decision variables. By creating a reliability constraint that changes marginally as the number of servers in a node changes, the constraint values dynamically adjust to the decision variables of server locations. For example, a node with 2 servers will be bound by one reliability constraint, but if the model considers locating a third server there, it would encounter a different reliability constraint. The nonlinear objective is thus equivalently considered in an implicit manner. Furthermore, the degree of equivalency is controllable through the choice of parameters so that a solution can focus more on reducing queue delay or on total cost of the original **P1** objective.

To be more precise, Marianov & Serra (2002) showed the delay objective can be achieved with an exogenously constructed utilization rate constraint as shown in Eq (13) – (15). Eq (13) is the static representation of the reliability constraint.

$$\sum_{i\in N} \lambda_i X_{ij} \leq \mu_j \rho_\alpha, \qquad \forall j \in N \tag{13}$$

where a ρ can be determined for any reliability level α as shown in Eq (14) for a given number of users in queue b and number of servers m. The constraint (13) is defined as: “at his/her arrival to the node, every passenger will wait on a line with no more than b other customers with a probability of at least α”.

$$\sum_{k=0}^{m-1} ((m-k)m!\, m^b/k!)\,(1/\rho^{m+b+1-k}) \geq 1/(1-\alpha) \tag{14}$$

Eq (13) is in turn modified into a marginal constraint that changes depending on whether the m^{th} server is located there Y_{jm}, as shown in Eq (15). Since Y_{jm} is bound to X_{ij}, which is in turn in the objective, the constraint is able to capture a delay objective implicitly.

$$\sum_{i\in N} \lambda_i X_{ij} \leq \mu_j \left[Y_{j1}\rho_{\alpha j1} + \sum_{m=2}^{C_j} Y_{jm}\big(\rho_{\alpha jm} - \rho_{\alpha j,m-1}\big) \right], \qquad \forall j \in N \tag{15}$$

Note that the values of Eq (14) are completely independent of the decision variables, so fixed values of $\rho_{\alpha jm}$ are computed for every possible α, j, m prior to running the model. As an example, when $b = 0$ and $\alpha = 0.95$, then solving for $\rho_{\alpha jm}$ for $m = 1{,}2$ we would get $\rho_{0.95,j1} = 0.2236$ and $\rho_{0.95,j2} = 0.6417$. When $\alpha = 0.85$, we get $\rho_{0.85,j1} = 0.3873$.

The objective of **P1** is replaced with a linear objective Eq (16) and the additional constraint (15) to create the equivalent **P2**.

P2: Equivalent Mixed Integer Linear Programming Problem

Additional inputs:

$\rho_{\alpha jm}$: utilization ratio that corresponds to confidence level α for node j with m servers

$$Z_2 = min \sum_{i \in N} \sum_{j \in N} \lambda_i t_{ij} X_{ij} + \theta \sum_{i \in N} \sum_{j \in N} r_{ij} W_{ij} \tag{16}$$

Subject to:

Constraints (3 to 12, 15)

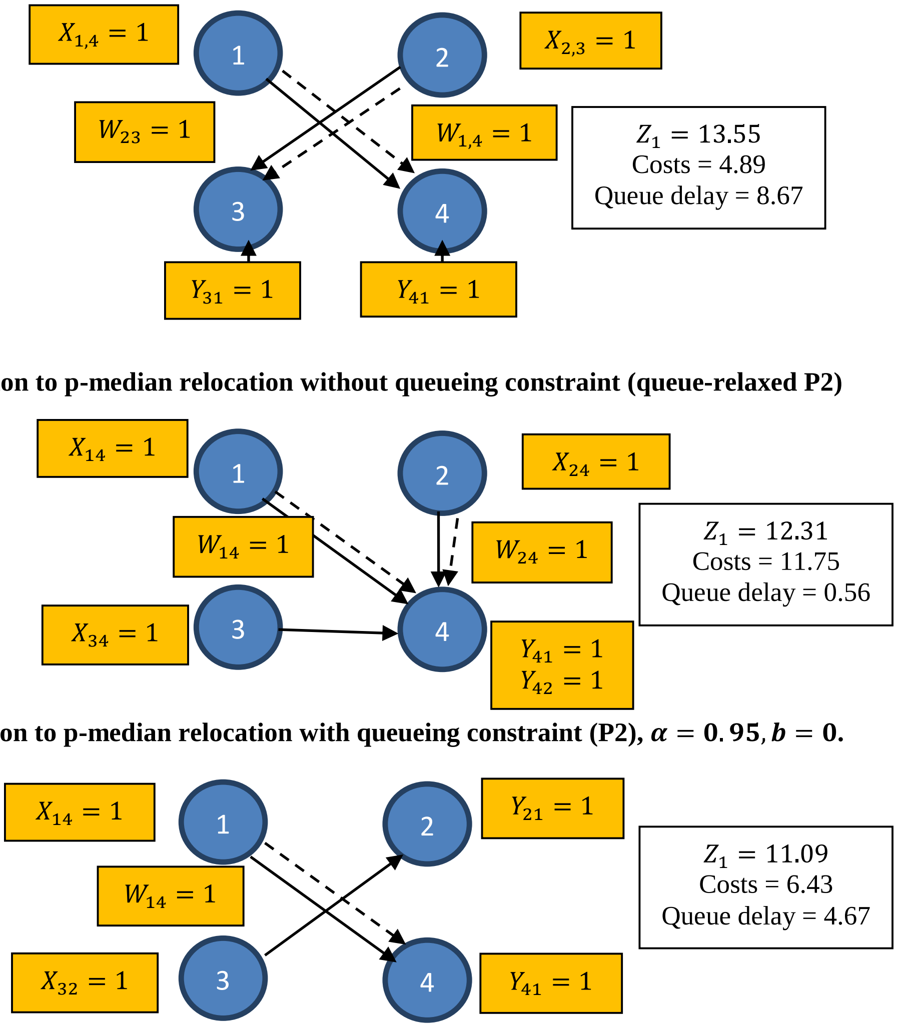
3.2. Illustration of P2

For a better understanding of how **P2** works, we illustrate the model difference with one without a queueing constraint. For this illustration, we use the following parameters with four nodes: $B = 2, \theta = 0.2$, and initial locations $y^0 = (1{,}1{,}0{,}0)$. Symmetric relocation costs and travel times are assumed: $r_{21} = 0.043$, $r_{31} = 0.032$, $r_{41} = 0.049$, $r_{32} = 0.030$, $r_{42} = 0.073$, $r_{43} = 0.077$, $t_{21} = 0.950$, $t_{31} = 1.265$, $t_{41} = 0.638$, $t_{32} = 0.773$, $t_{42} = 1.473$, and $t_{43} = 0.950$. The demand is based on the arrival rates $\lambda_1 = 4,\ \lambda_2 = 3, \lambda_3 = 5, \lambda_4 = 6)$. These parameters would be sufficient to solve a p-median relocation problem. The solution is shown in **Figure 3(a)**. The optimal solution when queueing is ignored is to move the server in node 1 to node 4, and the server from node 2 to node 3. But is that the right model in a dynamic context? Suppose node 4 is shown to present the most opportunity for located servers to subsequently serve customers, due to a combination of customer origin-destination patterns (e.g. node 4 may have many short trips that start and end near node 4), while the other three nodes present less opportunity. We model this difference in future opportunity with service rates $\mu_1 = 8, \mu_2 = 10, \mu_3 = 9,\ \mu_4 = 25$ in a queueing system. In this case, taking into account the conditional future expected costs of the solution, the objective value of Z1 is 13.55 (4.89 from just the coverage and relocation cost, and 8.67 from the conditional expected cost of relocating to nodes 3 and 4 modeled as queue delay).

Solving **P2** not only accounts for the queue delay used to approximate the future cost, it is also able to control the degree of inclusion. For example, consider parameters $\alpha = 0.95$ and $b = 0$. In this case, the operator wants to ensure that vehicle availability is a top priority through the high significance level and focus on probability of remaining idle. The solution to this problem is shown in **Figure 3(b)**, which recommends relocating both servers to node 4. While the queue delay is minimal, the trade-off is overall higher coverage costs. The objective value of Z1 is 12.31 (11.75 from coverage and relocation, and 0.56 from queue delay).

Now consider what happens when we change our parameters to $\alpha = 0.30$ and $b = 1$. The focus on delay is relaxed to queues of one or less. The result is shown in **Figure 3(c)**, which recommends assigning the server at node 1 to node 4 while leaving the server at node 2 in place.

The model recognizes that node 3 has a low service rate (which is interpreted as having a high opportunity cost of being relocated there) and not worth the relocation despite the reduced coverage cost. The objective value of Z1 is 11.09 (6.43 from coverage and relocation, and 4.67 from queue delay). Both **P2** models can improve upon the naïve model without queue consideration, and while the second **P2** model reduces the overall cost, the queue delay is much higher than the first **P2** model.



a. Solution to p-median relocation without queueing constraint (queue-relaxed P2)

b. Solution to p-median relocation with queueing constraint (P2), $\boldsymbol{\alpha = 0.95, b = 0}$.

c. Solution to p-median relocation with queueing constraint (P2), $\boldsymbol{\alpha = 0.30, b = 1}$.

Fig. 3. Comparison of solutions from illustrative example.

3.3. LP-relaxed lower bound

Because of the computational burden of large number of binary variables when the number of servers becomes large, we also consider a LP-relaxed formulation as a lower bound for comparison. In this case, we can recognize that the constraint (15) is essentially a server-by-server piecewise-linear representation of the queueing delay as a constraint. Since the queue delay is convex with a cost minimization objective under a budget constraint, Bradley et al. (1977) has shown that it is possible to just relax the integer constraints. We take our original formulation (P2) and just set $Y_{j\mathrm{m}}$ to be continuous between 0 and 1. We then introduce a new integer variable Y_j as a decision variable that represents the number of vehicles in zone j. So, we create a sum variable $\sum_{m=1}^{C_j} Y_{jm} = Y_j \ \ \forall j \in N$ and require the sum to be integer. This could significantly simplify the model: by greatly reducing the number of binary decision variables. However, in some cases it may be possible that an infeasible solution is obtained.

3.4. Algorithm design requirements

If we let N_1 be the set of demand nodes, N_2 be the set of server nodes, and C be the uniform server capacity for each node, then the number of design variables is: $(2N_1N_2 \ + N_2C + \ N_1 \ + \ N_1)$. The total number of constraints is equal to$(3(\ N_1 + N_2) \ + N_2C)$. The total number of binary variables is equal to$(N_1N_2 \ + N_2C)$. For example, if the number of both demand nodes and server nodes is equal to 30 and the uniform server capacity for each node is 10, then the number of design variables is equal to 2160. The total number of constraints is equal to 480 and the total number of binary variables is equal to 1200. Note that the input parameters, $i \in N_1$, $j \in N_2$, and $m \in C_j$, govern the size of the problem with respect to both the design variables and the constraints.

The increase in the number of variables and constraints is especially inconvenient because the probabilistic constraint has the form of a capacity constraint and capacitated problems are especially difficult to solve. We propose a heuristic based on Lagrangian decomposition in Section 4 to address this challenge. Its premise is to separate the model into two sub-problems: 1) a p-median problem in the space of the X_{ij} and Y_{jm}, and 2) a transportation problem in the space W_{ij} that is the relocation cost.

4. Proposed Lagrangian Decomposition heuristic

4.1. Algorithm derivation

We propose a Lagrangian relaxation method with decomposition of our problem into two sub-problems to find feasible bounds for large problems. **P2** is decomposed into a p-median problem in the space of the X_{ij} and Y_{jm} called **P3**, where $i, j \in N$ and $m \leq C_j$, and a relocation problem in the space W_{ij} called **P4**.

P3 (IP):

$$Z_3 = \min \sum_{i \in N} \sum_{j \in N} \lambda_i t_{ij} X_{ij}$$

Subject to:

$$(3), (4), (5), (6), (11), and \ (15)$$

In the case of **P3**, we relax constraint (3) to obtain a Lagrangian with multiplier vectors L_i, as shown here.

$$\hat{Z}_3 = \min \sum_{i \in N} \sum_{j \in N} (\lambda_i t_{ij} - L_i) X_{ij} + \sum_{i \in N} L_i$$

Subject to:

$$(4), (5), (6), (11) and \ (15)$$

$$X_{ij}, Y_{jm} \in \{0,1\}$$

We derive an upper bound for the variable Y_{jm} to use in determining the upper bound primal solution. If we sum the set of constraints in Eq (15) together we obtain:

$$\sum_{i\in N} \lambda_i \sum_{j\in N} X_{ij} \leq \sum_{j\in N} \mu_j \left[Y_{j1}\rho_{\alpha j1} + \sum_{m=2}^{C_j} Y_{jm}\left(\rho_{\alpha jm} - \rho_{\alpha j,m-1}\right) \right]$$

From Eq (3), the $\sum_{j\in N} X_{ij}$ term drops out of the left hand side, leading to Eq (17).

$$\sum_{i\in N} \lambda_i \leq \sum_{j\in N} \mu_j Y_{j1}\rho_{\alpha j1} + \sum_{j\in N}\sum_{m=2} \mu_j Y_{jm}\left(\rho_{\alpha jm} - \rho_{\alpha j,m-1}\right) \qquad (17)$$

We use this relationship to help determine a set of feasible Y_{jm}. As for **P4**, we have the following.

P4 (LP):

$$Z_4 = \min \left[\hat{Z}_3\left(Y_{jm}^*, X_{ij}^*, L_i^*\right) + \theta \sum_{i\in N}\sum_{j\in N} r_{ij} W_{ij} \right] \qquad (18)$$

Subject to:

$$(7), (8), (9), (10)$$

$$D_j, S_j, W_{ij} \geq 0$$

The proposed algorithm is shown below.

Lagrangian Decomposition (LD) heuristic for solving P2

Step 0. Initiate iteration counter $n = 0$, given a maximum number of iterations n_{max}. Calculate the matrix $H = \left(h_{ij}\right) = \lambda_i t_{ij}$, $\gamma = \gamma_{ij}$, Lagrangian multipliers $L_i^n = 0$, $\forall i \in N$ so that $\gamma_{ij} = \min(0, h_{ij} - L_i)$, $\gamma_j = \sum_{i\in N} \gamma_{ij}$ $\forall j \in N$, $UB = \infty$ and $LB = 0$. Initialize $X_{ij}^n = 0$ $\forall i, j \in N$.

Step 1. Determine a feasible Y_{jm}^{n+1} that satisfies constraints (17), (4), (5), and (6) based on X_{ij}^n.

Step 2. Update a set of candidate variables $\bar{X}_{ij}^k$ $\forall k \in \{0, N\}$:

2A. Initiate $\bar{X}_{ij}^k = 0$ $\forall i, j, k, X_{ij}^{n+1} = 0$ $\forall i, j$.

2B. Set $X_{ij}^{n+1} = 1$ if $Y_{jm}^{n+1} = 1$ and $\gamma_{ij} < 0$. Set $\bar{X}_{ij}^k = 1$ $\forall k$ if $Y_{jm}^{n+1} = 1$ and $(i, j) = \mathrm{argmin}_{\mathrm{i,j}}\left(h_{ij}\right)$ $\forall i, j, m$.

2C. For $k = 1$ to $|N|$, find (i', k) where $(i', k) = \mathrm{argmin}_{\mathrm{i',k}}(h_{i'k})$: $\{(i' \neq i)\}$. Set $\bar{X}_{i'k}^k = 1$ and $\bar{X}_{ik}^k = 0$ if $Y_{jm}^n = 1$.

2D. Remove infeasible solutions $\bar{X}_{ij}^k$ that do not satisfy constraint (15) with Y_{jm}^{n+1}.

2E. Compute $\underline{Z}_D$ and $\bar{Z}_k$ for the remaining feasible solutions:
$\underline{Z}_D = \sum_m \sum_{j\in N} \gamma_j Y_{jm}^{n+1} + \sum_{i\in N} L_i{}^n$, and $\bar{Z}_k = \sum_{i\in N} \sum_{j\in N} \lambda_i t_{ij} \bar{X}_{ij}^k$.

2F. Set $UB = \min\left\{UB, \min_k \bar{Z}_k\right\}$, $LB = \max\left\{LB, \underline{Z}_D\right\}$, $X_{ij}^{n+1} = \text{argmin}_{\bar{X}_{ij}^k}(\bar{Z}_k)$, and $\hat{Z}_3\left(X_{ij}^{n+1}, L_i^n\right)$ (**P3**). If $UB = LB$ and $(Y_{jm}^{n+1}, X_{ij}^{n+1})$ satisfy constraint (15), go to Step 4, else go to Step 3.

Step3. Update multipliers.

3A. Determine $\tau = (UB - LB)/\sum_i\left(\sum_j X_{ij}^{n+1} - 1\right)^2$

3B. Update multipliers: $L_i^{n+1} = \max\left\{0, L_i^n - \tau\left(\sum_j X_{ij}^{n+1} - 1\right)\right\}$. If $n < n_{max}$, set $n = n + 1$ and go to Step 1, else go to Step 4.

Step 4. Set $\left(Y_{jm}^*, X_{ij}^*, L_i^*\right) = (Y_{jm}^n, X_{ij}^n, L_i^n)$. Solve **P4** to obtain D_j^*, S_j^*, W_{ij}^* to accompany $(Y_{jm}^*, X_{ij}^*, L_i^*)$.

Contrary to the earlier study by Marianov and Serra (2002), our proposed algorithm does not rely on random initial solutions. For large-scale problems, these random methods can run until the computer is out of memory without termination, whereas our algorithm can find good solutions to the same instances in a reasonable time.

4.2. Computational tests

We provide twelve instances to test the computational performance of our proposed heuristic. Data for these instances are publicly available at BUILT@NYU Lab's data library: https://github.com/BUILTNYU/Nonmyopic-relocation-MOD. All but one of the instances are solved using an exact method in addition to the proposed algorithm for benchmarking purposes. The exact solution method used in this experiment was implemented using LINDO 8 via the branch and bound approach. The Lagrangian Decomposition algorithm was run in Matlab on an Intel Core i5-2450 CPU with 2.5 GHz and 8 GB RAM, running on a 64-bit Windows 7 operating system.

Table 1 presents the computational results obtained on the 12 test problems to compare the LD solution with the exact solution as well as the LP relaxed lower bound formulation discussed in Section 3.3. The exact algorithm is unable to converge to a solution in one hour for problem 12. For problems 1 to 4, we are able to find exact solutions, but we reach $n_{max} = 5$ million iterations for large problems between 5 to 12. The maximum difference between the exact and the LD solution is at most 7% in the instances tested, with only a *fraction* of the computation time of the exact solution method. This is a crucial advantage, since the algorithm needs to run in an online setting using real time data every 5 to 15 minutes. The LP relaxed solution is also very effective, although in some cases the lower bound gap is quite large (denoted with negative GAP values).

Table 1. Comparison results of the Lagrangian Decomposition with exact solution

Prob.	No. of nodes	C	No. of variables	No. of constraints	LD	Exact	LP	GAP		CPU time		
								LD	LP	LD	LP	Exact
1	4	6	64	48	2.410	2.410	2.410	0%	0%	1s	1s	10s
2	5	6	90	60	4.894	4.894	4.894	0%	0%	1s	1s	15s
3	6	6	120	72	18.846	18.846	11.931	0%	-37%	2s	2s	15s
4	7	6	154	84	7.654	7.653	7.653	0%	0%	3s	3s	17s
5	8	6	192	96	9.513	9.414	9.414	1%	0%	3s	3s	17s
6	8	4	176	80	12.345	12.145	12.145	2%	0%	4s	4s	13s
7	9	6	234	108	13.963	13.474	13.474	4%	0%	4s	4s	40s
8	9	4	216	90	30.396	29.892	22.863	2%	-24%	4s	4s	20s
9	10	5	270	110	12.204	11.198	11.198	8%	0%	10s	6s	150s
10	10	6	280	120	28.625	28.051	16.503	2%	-41%	10s	6s	200s
11	15	15	705	315	48.288	44.831	43.758	7%	-2%	20s	12s	600s
12	30	15	2310	630	45.250	N/A	42.411	N/A	N/A	30s	15s	1h

5. New York Taxi Cruising Simulation Experiment

While the proposed model is designed to locate idle vehicles as an infinite horizon queueing system, the reality is that demand used at the start of a time interval may not necessarily reflect an infinite horizon setting, particularly if demand is extremely volatile and non-stationary. The model is an approximation method, and therefore requires computational experimentation to evaluate its limitations.

To determine the effectiveness of the relocation policy, we run it within a simulation and measure the actual performance of a taxi fleet operation instead of comparing only coverage and queue delay costs. The proposed non-myopic policy was tested on a simulation experiment drawn from real data. In 2011, 13,586 taxicabs in New York carried over 172 million trips with passengers, with over 150 million taxi trips that originated or ended in Manhattan (Santi et al., 2013). A significant portion of time and fuel was spent by taxis cruising on non-revenue generating activities without passengers, leading to inefficiencies (Li, 2006).

We conduct a comparison test between a policy operating with the idle taxi relocation with queueing (Scenario A) against a myopic policy in which no queueing is considered (Scenario B). A simulation of pickups and drop-offs of passengers over a representative day is conducted. In each time interval of the simulation, taxis are either busy or idle. Busy taxis are dispatched using the same dispatch algorithm for both scenarios; passenger arrivals are also the same across scenarios. The only difference between scenarios is the idle taxi relocation policy.

5.1. New York Taxi Data

Taxi pickup data is used to represent arrival processes of “picked up passengers”. The data is collected by the NY Taxi and Limousine Commission (TLC) (see Gonzales et al., 2014), who logs the GPS data for every taxi trip in the city, and pickup/drop-off locations and times are made available to researchers as an open data source. All the experiments are conducted using data from six nodes (aggregated at the Neighborhood Tabulation Area (NTA) level) in Manhattan shown in **Table 2.**

Table 2. Selected nodes in Manhattan study area

NTA Code	NTA Name
MN13	Hudson Yards-Chelsea-Flatiron-Union Square
MN15	Clinton
MN17	Midtown-Midtown South
MN19	Turtle Bay-East Midtown
MN20	Murray Hill-Kips Bay
MN21	Gramercy

Figure 4 shows pickups in March of 2013, categorized by all trips in NYC and all trips originating from Public Use Microdata Sample Areas (PUMA) 3807 and 3808 in Manhattan. For our empirical study, we selected six nodes with high congestion. Each file has about 1 million rows, and each node has about 100,000 rows contains household id, pickup date/time drop-off date/time, passenger count, trip time in seconds, trip distance, and latitude/longitude coordinates for the pickup and drop-off locations. Only weekdays are used since weekend taxi demand patterns can differ significantly. For our experiment, 5 full weekdays of data are extracted from the file to estimate the arrival process parameters for the simulation scenario, with each day divided into arrival counts per 15 minutes resulting in 480 observations. The time interval of 15 minutes is chosen to ensure that idle taxis have sufficient time to reach a customer within that interval. Thus, we considered 1 billion samples $(6 \times 100000 \times 5 \times 15 \times 24)$ for six nodes to provide the parameter setting in our large case study.

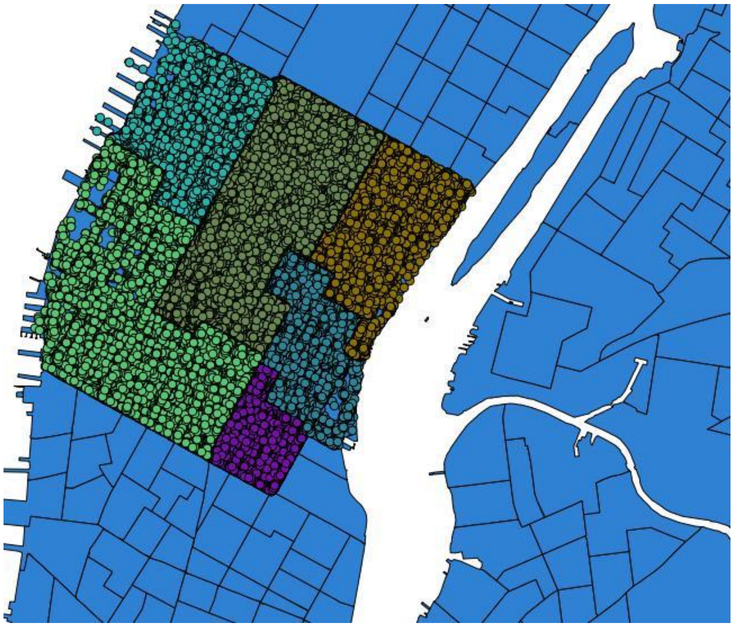


Fig. 4. Pick-up locations of passengers in New York in March 2013 (via QGIS).

To run the idle taxi relocation policy, we use the following parameters with six nodes: $\alpha = 0.95$, $\theta = 0.2, b = 0$, $B = 150$, $Y_0 = \{25,25,25,25,25,25\}$, $C_j = 150 \quad \forall m = 1,2, \dots 150$. The parameters $\lambda_{i\tau}$ and μ_j are exogenous and approximated from historical data. **Figure 5** shows the variation in the number of passengers per 15 minutes in a day. The service time is the time a user is picked up (i.e. matched under a ride-hail system) until they are dropped off. **Figure 6** presents the service rates per 15 minutes for each origin. The travel and relocation costs are calculated from historic data as shown in **Table 3** and **Table 4**.

Table 3. Travel cost from zone *i* to zone *j*

	MN13	MN15	MN17	MN19	MN20	MN21
MN13	0	1.860	1.815	2.140	1.695	1.650
MN15	1.860	0	1.696	1.968	1.873	2.130
MN17	1.815	1.696	0	1.260	1.160	1.660
MN19	2.140	1.968	1.260	0	1.160	1.670
MN20	1.695	1.873	1.160	1.160	0	1.058
MN21	1.650	2.130	1.660	1.670	1.058	0

Table 4. The cost of relocation from zone *i* to zone *j*

	MN13	MN15	MN17	MN19	MN20	MN21
MN13	-	0.186	0.1815	0.214	0.1695	0.165
MN15	0.186	-	0.169633	0.196833	0.187333	0.213
MN17	0.1815	0.169633	-	0.126	0.116	0.166
MN19	0.214	0.196833	0.126	-	0.116	0.167
MN20	0.1695	0.187333	0.116	0.116	-	0.1058
MN21	0.165	0.213	0.166	0.167	0.1058	-

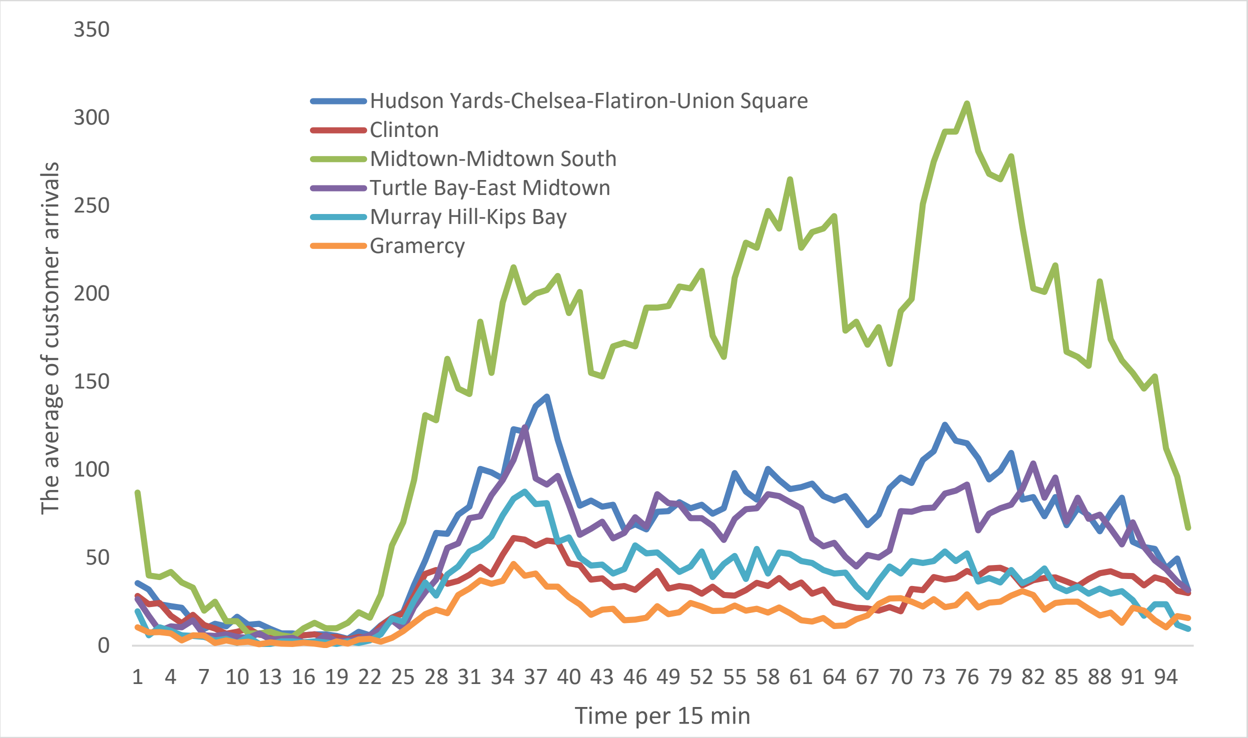


Fig. 5. Customer arrivals per 15 minutes in a 24-hour period (96 periods), averaged over 5 days.

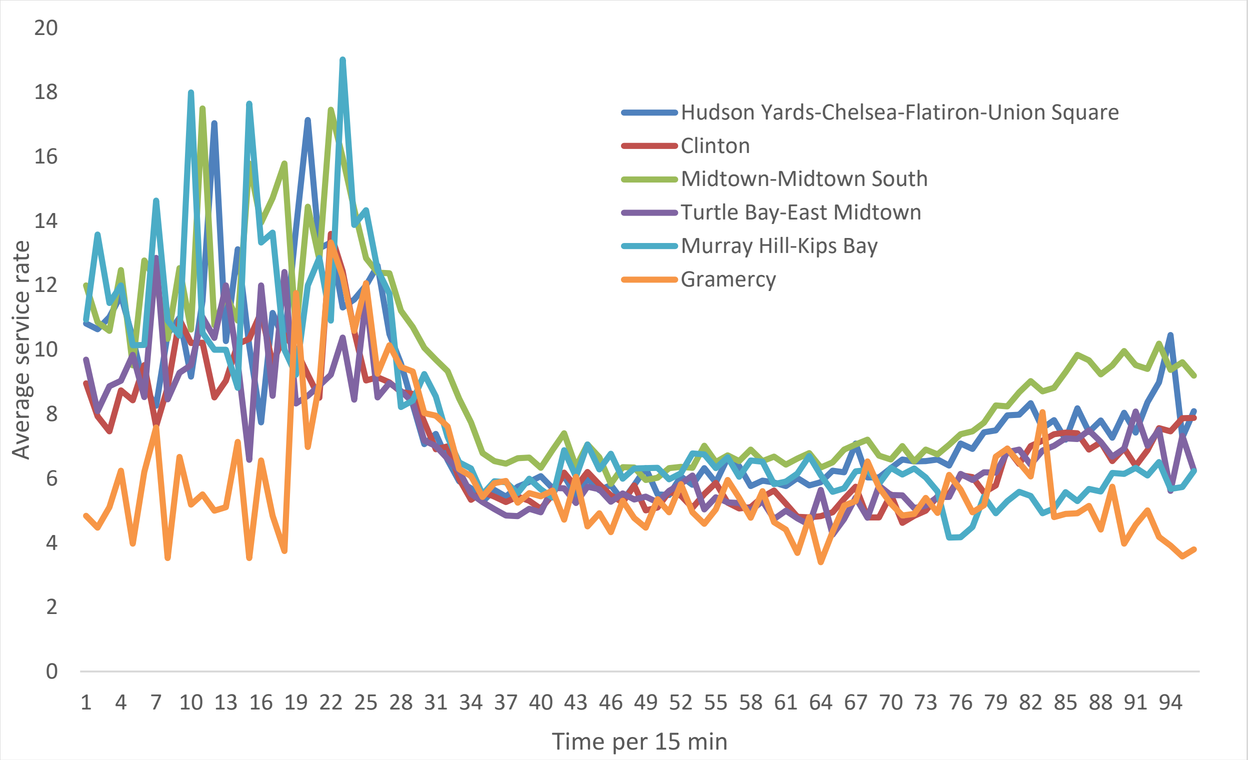


Fig. 6. Service rates per 15 minutes in a 24-hour period (96 periods), averaged over 5 days.

5.2. The simulation setup

We run a simulation of a taxi fleet serving this study area every 15-minute periods over a 24-hr period with 90 passenger arrivals per 15-minute. The passenger arrival zones are distributed according to the $\lambda_{i\tau}$; i.e. if $\lambda_{i\tau} = 2\lambda_{j\tau}$ then zone i is twice as likely to generate a request during that 15 minute interval in the simulation. Destinations of these simulated trips are assumed to be within the six zones in the study area. This experimental design focuses on the ability of the look-ahead approximation method to capture realistic changes in arrival demand over the course of a day.

Two scenarios are simulated; Scenario A uses the proposed relocation policy to direct idle taxis at the start of each time interval, while Scenario B uses a relocation policy that ignores queueing approximation. Both scenarios use the same dispatching and routing policy to simulate the movement of taxis and passengers. For convenience, the Hyytiä policy (Hyytiä et al., 2012) tested in Sayarshad and Chow (2015) is used for assigning vehicles to passengers and for routing the vehicles through the nodes in order to advance the simulation from the output of the idle vehicle relocation algorithm from one time interval to the next. The Hyytiä policy dispatches vehicles based on a cost function shown in Eq (19), where each vehicle is considered for dispatch by solving a traveling salesman problem with pickup and delivery (TSPPD). The vehicle and route with lowest increase in cost is assigned to a new customer. Note that we are not proposing a new dispatch or routing model here, merely using it as part of a simulation to evaluate the long-term performance of the idle vehicle relocation policy over multiple time intervals.

$$c(v,\xi) = \gamma\zeta(v,\xi) + (1-\gamma)\left(\beta\zeta(v,\xi)^2 + \sum_i So_i(v,\xi)\right) \qquad (19)$$

where v is a vehicle, ξ is a tour obtained from TSPPD, c is the cost function, ζ is the tour length, So_i is the total delay for customer i (service plus wait time, i.e. time from call in to time they are delivered), and γ and β are control parameters to adjust the degree of system cost versus user cost (γ) and degree of look ahead (β).

For this experiment, we use a Matlab script to convert Latitude and Longitude coordinate system to X and Y coordinate by inputting the shape file. The following parameters are used for the dispatch algorithm: speed = 4/6 km/min, vehicle capacity = 4, $\gamma = 0.4$, $\beta = 0.5$, and 96 different 15-minute periods over a 24-hr period which do 30 simulated runs for full-day. To ensure reliable results, we run the simulations thirty different times.

Comparisons are made between non-myopic and myopic policies for two solution methods: the proposed LD-based solution method, as well as the LP-relaxed lower bound formulation. As a result, we run 4 scenarios over 30 runs each, for a total of 120 runs.

5.3. Results

We evaluate the average performance of the proposed non-myopic policy against the myopic policy over each 15 minutes. By updating Y_{jm} in each time interval, our approach is able to realize the total number of idle vehicles located at node. In this scenario, we divide one day to 96 different 15-minute periods with 90 passenger arrivals per 15-minute, then we obtain the average system cost per user (T/Q) and the average user cost per user $\left(S_q\right)$ after 30 simulated runs for full-day.

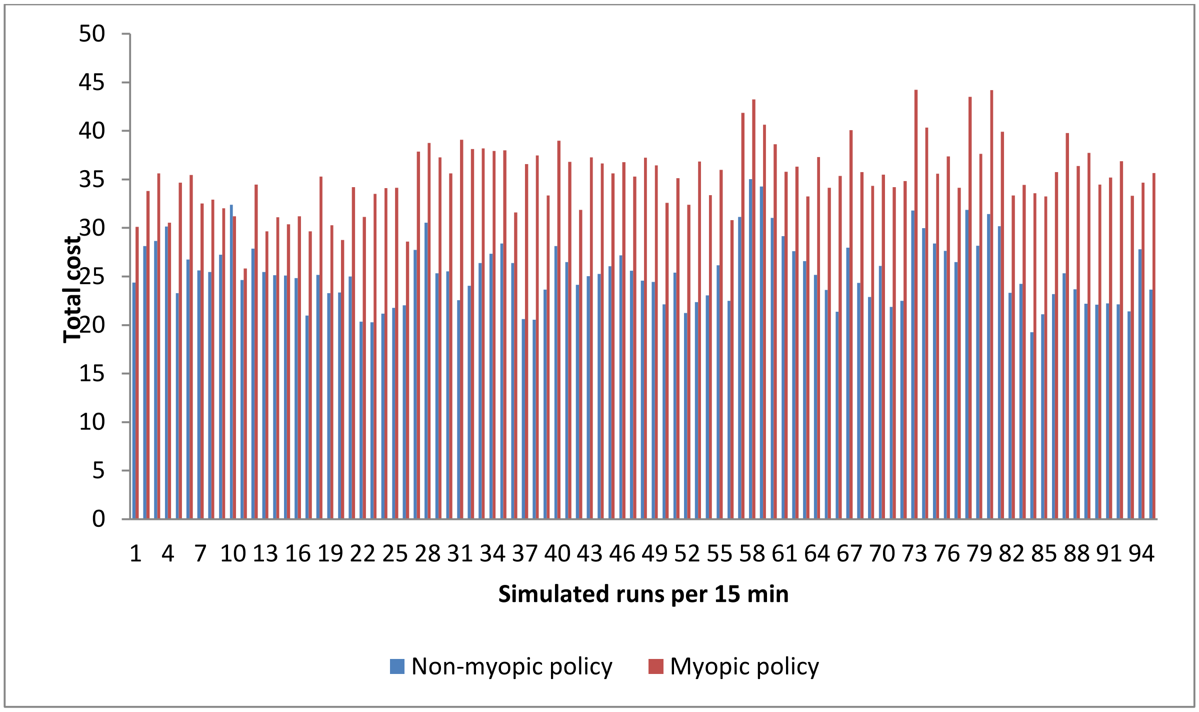
Figure 7 shows a comparison of the total realized cost for each of the 15 minutes over the two policies. Under non-myopic location policy, the average system cost per user is 12.5955 and the average user cost per user is 12.8289, while for myopic allocation policy it is 17.64 and 17.6664, respectively. Furthermore, the non-myopic preposition can have an even greater effect where the minimum, average and maximum total cost under non-myopic is 19.256, 25.4245, and 35.0269, whereas for the myopic policy it is 25.8385, 35.3065, and 44.2333. The non-myopic allocation problem performs better than myopic case where the total cost decreases by 25% (Min), 27% (Ave), and 20% (Max) over the 30 runs.

Using the LP relaxed lower bound, solutions may end up being infeasible leading to suboptimal decisions during the simulation of the fleet operations. The minimum, average and maximum total costs under non-myopic are 22.145, 28.844, and 36.875, respectively, whereas for the myopic policy they are 29.8596, 38.7543, and 47.5820, respectively. The non-myopic allocation problem performs better than myopic case where the total cost decreases by 25% (Min), 25% (Ave), and 22% (Max) over the 30 runs. Overall, however, it is clear that a non-myopic policy based on the LD algorithm outperforms a similar policy using the lower bound decisions by 11.9%.

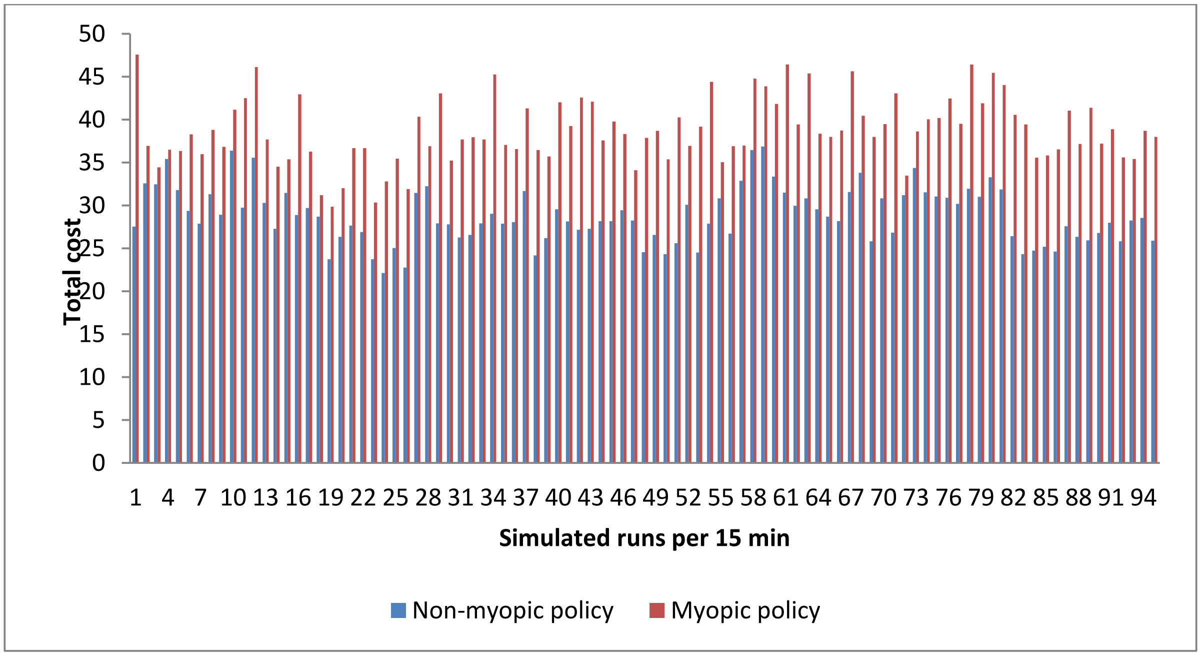
One particularly interesting result is that the proposed approach can reduce the total time the customer must wait, and thus improve the quality of the service. From the point of view of facility location, the proposed non-myopic policy is able to allocate of the right number of idle vehicle to each node and the optimal assignment of customers to them.

Table 5. Comparisons of the accrued costs over the 30 runs for each of the four scenarios.

Over 30 runs	Myopic-LP	Non-myopic LP	Myopic-LD	Non-myopic LD
Min cost	29.860	22.145	25.839	19.256
Avg cost	38.754	28.844	35.307	25.425
Max cost	47.582	36.875	44.233	35.027



a. The results by the LD-based policy



b. Results by the LP relaxed lower bound-based policy

Fig. 7. Comparison of 96 consecutive 15-minute costs between proposed policy (under (a) LD algorithm and (b) LP lower bound) and a myopic policy, averaged over 30 simulation runs.

6. Conclusions

We proposed a new formulation and solution algorithm for the online policy of relocating idle vehicles in a demand responsive service system. To the best of our knowledge, this is the first dynamic facility location model to use queueing-based approximation for look-ahead. We proposed a Lagrangian decomposition heuristic to solve this model, and conducted various tests to demonstrate the effectiveness of the online policy, the model formulation, and solution algorithm. Computational tests to evaluate the performance of the algorithm across twelve different instances show promise. Lastly, a case study using NYC taxi data demonstrated the effectiveness of the algorithm under a simulated fleet operation environment when compared to a myopic policy.

We were able to solve small-size instances by an exact approach in a fair amount of CPU time, but we were unable to solve the problem to optimality for medium and large- size instances. To tackle this problem, a Lagrangian Decomposition (LD) algorithm was proposed to solve the model. Numerical examples are solved to check for the efficiency and validity of the Lagrangian Decomposition algorithm. The difference between the exact solution and the LD solution in the instances tested is at most 6% - 7%, which is reasonable. An LP relaxed lower bound was also computed for comparison.

We empirically proved the effectiveness of the proposed algorithm by comparing it against a myopic prepositioning policy using taxicab passenger data and a common dispatch algorithm. The proposed queueing-based relocation policy is compared to a myopic policy that ignores queueing, using the same simulation of passenger arrivals, initial taxi positions, same dispatch and routing algorithm, and same length of day. The proposed algorithm is shown to improve over the myopic case by an average of 27% in total cost across thirty simulation runs.

One key future research direction is identifying the impact of the relocation policy on fuel use, which can have a great effect on the environment and energy savings (see Greenblatt and Saxena, 2015), vehicle miles traveled (VMT), and traffic congestion. Finally, there are opportunities to integrate the relocation policy with other dynamic optimization policies to systematically control hybrid fixed and on-demand automated vehicle fleets. Further policies can be examined in conjunction, including fare structure, parking charges, transit level of service, and overall system level of service.

Acknowledgements

This research was undertaken, in part, thanks to funding from the Canada Research Chairs program. One co-author was partly supported by staff and resources from the C^{2}SMART University Transportation Center, which is gratefully acknowledged. Any errors found are solely the authors'. Comments from four anonymous referees are much appreciated.

Appendix: Illustration of algorithm

To help explain the algorithm, we present a step-by-step illustration of one iteration of an example. At a particular iteration for an example with 4 nodes, suppose we have the following Lagrange multipliers $\mathrm{L}_1 = 0.5,\ \mathrm{L}_2 = 0.6, \mathrm{L}_3 = 0.7, \mathrm{L}_4 = 0.8$ and the matrix

$$H = \begin{bmatrix} 0 & 3.93 & 5.24 & 2.64 \\ 3.13 & 0 & 2.55 & 4.86 \\ 6.05 & 3.7 & 0 & 4.55 \\ 3.63 & 8.39 & 5.41 & 0 \end{bmatrix}$$

where $H = \left(h_{ij}\right) = \lambda_i t_{ij}\ \forall\ i, j$. The matrix γ is updated for each element $\gamma_{ij} = \min(0, h_{ij} - L_i)$.

$$\gamma = \begin{bmatrix} -0.5 & 0 & 0 & 0 \\ 0 & -0.6 & 0 & 0 \\ 0 & 0 & -0.7 & 0 \\ 0 & 0 & 0 & -0.8 \end{bmatrix}$$

At the current iteration, the current solution is at $Y_{21}^1 = Y_{41}^1 = 1$, and $Y_{11}^1 = Y_{12}^1 = Y_{22}^1 = Y_{31}^1 = Y_{32}^1 = Y_{42}^1 = 0$. This implies the primal solutions $\bar{X}_{ij}^0$ and dual solutions X_{ij}^1 as shown below. Again, $X_{ij}^1 = 1$ if $Y_{jm}^1 = 1$ and $\gamma_{ij} < 0$ and $\bar{X}_{ij} = 1$ if $Y_{jm}^1 = 1$ and $\mathrm{H} = \min(\lambda_i t_{ij})\ \forall i, j, m$.

$$\bar{X}^0 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} and\ \ X^1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The value $\bar{X}^0$ is perturbed four times, once for each row, to swap the value of one from the current solution to a column with the next lowest value of h_{ij}. Each newly generated solution is shown below. All of them are checked for feasibility (they are all feasible).

$$\bar{X}^1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ \bar{X}^2 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ \bar{X}^3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{and}$$

$$\bar{X}^4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Then $\bar{Z}_k = \sum_{i \in N} \sum_{j \in N} \lambda_i t_{ij} \bar{X}_{ij}^k = 0 + 3.70 + 2.64 + 0 = 7.346$, and $\underline{Z}_D = \sum_m \sum_{j \in N} \gamma_j . Y_{jm}^{n+1} + \sum_{i \in N} L_i{}^n = -0.6 - 0.8 + 2.6 = 1.2$, so that $UB = 7.364$ and $LB = 1.2$. Since there is a gap, we need to update the multipliers. We first compute the step size $\tau = \frac{1(7.364 - 1.2)}{1+0+1+0} = 3.07$. The numbers in the denominator are based on the dual solution X_{ij}^1 that, at present, provides single assignments in row 2 and 4, and no assignment in row 1 and 3. The new multipliers are then calculated as $\mathrm{L}_1 = 0.5 - 3.07(-1) = 3.57$, $\mathrm{L}_2 = 0.6$, $\mathrm{L}_3 = 0.7 - 3.07(-1) = 3.77, \mathrm{L}_4 = 0.8$. From there, the algorithm would continue to the next iteration until there is no gap or if the maximum number of iterations is reached.

References

Agatz, N., Erera, A., Savelsbergh, M., Wang, X., 2012. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2), 295-303.

Batta, R., 1989. A queuing–location model with expected service time dependent queuing disciplines, *European Journal of Operational Research* 39(2), 192–205.

Batta, R., Larson, R., Odoni, A., 1988. A single-server priority queuing–location model, *Networks* 18(2), 87–103.

Berman O., Drezner, Z., 2007. The Multiple Server Location Problem, *Journal of the Operational Research Society*, 58(1), 91-99.

Berman, O., Larson, R., Chiu, S., 1985. Optimal server location on a network operating as a M/G/1 Queue, *Operations Research* 33(4), 746–771.

Berman, O., Larson, R.C., Parkan, C., 1987. The stochastic queue p-median problem. *Transportation Science* 21(3), 207-216.

Berman, O., Odoni, A.R., 1982. Locating mobile servers on a network with markovian properties. *Networks* 12(1), 73-86.

Boccia, M., Sforza, A., Sterle, C., Vasilyev, I., 2008. A cut and branch approach for the capacitated p-median problem based on Fenchel cutting planes. *Journal of Mathematical Modelling and Algorithms* 7(1), 43–58.

Bradley, S., Hax, A. and Magnanti, T., 1977. *Applied mathematical programming*. Addison-Wesley.

Brotcorne, L, G. Laporte, Semet, F., 2003. Ambulance location and relocation models, *European Journal of Operational Research* 147(3), 451–463.

Brownell, C., Kornhauser, A., 2014. A driverless alternative: fleet size and cost requirements for a statewide autonomous taxi network in New Jersey. *Transportation Research Record* (2416), 73-81.

Ceselli, A., 2003. Two exact algorithms for the capacitated p-median problem. *4OR* 1(4), 319–340.

Ceselli, A., Righini, G., 2005. A branch-and-price algorithm for the capacitated p-median problem. *Networks* 45(3), 125–142.

Cheng, S., Qu, X., 2009. A service choice model for optimizing taxi service delivery, In: 12th International IEEE Conference on Intelligent Transportation Systems, St. Louis, MO.

Chow, J.Y.J., Regan, A.C., 2011a. Resource location and relocation models with rolling horizon forecasting for wildland fire planning. *INFOR* 49(1), 31-43.

Chow, J.Y.J., Regan, A.C., 2011b. Network-based real option models. *Transportation Research Part B* 45(4), 682-695.

Chow, J.Y.J., Sayarshad, H.R., 2015. Reference policies for non-myopic sequential network design and timing problems. *Networks and Spatial Economics*, 16(4), 1183-1209.

Contardo, C., Morency, C., Rousseau, L. M., 2012. Balancing a dynamic public bike-sharing system, CIRRELT.

Cooper, L., 1964. Heuristic methods for location-allocation problems. *SIAM Review* 6(1), 37–53.

Cornuejols, G., Sridharan R., Thizy, J., 1991. A comparison of heuristics and relaxations for the capacitated plant location problem, *European Journal of Operational Research* 50, 280–297.

Current, J., Storbeck, J., 1988. Capacitated covering models, *Environment and Planning B* 15,153–164.

Daganzo, C. F., 1978. An approximate analytic model of many-to-many demand responsive transportation systems. *Transportation Research*, 12(5), 325-333.

Davis, P., Ray, T., 1969. A branch and bound algorithm for the Capacitated Facilities Location Problem, *Naval Research Logistics Quarterly* 16, 331–334.

Djavadian, S., Chow, J.Y.J., 2016. Agent-based day-to-day adjustment process to evaluate dynamic flexible transport service policies. *Transportmetrica B: Transport Dynamics*, in press.

Fagnant, D. J., Kockelman, K. M., Bansal, P., 2015. Operations of shared autonomous vehicle Fleet for Austin, Texas, market. *Transportation Research Record* (2536), 98-106.

Fernandez, J.E., De Cea, J., Briones, J., 2006. A diagrammatic analysis of the market for cruising taxis. *Transportation Research Part E* 42(6), 498–526.

Gan, H.C., Ye, X., Wang, Q., 2010. Investigating the effect of travel time variability on drivers' route choice decisions in Shanghai, China. *Transportation Planning and Technology* 33(8), 657–669.

García-López, F., Melián Batista, B., Moreno Pérez, J.A., Moreno-Vega, J.M., 2003. Parallelization of the scatter search for the p-median problem. *Parallel Computing* 29(5), 575–589.

Garrick, D., 2016. Car2Go ceases San Diego operations. *The San Diego Union-Tribune*, December 31, 2016.

Gendreau, M., Laporte, G., Semet, F., 2006. The maximal expected coverage relocation problem for emergency vehicles. *Journal of the Operational Research Society* 57(1), 22-28.

Geoffrion, A., 1974. Lagrangean relaxation for integer programming, Mathematical Programming Study 2, 82–114.

Gonzales, E.J., Yang, J., Morgul, E.F., Ozbay, K., 2014. Modeling Taxi Demand with GPS Data from Taxis and Transit. MNTRC Report 12-16, Mineta National Transit Research Consortium.

Greenblatt, J.B., Saxena, S. 2015. Autonomous taxis could greatly reduce greenhouse gas emissions of U.S. light-duty vehicles, *Nature Climate Change* 5, 860–863

Guignard. M., 2003. Lagrangian relaxation. *TOP* 11(2), 151–228.

Hou, Y., Li, X., Zhao, Y., Jia, X., Sadek, A. W., Hulme, K., Qiao, C., 2013. Towards efficient vacant taxis cruising guidance. Proc. 2013 IEEE Global Communications Conference, Atlanta, GA.

Hyytiä, E., Penttinen, A., Sulonen, R., 2012. Non-myopic vehicle and route selection in dynamic DARP with travel time and workload objectives. *Computers & Operations Research* 39(12), 3021-3030.

Jung, J., Chow, J.Y.J., Jayakrishnan, R., Park, J.Y., 2014. Stochastic dynamic itinerary interception refueling location with queue delay for electric taxi charging stations. *Transportation Research Part C* 40, 123-142.

Kolesar, P., Walker, W. E., 1974. An algorithm for the dynamic relocation of fire companies. *Operations Research* 22(2), 249–274.

Levanova, T., Loresh, M.A., 2004. Algorithms of ant system and simulated annealing for the p-median problem. *Automation and Remote Control* 65(3), 431–438.

Li S. Multi-attribute taxi logistics optimization. Master's thesis, Massachusetts Institute of Technology, 2006.

Li, B., Zhang, D., Sun, L., Chen, C., Li, S., Qi, G., Yang. Q., 2011. Hunting or waiting? discovering passenger-finding strategies from a large-scale real-world taxi dataset. In: 2011 IEEE Conference on Pervasive Computing and Communications Workshops, Seattle, WA, 63 –68.

Li, Y., Miller, M.A., Cassidy, M.J., 2009. Improving mobility through enhanced transit services: Transit Taxi Service for Areas with Low Passenger Demand Density, working paper UCB-ITS-PRR-2009-18, University of California, Berkeley.

Liang, X., de Almeida Correia, G. H., van Arem, B., 2016. Optimizing the service area and trip selection of an electric automated taxi system used for the last mile of train trips. *Transportation Research Part E* 93, 115-129.

Marianov, V., ReVelle, C., 1994. The queueing probabilistic location set covering problem and some extensions. *Socio-Economic Planning Sciences* 28(3), 167–178.

Marianov, V., ReVelle, C.,1996. The queueing maximum availability location problem: a model for the siting of emergency vehicles. *European Journal of Operational Research* 93(1), 110–120.

Marianov, V., Serra, D., 1998. Probabilistic, maximal covering location–allocation models for congested systems. *Journal of Regional Science* 38(3), 401–424.

Marianov, V., Serra, D., 2002. Location–allocation of multiple-server service centers with constrained queues or waiting times. *Annals of Operations Research* 111(1), 35–50.

Mendes, L.M., Bennàssar, M.R., Chow, J.Y.J., 2017. Simulation experiment to compare light rail streetcar against shared autonomous vehicle fleet for Brooklyn Queens Connector. *Transportation Research Record*, in press.

Mladenović, N., Brimberg, J., Hansen, P., Moreno-Pérez, J.A., 2007. The p-median problem: a survey of metaheuristic approaches. *European Journal of Operational Research* 179(3), 927–939.

Nair, R., Miller-Hooks, E., 2009. Evaluation of relocation strategies for emergency medical service vehicles. *Transportation Research Record* 2137, 63-73.

Nair, R., Miller-Hooks, E., 2012. Fleet management for vehicle sharing operations. *Transportation Science* 45(4), 524-540.

Neebe, W.A., 1978. A branch and bound algorithm for the p-median transportation problem. *Journal of the Operational Research Society* 29(10), 989-995.

NHTSA (National Highway Traffic Safety Administration). U.S. Department of Transportation Releases Policy on Automated Vehicle Development. NHTSA 14–13, May 30, (2013) accessed 26 October 2014.

Nourinejad, M., & Roorda, M. J., 2014. A dynamic carsharing decision support system. *Transportation Research Part E: Logistics and Transportation Review*, 66, 36-50.

Owen, S. H., Daskin, M. S., 1998. Strategic facility location: a review. *European Journal of Operational Research* 111(3), 423–447.

Pasandideh, S.H.R., Niaki, S.T.A., 2012. Genetic application in a facility location problem with random demand within queuing framework. *Journal of Intelligent Manufacturing* 23(3), 651–659.

Phithakkitnukoon, S., Veloso, M., Bento, C., Biderman, A., Ratti, C., 2010. Taxi-aware map: Identifying and predicting vacant taxis in the city. In Proc. AMI, Malaga, Spain, 86-95.

Pirkul H., Schilling, D., 1991. The maximal covering location problem with capacities on total workload, *Management Science* 37, 233-248.

Powell, J., Huang, Y., Bastani, F., Ji. M., 2011. Towards reducing taxicab cruising time using spatio-temporal profitability maps. *Advances in Spatial and Temporal Databases*, eds. Pfoser et al., 242-260.

Raviv, T., Tzur, M., Forma, I. A., 2013. Static repositioning in a bike-sharing system: models and solution approaches. *EURO Journal on Transportation and Logistics*, 2(3), 187-229.

Resende, G.C., Werneck, R.F., 2007. A fast swap-based local search procedure for location problems. *Annals of Operation Research* 150(1), 205–230.

ReVelle, C.. 1993. Facility siting and integer-friendly programming, *European Journal of Operational Research* 65 147–158.

Rosing, K., 1997. An empirical investigation of the effectiveness of a vertex substitution heuristic. *Environment and Planning B* 24(1), 59–67.

Santi, P. Resta, G. Szell, M., 2013. Sobolevsky, S., Strogatz S., Ratti. C. Taxi pooling in New York city: a network-based approach to social sharing problems, working paper.

Sayarshad, H.R., 2015. Smart Transit Dynamic Optimization and Informatics. PhD Dissertation, Ryerson University, Toronto.

Sayarshad, H.R., Chow, J.Y.J., 2015. A scalable non-myopic dynamic dial-a-ride and pricing problem. *Transportation Research* Part B 81, 539-554.

Sayarshad, H.R., Chow, J.Y.J., 2016. Survey and empirical evaluation of nonhomogeneous arrival process models with taxi data. *Journal of Advanced Transportation* 50(7), 1275-1294.

Sayarshad, H.R., Tavassoli, S., Zhao, F., 2012. A multi-periodic optimization formulation for bike planning and bike utilization. Applied Mathematical Modeling 36(10), 4944-4951.

Schaller Consulting, 2006. The New York City Taxicab Fact Book, Schaller Consulting, Brooklyn, NY, available at http://www.schallerconsult.com/taxi/taxifb.pdf.

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

Sirisoma, R.M.N.T., Wong, S.C., Lam, W.H.K., Wang, D., Yan, H., Zhang, P., 2009. Empirical evidence for taxi customer-search model. In: Proc. 88th Annual Meeting of the TRB, Washington, DC.

Thomas, B.W., White III, C.C., 2004. Anticipatory route selection. *Transportation Science* 38(4), 473-487.

Waserhole, A., Jost, V., 2016. Pricing in vehicle sharing systems: Optimization in queuing networks with product forms. *EURO Journal on Transportation and Logistics*, 5(3), 293-320.

Yang, H., Fung, C.S., Wong, K.I., Wong, S.C., 2010. Nonlinear pricing of taxi services. *Transportation Research Part A* 44(5), 337-348.

Yuan, J., Zheng, Y., Zhang, L., Xie, X., Sun, G., 2011. Where to find my next passenger? Proc. UbiComp 2011, New York, NY.

Zhang, D., He, T., Lin, S., Munir, S., Stankovic J., 2015. Online cruising mile reduction in large-scale taxicab networks. *IEEE Transactions on Parallel and Distributed Systems*, 26(11):3122–3135, 2015.

Zhang, D., Yu, C., Desai, J., Lau, H. Y. K., Srivathsan, S., 2016. A time-space network flow approach to dynamic repositioning in bicycle sharing systems. *Transportation Research Part B: Methodological*, http://dx.doi.org/10.1016/j.trb.2016.12.006.

Zhang, R., Pavone, M., 2016. Control of robotic mobility-on-demand systems: a queueing-theoretical perspective. *The International Journal of Robotics Research*, 35(1-3), 186-203.

View publication stats