Transportation Research Part E 66 (2014) 36–50

Contents lists available at ScienceDirect

Transportation Research Part E

journal homepage: www.elsevier.com/locate/tre

A dynamic carsharing decision support system

Mehdi Nourinejad *, Matthew J. Roorda [1]

Department of Civil Engineering, University of Toronto, 35 St. George Street, Toronto, ON M5S 1A4, Canada

ARTICLE INFO

Article history:
Received 29 July 2013
Received in revised form 2 January 2014
Accepted 16 March 2014

Keywords:
Carsharing
Decision support system
Optimization
Simulation

ABSTRACT

This paper proposes a dynamic optimization–simulation model as a decision support system for one-way carsharing organizations. To reduce the vehicle imbalance in one-way systems, a Vehicle Relocation Optimization model is solved successively in a discrete event simulation. Each event is the arrival of a new user. The model is compared to an a priori benchmark model. Autoshare is chosen as a case study. Results show that increasing the reservation time (time between requesting and picking up a vehicle) from 0 to 30 min reduces fleet size by 86%. The model captures a tradeoff between vehicle relocation hours and fleet size.

1. Introduction

Urban carsharing services provide individuals with access to a fleet of shared-use vehicles without the costs and responsibilities of private vehicle ownership. Members of these services typically pay for subscription-access plans and are charged through hourly rates. Further benefits of carsharing are reduced parking costs, mitigated environmental impact, and availability of an alternative transportation mode (Katzev, 2003). City Carshare in San Francisco, the largest non-profit carsharing organization in North America, released an environmental report in 2013 outlining its role in reducing a total of 25 million vehicle miles, 85 million pounds of CO_2 emissions, and 4.3 million gallons of gasoline (City Carshare, 2013).

CarSharing organizations (CSO) are commonly classified based on configuration into one-way and two-way systems. Two-way systems (e.g., Zipcar and Autoshare) restrict vehicles to be picked up from and returned to the same station. One-way carsharing systems (e.g., ICVS and Praxitele), on the other hand, permit users to return the vehicle to a location of choice as long as the drop-off station and time is indicated in advance. While two-way systems are more common and account for 94% of all North American carsharing memberships (Shaheen et al., 2006), one-way systems are less adopted. This is mainly due to the issue of vehicle imbalance which happens when cars shift towards certain destinations in the network. Some CSOs such as Car2Go address vehicle imbalance by employing drivers to relocate the vehicles to high demand locations. Such relocation operations increase costs for the CSOs.

Despite high relocation costs, the number of one-way systems is rising. Communauto, a privately owned carsharing organization founded in city of Québec in 1994, has inaugurated the first electric one-way carsharing service in Canada (Communauto, 2013). This pilot project aimed to evaluate the benefits of one-way systems and was initiated due to public consultations that showed the demand for such systems. To complement such pilot projects, better dynamic vehicle relocation decision support tools need to be designed which consider dynamically the location of all vehicles in the fleet and locations of new user requests. Accounting for these two, the objective is to minimize total vehicle relocation costs. This

* Corresponding author. Tel.: +1 647 262 6234.
E-mail addresses: mehdi.nourinejad@mail.utoronto.ca (M. Nourinejad), roordam@ecf.utoronto.ca (M.J. Roorda).
[1] Tel.: +1 416 978 5976.

http://dx.doi.org/10.1016/j.tre.2014.03.003
1366-5545/

tactical model differs from higher level decision making models which mainly focus on where to locate carsharing parking stations and what fleet size to use based on aggregate demand values.

The main objectives of this paper are as follows:

- Present a benchmark model that considers simultaneously the complete set of all user requests received in a particular day assuming user requests are known in advance.
- Propose a dynamic integrated simulation–optimization model which takes online user requests and acts as a decision support tool for CSOs to maximize system profit.
- Perform sensitivity analysis on the fleet size of each system configuration and highlight the important factors and policies which impact both the fleet size and vehicle relocation costs.

This paper is structured as follows. In Section 2, we describe the literature review of previous operational models on carsharing systems. In Section 3, we explain the user preferences, constraints, and problem assumptions. Sections 4 and 5 present the benchmark model and the dynamic model, respectively. In Section 6, we analyze both models for the case of Autoshare in Toronto. Finally, in Section 7, we highlight the major findings of the article.

2. Literature review

Previous research on carsharing mainly focuses on its environmental impacts (Steininger et al., 1996; Cervero et al., 2007; Firnkorn and Muller, 2011), market dynamics (Shaheen and Cohen, 2007; Shaheen et al., 2006; Vine et al., 2013), users' behavior (Celsor and Millard-Ball, 2007; Morency et al., 2010; Habib et al., 2009, 2012), and relationship with public transit (Stillwater et al., 2009). The core of CSO operations, however, has received less attention. Table 1 presents previous operations models of CSOs.

Barth and Todd (1999) develop a simulation model of carshare operations with inputs and measures of effectiveness that allow for scenario analysis. They conclude that a sufficient fleet size for satisfying customers is 3–6 vehicles for every 100 trips but that 18–24 vehicles per 100 trips are required to minimize relocation costs. Fan et al. (2008) propose a multi-stage stochastic linear integer model which attempts to capture system uncertainties such as carsharing demand variation. The objective function of their model maximizes the revenue obtained from servicing customers while minimizing the cost of vehicle relocation.

More recently, Kek et al. (2009) and Correia and Antunes (2012) propose two distinct mixed integer programming models (MIP) that aim at optimizing specific features of CSO operations. Kek et al. design a novel three phase optimization-trend-simulation (OTS) decision support system for CSOs to indicate a set of near-optimal manpower and operating parameters for the vehicle relocation problem. For a carsharing company in Singapore, they conclude that optimization of manpower can reduce staff expenses by up to 50% and zero vehicle time (duration of vehicle shortage at parking stations) by up to 13%. Correia and Antunes (2012), on the other hand, focus on the fleet size, number of vehicle relocations, depot size, and location of potential parking stations. Considering all these decision variables, the authors present a mixed integer optimization MIP approach to maximize CSO revenues while minimizing costs such as vehicle maintenance, parking provision, vehicle depreciation, and vehicle

Table 1
Classification of previous work on operational models of carsharing.

Authors (year)	Objective function	Main decision variables	Solution methodology	System configuration	Study area
Barth and Todd (1999)	Minimize average wait time, number of customers waiting, number of relocations	Effective fleet size	Simulation	One-way	Coachella Valley
Fan et al. (2008)	Maximize revenue, minimize vehicle relocations	Vehicle usage, fleet size	Stochastic programming	One-way	_
Kek et al. (2009)	Minimize vehicle relocation, minimize staff utilization cost, minimize demand rejection penalty	Crew size, staff waiting time, vehicle relocation	Mixed Integer Programming (MIP)	One-way	Singapore
El Fassi et al. (2012)	Maximize member satisfaction, minimize fleet size	Parking capacity, station locations	Discrete event simulation	_	Montreal
Correia and Antunes (2012)	Maximize revenue, minimize vehicle maintenance, relocation, and depreciation	Depot size, depot location, fleet size, vehicle relocations	Mixed Integer Programming (MIP)	One-way	Lisbon
Jorge et al. (2012)	Maximize revenue, minimize vehicle maintenance, relocation, and depreciation	Depot size, depot location, fleet size, vehicle relocations	Simulation – Mixed Integer Programming (MIP)	One-way	Lisbon
Correia and Jorge (in press)	Maximize total daily revenue, minimizing maintenance cost, operational cost of a vehicle, and vehicle ownership cost	Number of vehicles that are parked at each station, origin and destination stations of each trip	Mixed Integer programming (MIP)	One-way	Lisbon

relocation. The study concludes that a vehicle to trip ratio of 22.7 per 100 trips is the optimal fleet composition which confirms the finding of Barth and Todd (1999).

Correia and Antunes and Kek et al. both verify the importance of system optimization in reducing the expenses of one-way carsharing systems. Both, however, consider temporally aggregated demands at different time steps which result in static models. This can lead to unrealistic results if the time steps are too large. Furthermore, the results can also be biased since trip specifications such as user request time are neglected. Realizing this drawback, Jorge et al. (2012) apply the Correia and Antunes model in a simulation platform to capture the impact of demand variability on model results. They use the traditional minimum cost flow algorithm to setup the relocation problem and set the fleet size to the maximum number of required vehicles during the simulation. The model, while acknowledging the importance of user request patterns, does not capture trade-offs between CSO features such as fleet size and vehicle relocations. El Fassi et al. (2012) propose a more holistic discrete event simulation that assists decision makers in selecting the best system improvement strategies. The model considers the demand growth of the system while maximizing members' satisfaction and minimizing the fleet size.

A comprehensive literature review paper is proposed by Jorge and Correia (2013) with a focus on carsharing demand models and one-way system management models. The results of this study show that carsharing demand modeling is difficult and most of the available models are case specific. On one-way systems, the review highlights two major modeling approaches which are simulation and optimization. Although simulations models offer very detailed representations of the systems they commonly lack relocation of vehicles between stations. Some optimization models, on the other hand, do consider vehicle balancing between stations but are limited in scope, lack an acceptable level of detail, and make many simplifying assumptions. One of such simplifications is to assume that users will only request service from one specified station. While this may be valid in cases where the stations are located far from each other (thus eliminating the chance that users may use multiple stations) it is not strongly justified in cases where stations are within walking distance of each other. To address this issue, Correia and Jorge (in press) propose that users can choose from a number of stations by checking real-time information on vehicle stocks at different stations.

3. Carsharing problem setting

3.1. User schedules

Carsharing members' needs can be accommodated if certain constraints are observed. Time is one important concern. Users who wish to leave their origin and arrive at their destination at a specific time have to provide information on their time schedule preferences. We assume that trips have certain attributes such as a desired departure time (from origin parking station), arrival time (to destination parking station), and reservation time. Trip reservation time is defined here as the time between the user's announcement time and the departure time, where the announcement time is the time when a user requests a vehicle. A minimum reservation time can be imposed by CSOs. Fig. 1 shows a generic illustration of user time schedules.

3.2. Fleet size and vehicle relocation

CSOs make decisions about fleet size and vehicle relocation policy. A CSO can reduce its fleet size by increasing vehicle relocations. Furthermore, increasing the required reservation time in the system allows more time to relocate the vehicles. Fig. 2 illustrates the trade-off between fleet size and vehicle relocation where user u_2 can be serviced by either relocating a vehicle from user u_1's destination to u_2's origin (Fig. 2a, dotted arrow) or positioning an extra vehicle at u_2's origin (Fig. 2b). Vehicle relocation requires that the time between u_1's arrival and u_2's departure is longer than the travel time between u_1's destination and u_2's origin. The relocation travel time must also be less than the reservation time for passenger u_2.

3.3. Solution method

Two solution models are presented for the one-way carsharing problem: (i) benchmark model and (ii) dynamic model. The benchmark model assumes perfect information of all user requests including user announcement times, departure times, arrival times, origin locations, and destination locations. The objective of the benchmark model is twofold. First, the model is a tool to assess policies assuming that the CSO has advance information about users' itineraries. Second, the

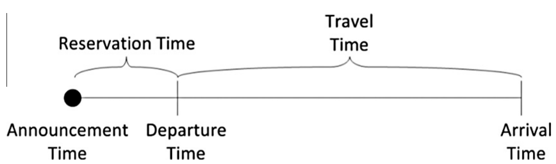

Fig. 1. Time schedule of users.

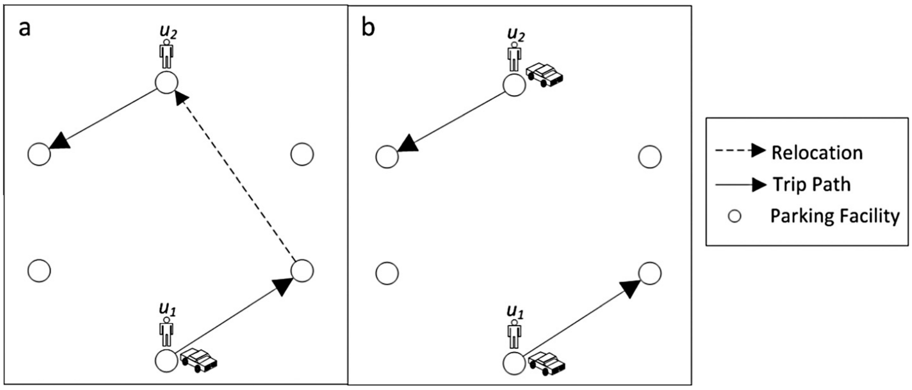

Fig. 2. (a) Relocation of the single vehicle in the fleet to serve both users and (b) provision of an additional vehicle to the fleet.

benchmark is designed to be compared to a more realistic dynamic model where users enter the system at their announcement times. The benchmark model, therefore, is not a good representation of reality since the announcement times of the users are known in advance. However, the obtained results act as an optimistic solution for the dynamic model (presented in Section 5).

The dynamic model is composed of an optimization model which is embedded in a discrete event simulation where each event is triggered when a user requests service. The optimization model itself is made of two sub-models called Vehicle Relocation Optimization (VRO) and Parking Inventory Optimization (PIO) which are executed successively at every event. The VRO model finds optimal relocations and the PIO model find the corresponding relocation times (i.e., when to relocate a vehicle).

A trip generator is used for both the benchmark and the dynamic model to synthesize users in the system. For the benchmark model (presented in Fig. 3 and labeled static), the trip generator produces a specified number of users along with their trip details. Since the benchmark model has perfect information over all users, the entire generated user population is passed onto the model. The dynamic model, on the other hand, does not possess perfect information. Therefore, the trip generator passes individual user trip details to the discrete event simulator at each specific event.

4. Benchmark model

The following are the notations of the sets, parameters, decision variables, and data vectors are used to represent the one-way system configuration:

Sets:

- $U = \{u_1, \ldots, u_i, \ldots, u_r\}$: set of members who wish to use the service during study period.
- P: set of all reserved parking stations of the CSO.
- K: set of all vehicles; this set is used only in the dynamic model. In the benchmark model, we assume the number of vehicles to be a decision variable denoted by $|K|$.

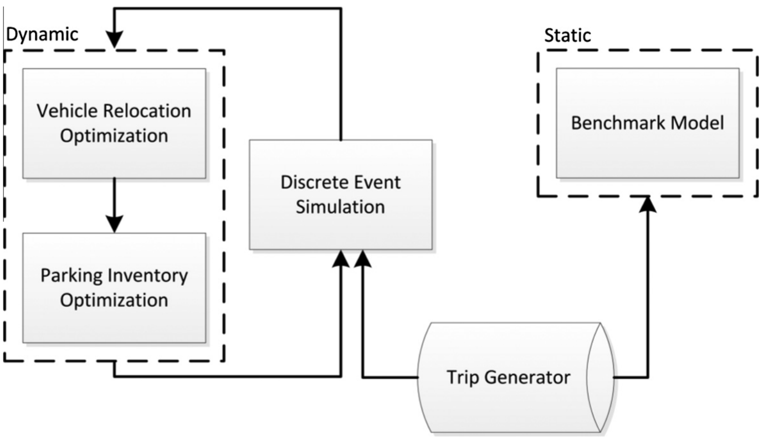

Fig. 3. Schematic of the two solution models.

Parameters:

- γ_p: unit cost of parking at parking station p [dollars per minute] paid by the CSO.
- Pr: price rate based on minutes traveled by users [dollars per minute].
- Cr: unit cost of relocation based on time [dollars per minute].
- F: system reservation time [minutes].
- n_k: initial location of vehicle k. This initial location can change in each iteration of the dynamic model.
- n_0: a hypothetical depot.

Decision variables:

- $x_{ij} = 1, 0$: 1 if a vehicle is relocated from the destination of user i to the origin of user j.
- $y_{kj} = 1, 0$: 1 if a vehicle is relocated from n_k to the origin of user j; this decision variable is only used in the dynamic model where the initial location of each vehicle k is n_k.
- $|K|$: number of required vehicles.
- st_{ij}: time when a vehicle leaves the destination of user i to relocate to the origin of user j.
- st_{kj}: time when a vehicle leaves its initial station (at the start of the day) to be relocated to the origin of user j.
- Av_k: availability time of vehicle k; time when vehicle k becomes available. Availability time of each vehicle can change in each iteration of the dynamic model.

Data vectors:

- M_i: origin parking station of u_i.
- N_i: destination parking station of u_i.
- $tr(a, b)$: travel time of a trip originating at parking station a and terminating at parking station b [minutes].
- A_i: arrival time of user u_i at his/her destination.
- D_i: departure time of user u_i from his/her origin; $D_i = A_i - tr(M_i, N_i)$.
- V_i: trip announcement time of user u_i; $V_i = D_i - F$.
- q_{ij}: maximum of A_i and V_j.

The following assumptions are made for the benchmark model:

1. Total demand is satisfied. This assumption is presented as one of the constraints in the problem formulation but can be relaxed in the dynamic model. Furthermore, we assume that the number of required vehicles is a decision variable in the benchmark model.
2. The capital cost of purchasing a vehicle for the system is much higher than any value of vehicle-km relocation. A new vehicle is added to the fleet only if a user cannot be serviced by any of the existing available vehicles. Similarly, the cost of vehicle relocation is much higher than parking costs.
3. Parking station locations are known in advance and parking spaces are abundant in all the designated stations. However, the variable cost for parking (measured in dollars per hour) in each station can vary.
4. Vehicles can be relocated between parking stations.
5. Reservation time is a policy imposed by the CSO which all users follow exactly. We assume that all user announcement times are their departure times less reservation time. Setting reservation time to zero implies that all users request a car with no prior notice.
6. We consider only trips where the vehicle is booked and returned on the same day.
7. CSOs charge users by minutes of vehicle use.

Using the above notations, the one-way system configuration model is formulated similar to the multiple traveling salesman problem (mTSP) with differences in time window constraints and sub-tour elimination constraints (Kara and Bektas, 2006). The presented Mixed Integer Program formulation is as follows:

$$Min\ \Pi = [Z \times |K|] + \sum_i \sum_j [x_{ij} \times tr(N_i, M_j)] \times Cr + \sum_i \sum_j [\gamma_{N_i} \times (st_{ij} - A_i) + \gamma_{M_j} \times (D_j - st_{ij} - tr(N_i, M_j))] \tag{1a}$$

$$\sum_{i \in \{n_0, 1, 2, \ldots, r\}} x_{ij} = 1 \quad \forall j \in U \tag{1b}$$

$$\sum_{i \in \{n_0, 1, 2, \ldots, r\}} x_{ij} - \sum_{i \in \{n_0, 1, 2, \ldots, r\}} x_{ji} = 0 \quad \forall j \in U \tag{1c}$$

$$q_{ij} = \max(V_j, A_i) \quad \forall i,\ j \in U \tag{1d}$$

$$x_{ij} \times (D_j - tr(N_i, M_j) - q_{ij}) \geqslant 0 \quad \forall i,\ j \in U \tag{1e}$$

$$st_{ij} \leqslant (D_j - tr(N_i, M_j)) \times x_{ij} \quad \forall i,\ j \in U \tag{1f}$$

$$st_{ij} \geqslant A_i \times x_{ij} \quad \forall i,\ j \in U \tag{1g}$$

$$x_{ij} \times tr(N_i, M_j) \leqslant F \quad \forall i,\ j \in U \tag{1h}$$

$$\sum_{j \in \{1,2,\ldots,r\}} x_{n_0 j} = |K| \tag{1i}$$

$$x_{ij} \in \{0,1\} \quad \forall i,\ j \in U \tag{1j}$$

The objective function (1a) is composed of three terms that minimize the fleet size, relocation operations, and parking costs, respectively. The first term is the product of a large number Z and the fleet size which complies with Assumption 2. The second term is the total relocation cost and the third term is composed of two different costs which are the costs of parking at user *i's* destination $(\gamma_{N_i} \times (st_{ij} - A_i))$ and user's *j's* origin $(\gamma_{M_j} \times (D_j - tr(N_i, M_j) - st_{ij}))$, respectively. Once the constants in the third term of (1a) are eliminated, it can be converted to $(\gamma_{N_i} - \gamma_{M_j}) \times st_{ij}$.

This objective function is designed in a way such that $Z \gg Cr \gg \gamma$ which implies that the marginal rates of substitution between the three components can be neglected. In (1a), we neglect the price rate (Pr) of the system because all demand is served which makes the monetary revenue of the system constant and equal to $\sum_i t(M_i, N_i) \times Pr$.

Constraints (1b) ensure that every user is served. Constraints (1c) are flow balancing constraints. In (1b) and (1c), n_0 is a hypothetical depot that is common in the conventional mTSP. The cost of traveling from n_0 to every user's origin is equal to Z and the cost of traveling from every user's destination to n_0 is zero. The high Z value ensures that vehicles are not added to the fleet unless they are needed. The zero cost of every user destination to n_0 ensures tour continuity. Constraints (1d) and (1e) act as sub-tour elimination constraints (Proposition 1). The purpose of (1e) is to ensure time feasibility between user i and *j's* itineraries. Constraints (1f) and (1g) ensure that the relocation time between users i and j is within the allowable time threshold. This threshold is between A_i and $(D_j - tr(N_i, M_j))$. In (1f), a vehicle travels $tr(N_i, M_j)$ minutes and has to arrive to M_j before D_j to serve user j. In (1g), a vehicle can leave N_i for M_j only after user i arrives which is at A_i These constraints along with the third term in the objective function $\left(\left(\gamma_{N_i} - \gamma_{M_j}\right) \times st_{ij}\right)$ indicate three different possibilities for st_{ij}:

1. st_{ij} is equal to 0 whenever x_{ij} is 0,
2. st_{ij} is equal to A_i whenever x_{ij} is 1 and $(\gamma_{N_i} - \gamma_{M_j}) > 0$,
3. st_{ij} is equal to $(D_j - tr(N_i, M_j))$ whenever x_{ij} is 1 and $(\gamma_{N_i} - \gamma_{M_j}) > 0$.

Constraints (1h) ensures that relocating a vehicle between N_i and M_j is only possible when the travel time $tr(N_i, M_j)$ is below the reservation time. Constraint (1i) computes the total number of required vehicles and constraint (1j) specifies the binary decision variables.

Proposition 1. Constraints (1e) act as Sub-tour Elimination Constraints (SEC).

Proof. We define relation R for which the domain is the 2 dimensional set of all *i's* and *j's* and the range is $\{0,1\}$ depending on whether a match between users i and j is possible. Therefore if x_{ij} has the potential to be 1, then $iRj = 1$. Conversely, if x_{ij} does not have the potential to be 1, then $iRj = 0$.

By showing that relation R is asymmetric, we prove that no sub-tour can be made between two users. Similarly, by showing that relation R is intransitive, we prove that no tour can be made between three or more users.

Asymmetric relation: According to (1e) x_{ij} can be equal to 1 ($iRj = 1$) if $A_i \leqslant D_j$ (I). Given the continuous and unidirectional nature of the time continuum, we know $D_j \leqslant A_j$ (II) and $D_i \leqslant A_i$ (III). Considering (I), (II), and (III) we conclude that $A_j \geqslant D_i$ which contradicts constraint (1e) and leads to $jRi = 0$.

Intransitivity relation: We show the intransitivity relation for the case of 3 users i, j, and k which can be easily extended to any higher number of users. Assuming that $iRj = 1$ and $jRk = 1$, we show that $kRi = 0$. From $iRj = 1$ we know $D_i \leqslant A_i \leqslant D_j$ (I′), and from $jRk = 1$ we know $A_j \leqslant D_k \leqslant A_k$ (II′). Given (I′) and (II′), we conclude that $D_i \leqslant A_k$ which contradicts constraint (1e) and leads to $kRi = 0$.

This completes the proof. □

Given the asymmetric relation presented in Proposition 1, the maximum number of solutions that can be considered for the benchmark is the number of edges in a complete graph with r (number of users) nodes which is $r \times (r - 1)$. Many of these potential matches are not feasible due to constraint (1e). We identify such infeasible matches to reduce computation time before running the model to decrease variables and computational burden.

5. Dynamic relocation model

In reality, carsharing companies have a fixed fleet size which is not subject to immediate changes. The dynamic relocation model takes the fleet size as an input and relocates the available vehicles between the users. Contrary to the benchmark model, the constant fleet size in the dynamic model implies that not all users will necessarily be serviced unless the minimum number of needed vehicles (calculated from the benchmark model) is obtained. In other words, any fleet size lower than the maximum fleet size calculated in the benchmark model leads to some unserviced demand. This motivates CSOs to relocate their vehicles with the incentive of maximizing profit. Profit is commonly generated through either travel time or travel distance of the users. Therefore vehicles are most beneficial when assigned to users who travel longer and require less relocation to be accommodated. Furthermore, in reality, CSOs reserve a specified number of parking stalls at various stations and would only have to pay extra if they surpass their reserved capacity (i.e., $\gamma^1 = 0$). This is also addressed in the dynamic model.

We present the dynamic relocation model through an integrated optimization–simulation platform. The optimization stage is composed of two phases: Vehicle Relocation Optimization (VRO) and Parking Inventory Optimization (PIO). In Section 5.1, we present the two optimization phases and explain the simulation platform in Section 5.2.

5.1. Optimization: vehicle relocation and parking inventory

5.1.1. Vehicle Relocation Optimization (VRO)

The VRO phase functions similar to the benchmark with the following exceptions:

1. The price rate is added to the objective function.
2. The fleet size is constant and initial vehicle locations are inputs to the model.
3. Cost of parking (third term of (1a)) is eliminated from the objective function. This cost is later reintroduced in a separate Mixed Integer Program (MIP) in Section 5.1.2. Eliminating the cost of parking in VRO does not imply that vehicles are not relocated but that they do not incur any parking costs.
4. Some demand is left unserviced.

The VRO problem is formulated as follows:

$$Max\ \omega = \sum_i \sum_j [x_{ij} \times [tr(M_j, N_j) \times Pr - tr(N_i, M_j) \times Cr]] - \sum_k \sum_j [y_{kj} \times [tr(M_j, N_j) \times Pr - tr(n_k, M_j) \times Cr]] \tag{2a}$$

$$\sum_{i \in \{1,2,\dots,r\}} x_{ij} + \sum_{k \in \{1,\dots,|K|\}} y_{kj} \leqslant 1 \quad \forall j \in U \tag{2b}$$

$$\sum_{i \in \{1,2,\dots,r\}} x_{ij} + \sum_{k \in \{1,\dots,|K|\}} y_{kj} - \sum_{i \in \{n_0,1,2,\dots,r\}} x_{ji} = 0 \quad \forall j \in U \tag{2c}$$

$$\sum_j y_{kj} \leqslant 1 \quad \forall k \in K \tag{2d}$$

$$y_{kj} \times (Av_k + tr(n_k, M_j) - D_j) \leqslant 0 \quad \forall i,\ j \in U \tag{2e}$$

$$x_{ij} \times (D_j - tr(N_i, M_j) - A_i) \geqslant 0 \quad \forall i,\ j \in U \tag{2f}$$

$$x_{ij} \in \{0, 1\} \quad \forall i,\ j \in U \tag{2g}$$

$$y_{kj} \in \{0, 1\} \quad \forall i,\ j \in U \tag{2h}$$

The objective function ω in (2a) maximizes the total profit of the system by considering the generated revenue from user j ($t(M_j, N_i) \times Pr$) when he/she travels from M_j to N_j and the cost of relocation when a vehicle is transferred from user i's destination to user j's origin or when vehicle k is transferred from its initial location n_k to user j's origin. Constraints (2b) ensure that no user is served more than once. Constraints (2c) are flow balancing constraints with n_0 as a dummy node where the cost of relocating a vehicle from every user's destination to n_0 is zero. n_0 is added to ensure constraints (2c). Constraints (2d) ensure that no vehicle is assigned to two users.

Constraints (2e) guarantee that there is enough time between the availability time of vehicle k and the departure time of user j if the vehicle is to be assigned to that user. Availability time of vehicle k (Av_k) is the point in time when vehicle k is free to be assigned to any user. Availability of vehicles changes every time the model is run and depends on which users are confirmed. This is further elaborated in Section 5.2. Constraints (2f) are similar to (1e) except that q_{ij} is replaced with A_i. This happens because the dynamic model has no a priori knowledge of all users and only considers users who have already announced their requests. Constraints (2g) and (2h) specify the binary decision variables.

5.1.2. Parking Inventory Optimization (PIO)

The objective of Parking Inventory Optimization (PIO) is to take x_{ij} and y_{kj} values of the VRO phase as an input and find the optimal relocation times (st_{ij}, st_{kj}). In this section, we relax the simplified assumption of Section 4 that parking stations do not have capacities and present a more realistic assumption. Carsharing organizations, in reality, have a predefined number of reserved parking stalls at designated parking stations. There is a higher incurred cost of storing any additional number of vehicles that surpasses the original number of reserved stalls. Note that this capacity is not the same as the total station capacity. For example a parking station may have 30 stalls where only 2 stalls are reserved by a CSO.

Fig. 4 presents reserved capacity at a station and the associated costs. The normal parking rate γ^1 increases to γ^2 when the number of occupied stalls are beyond reserved capacity.

The following constants and sets are used to formulate the PIO problem:

- W_p: set of every user where a vehicle is relocated from that user's destination to parking station p.
- L_p: set of every user where a vehicle is relocated from parking station p to that user's origin.
- S_p: set of all vehicles that are relocated from parking station p.
- H_p: set of all vehicles that are relocated to parking station p.
- I_p: total initial number of vehicles located at p.
- Ar^p_t: the number of users who arrive to parking station p before time t.
- De^p_t: the number of users who depart from parking station p before time t.
- cap_p: reserved capacity of parking station p.

Both Ar^p_t and De^p_t are easily obtained from the accepted set of users from the VRO phase. We denote total occupancy of parking station p at time t by $q_{pt}\left(=q^1_{pt}+q^2_{pt}\right)$ where q^1_{pt} is occupancy below capacity cap_p (allowable capacity of parking station p) and q^2_{pt} is any additional parking volume that violates the reserved capacity. Hence, q^2_{pt} can only be above zero when $q^1_{pt}=cap_p$.

To formulate the PIO problem, we discretize time and transform x_{ij} and y_{kj} of the VRO phase into $z^{p_1p_2}_{ijt}$ and $z^{p_1p_2}_{kjt}$ where $p_1=\{N_i, n_k\}$ and $p_2=M_j$. Decision variable $z^{p_1p_2}_{ijt}$ is a binary variable and is equal to 1 when a vehicle is relocated from P_1 (i.e., N_i) to p_2 (i.e., M_j) at time t. Decision variable $z^{p_1p_2}_{kjt}$ is a binary variable and is equal to 1 when a vehicle is relocated from p_1 (n_k) to p_2 (M_j) at time t.

We discretize time into T different segments of size μ where μT is the optimization period and formulate the problem using the concept of time–space networks as:

$$Min \sum_{t\in T}\sum_{p\in P}\left(q^1_{pt}\cdot\gamma^1_{pt}+q^2_{pt}\cdot\gamma^2_{pt}\right) \tag{3a}$$

$$0\leqslant q^1_{pt}\leqslant cap_p \tag{3b}$$

$$0\leqslant q^2_{pt} \tag{3c}$$

$$\sum_{k\in H_p}\sum_{t'=1:t-tr(n_k,M_j)}\left[z^{p_1p}_{kjt'}\right]-\sum_{k\in S_p}\sum_{t'=1:t}\left[z^{pp_2}_{kjt'}\right]+\sum_{i\in W_p}\sum_{t'=1:t-tr(N_i,M_j)}\left[z^{p_1p}_{ijt'}\right]-\sum_{j\in S_p}\sum_{t'=1:t}\left[z^{pp_2}_{ijt'}\right]+Ar^p_t-De^p_t+I_p\leqslant q^1_{pt}+q^2_{pt}\quad\forall p\in P,\ t\in T \tag{3d}$$

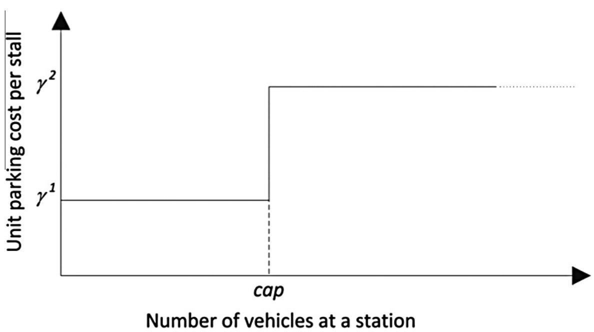

Fig. 4. Normal parking rate and increased parking rate when vehicles surpass capacity.

$$\sum_{t=1:A_i} z_{ijt}^{p_1p_2} = 0 \quad \forall i \in U \tag{3e}$$

$$\sum_{t=De_j-tr(N_i,M_j):T} z_{ijt}^{p_1p_2} = 0 \quad \forall j \in U \tag{3f}$$

$$\sum_{t=1:Av_k} z_{kjt}^{p_1p_2} = 0 \quad \forall kK \tag{3g}$$

$$\sum_{t=De_j-tr(n_k,M_j):T} z_{kjt}^{p_1p_2} = 0 \quad \forall j \in U \tag{3h}$$

$$\sum_t z_{ijt}^{p_1p_2} = 1 \quad \forall i \in U \tag{3i}$$

$$\sum_t z_{kjt}^{p_1p_2} = 1 \quad \forall k \in K \tag{3j}$$

$$z_{kjt}^{p_1p_2} = 0,1 \quad \forall k \in K,\ t \in T \tag{3k}$$

$$z_{ijt}^{p_1p_2} = 0,1 \quad \forall i,\ j \in U,\ t \in T \tag{3l}$$

The presented model is a binary integer programming model with a piecewise convex objective function (with the marginal costs shown in Fig. 4) which attempts to minimize the total cost of parking. Constraints (3b) and (3c) ensure that q_{pt} values are within the correct boundaries.

Constraints (3d) have seven terms on their left side. The first two terms present initial vehicle movements between a vehicle depot and a user and the second two terms present vehicle relocation movements between users. The first term is the total number of vehicles that are located from their initial location (p_1) to p, and the second term is the total number of vehicles that are relocated from their original location p to any other location (p_2). The third term is the total number of vehicles relocated from user i's destination (N_i) to parking p to serve user j ($p = M_j$) and the fourth term is the number of vehicles relocated from parking p where $p = N_i$ to user j's origin (N_j). The first and the third term in (3d) sum over the first to the $(t - tr(n_k, M_j))$th and $(t - tr(N_i, M_j))$th time increments, respectively, because that is how long it takes a vehicle to reach M_j. The next three terms in (3d) are explained in the notations provided above. Constraints (3e)–(3h) check for relocation time boundary violations which correspond to constraints (1f) and (1g). Constraints (3i) and (3j) ensure that every relocation process is initiated only once and constraints (3k) and (3l) are binary variable declarations.

One major factor in the discretization of the PIO model is the value of μ. A low value of μ would increase the number of nodes in the time–space diagram which leads to high computation times. On the other hand, high values of μ can make the problem infeasible in cases where μ is higher than the relocation time window of at least one relocation process ($\mu > De_j - tr(N_i, M_j) - Ar_i$). In such a case, no suitable time can be obtained for relocating a vehicle from N_i to M_j. High μ values can also lead to inefficient answers since some potential relocation start times will be neglected. We, therefore, present a meta-heuristic Particle Swarm Optimization (PSO) algorithm which treats relocation start times as continuous variables within allowable constraints ((3e)–(3h)), calculates q_{pt}^1 and q_{pt}^2 for different time segments, and obtains the objective function in (3a). The st values are then altered in future iterations of PSO to find better values of (3a). PSO has been applied in other transportation related problems including the vehicle routing problem (Tu et al., 2013; Xu et al., 2011), cross-dock scheduling (Soltani and Sadjadi, 2010), and block size optimization in container yards (Lee and Kim, 2010).

Particle Swarm Optimization (PSO), a population based stochastic optimization technique, shares many similarities with evolutionary computation techniques such as Genetic Algorithm (Eberhart and Kennedy, 1995). Analogous to GA, PSO initiates with a population of random solutions and evolves the position of the particles based on the location of current optimal particles, until a desirable one is found. Considering a N_x-dimensional problem (N_x vehicles are awaiting their relocation operation), the position of the τth particle is represented by $L_\tau = (l_{\tau1}, l_{\tau2}, \ldots, l_{\tau N_x})$, where $l_{\tau\sigma}$ is the σth dimension of the τth particle. In other words, every particle is made of relocation times *(st)*. In every iteration, each particle seeks solutions by moving in the problem space with a velocity vector presented as $Ve_\tau = (v_{\tau1}, v_{\tau2}, \ldots, v_{\tau N_x})$, where $v_{\tau\sigma}$ is the velocity of the τth particle in the σth dimension. In the search process, each particle keeps track of its own best solution G_{local} along with its location $G_\tau = (g_{\tau\sigma})_{\sigma=1:N_x}$, and the best solution of the entire swarm G_{global} accompanied by its location $G = (G_\sigma)_{\sigma=1:N_x}$. The velocity of every particle is updated in iteration $z + 1$, (where z denotes the iteration counter) based on G_{local} and G_{global}:

$$v_{\tau\sigma}(z+1) = w \times v_{\tau\sigma}(z) + c_1r_1(g_{\tau\sigma}(z) - l_{\tau\sigma}(z)) + c_2r_2(G_\sigma(z) - l_{\tau\sigma}(z)) \tag{4}$$

where w is the inertia weight, c_1 and c_2 are constant values set to 2 (Eberhart and Kennedy, 1995), while r_1 and r_2 are random numbers uniformly distributed in the interval $[0,1]$ (Abraham et al., 2006). For the purpose of guiding the particles effectively, the velocity in any iteration must be held within the interval $[-V_{max}, V_{max}]$ (Abraham et al., 2006). The inertia weight, developed to better balance explorations and exploitation, affects the convergence speed of the PSO through controlling the impact of the history of velocities on the current velocity (Abraham et al., 2006). Eberhart and Shi (2000) emphasize that initially setting the inertia weight to a large value and linearly decreasing it with time has a better performance than using a fixed value.

After updating the velocity values, the new particle locations are computed as:

$$l_{\tau\sigma}(z+1) = l_{\tau\sigma}(z) + v_{\tau\sigma}(z+1) \tag{5}$$

which need to be within the constraints ((3e)–(3h)).

5.2. Simulation

The two presented optimization models run consecutively in a discrete event simulation environment where an event is defined as the arrival of a user. We choose a discrete event model instead of a continuous simulation model because the user announcements are situated fairly apart. Therefore a continuous model would lead to extra unnecessary runs. However, the optimization formulations (VRO and PIO) can be easily embedded in a continuous simulation framework as well.

At every event (user arrival), the VRO and PIO models are executed which determine the accepted users, vehicle relocations, and start time of vehicle relocation operations to serve those users. A greedy approach would be to finalize requests (accept or reject them) as soon as they are placed. Therefore, those rejected will not be reconsidered even if later on they become beneficial. The appointed vehicles to accepted users would become unavailable until they reach the end of their tours. The presented simulation in this paper, however, follows a rolling horizon approach which does not finalize any requests until the simulation time passes each assigned vehicle's relocation time *(st)*. This strategy keeps the vehicles unassigned in case a more profitable user request arrives. The flowchart in Fig. 5 illustrates the rolling horizon approach and is interpreted in the following steps:

1. The algorithm is initiated by setting time t to V_1 which is the request time of the first user. Set availability time of every vehicle k (Av_k) to V_1.
2. A pool of the user requests is passed onto the VRO model and the PIO model. At the first iteration, V_1 is the only user in the pool.
3. The VRO finds the most beneficial user requests and relocates the vehicles to serve them.
4. The PIO model takes the results of the VRO and finds the optimal starting time for each relocation operation to minimize parking costs.
5. The next event is defined at the arrival time of the next user q ($t = V_q$).
6. Every user j for whom st_{ij} is smaller than t is eliminated from the user pool and the corresponding vehicle for that user is relocated to its origin location M_j. This is presented in Fig. 5 at the "Exit Users" step. The users who reach their departure time before t ($D_j < t$) and have not been assigned a vehicle are also terminated from the model and are considered unanswered demand.
7. Availability and location of relocated vehicles is updated. Available vehicles are then passed onto the vehicle pool along with their information such as location (n_k) and availability time (Av_k). To find the new the Av_k and n_k, the tour itinerary of vehicle k is checked to find the last user g where $st_{ig} < t$. Then, $Av_k = A_g$ and $n_k = N_g$ so that vehicle k is available once user g arrives at his/her destination (N_g) at his/her arrival time (A_g). The updating processing passes Av_k and n_k values to the VRO and I_p values to the PIO.
8. The termination condition checks if all the waiting users have announced their requests. If yes, the model terminates. If no, the user of that event (q) is passed onto the user pool.

5.3. Measures of Effectiveness (MOE)

We propose various Measures of Effectiveness (MOE) to evaluate the performance of the dynamic model. These MOEs are total revenue, total cost of relocation, fleet utilization, and system reliability. Total revenue and cost of relocation are the revenue obtained from users and the required relocation costs to generate those revenues, respectively. Fleet utilization is the ratio of the number of vehicles used over the available number of vehicles over the course of the entire day. A high number of available vehicles with a low demand lead to a low fleet utilization index. System reliability is the total number of accepted users over total demand. This index is between 0 (no user is served) and 1 (all users are served).

6. Example problem: the case of Autoshare

Autoshare, founded in 1998, is a Toronto based carsharing company with 209 parking locations across the city (Fig. 6), more than 10,000 users, and an average of 200 vehicle requests a day (Constain et al., 2012). Autoshare has a flexible policy on vehicle reservation where users can book a vehicle on the spur-of-the-moment or up to a year in advance. The service has no restrictions on how far a vehicle can be driven from the origin parking station but requires that the users return the vehicle to its original location. In this section, we use the case of Autoshare to evaluate the performance of the benchmark model

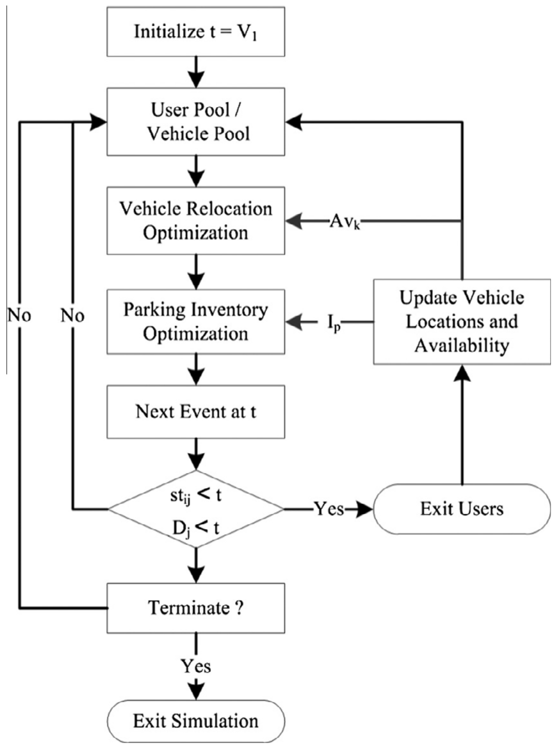

Fig. 5. Flowchart of the simulation model.

presented in Section 3. Autoshare is a potential candidate for a one-way system since its parking stations are fairly closely spaced with a mean distance of 5.48 km which we expect would lead to reasonable vehicle relocation cost.

Autoshare reserves up to 3 parking stalls in the parking stations presented in Fig. 6. A total fleet of 224 vehicles are located at these stations to serve the users. However, the company does not currently have any vehicle relocation policies because (i) there are already more vehicles in the system than the daily demand (200 user/day) and (ii) users are obliged to return vehicles to the original pick up location. Such two-way system return policy restrictions are a source of discomfort for the users which motivates CSOs such as Communato to launch new one-way carsharing systems.

We have used all available Autoshare data in our analysis except station capacity because Autoshare by nature is a two-way system which currently does not have any rigorous relocation policy. Therefore it must have a large fleet distributed between its many stations to answer the demand. Our analysis in the next sections, on the other hand, shows that fleet size can be majorly reduced through relocation by up to 86%. Therefore Autoshare currently has higher capacity than would be required (if vehicle relocation was allowed) and consequently reserves more parking stalls than is necessary. For this reason we have chosen to assume one reserved parking stall per station.

6.1. Benchmark model analysis

To evaluate the benchmark model we execute it 200 times for each of four different demand scenarios (50, 100, 150, and 200 users/day). Every model run is executed using a different set of user departure times obtained randomly from a curve fit to the observed data within the range of 12:00 PM to 9:00 PM (peak hours of the service) for a typical weekday. The origin/destination of the users is obtained randomly from choice probabilities calculated from the observed data. The probability of a parking station p being chosen as an origin (destination) is the number of times it was chosen as an origin (destination) over the total number of requests in a day. The arrival time at the destination is calculated as departure time plus the travel time from origin to destination. Travel times are computed by dividing an average velocity of 20 km/h by Manhattan distances between the parking stations. We assume a request time threshold (reservation time) of 30 min. However, reservation times can also be randomly generated based on any given data.

Fig. 7 illustrates the results of the simulation showing a trade-off between the hours of relocation and fleet size. The model runs with a lower fleet size are accompanied with higher hours of vehicle relocation and vice versa. Both fleet size and relocation hours increase with higher demands.

A common notion in the carsharing literature, which stems from aggregate treatment of demand, is that a specific fleet size is needed to answer all or a portion of the demand (Kek et al., 2009; Correia and Antunes, 2012). Fig. 7 and Table 2 show

Fig. 6. Parking locations of Autoshare (Toronto, Ontario).

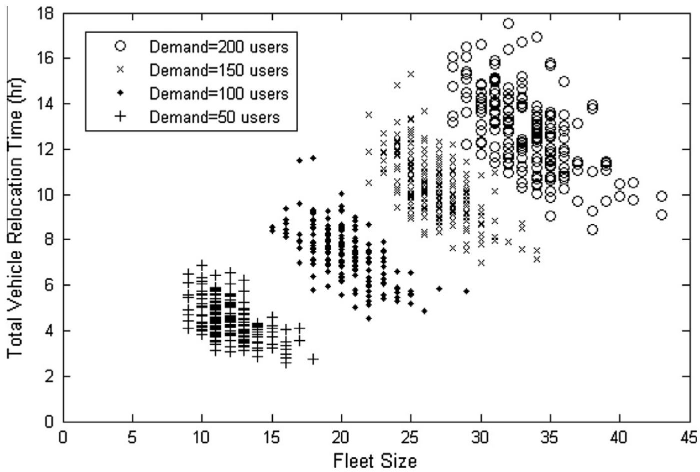

Fig. 7. Benchmark model fleet size and relocation time for four demand scenarios (200 runs/scenario).

that the required fleet size is a function of demand. Larger fleets are generally needed in cases of higher demand, where user announcements are more closely spaced and vehicle relocation is challenging. On the contrary, in cases of lower demand the system can relocate a smaller fleet to serve all users. Suitable fleet size for every scenario, therefore, also varies depending on the announcement patterns of the users. For example, if too many requests are placed around the same time then the CSO would need many vehicles since it would be difficult to relocate a smaller number of vehicles while answering the entire demand. However, if the requests are placed fairly apart in time then there is more time for the CSO to relocate the vehicles with a smaller fleet size.

The decreasing ratio of vehicles per user in the last row of Table 2 shows increasing economies of scale as the demand increases. Longer reservation time policies give CSOs more time to relocate the vehicles between the parking stations but make the system inconvenient for users who need immediate access. To assess the importance of this component, we run the benchmark model 30 times for different reservation time policies with the demand of 200 users/day. Fig. 8 depicts the impact of reservation time on the average fleet size and total relocation time where the fleet size decreases and relocation time increase with higher reservation time values. The figure illustrates that fleet size stays roughly the same above 30 min of reservation time. This is dependent on the proximity and locations of the parking stations. In 30 min of reservation time, vehicles can be easily transferred between any two parking stations. Furthermore, given that the cost of acquiring a vehicle is assumed to be much higher than vehicle relocation, the CSO incurs the highest cost at a reservation time of 0 min.

6.2. Dynamic model analysis

In this section, we assess the dynamic model and emphasize the importance of the price rate for users (Pr) and the cost rate for relocations (Cr) parameters. The dynamic model is set up using the same parameters from the benchmark model

Table 2
Benchmark model fleet size for four demand scenarios.

	Demand (users/day)			
	50	100	150	200
Maximum fleet size	18	29	34	43
Mean fleet size	12.08	20.24	26.79	33.36
Minimum fleet size	9	15	22	28
Variance of fleet size	2.97	4.95	5.42	8.41
Mean fleet size/demand	0.24	0.20	0.18	0.16

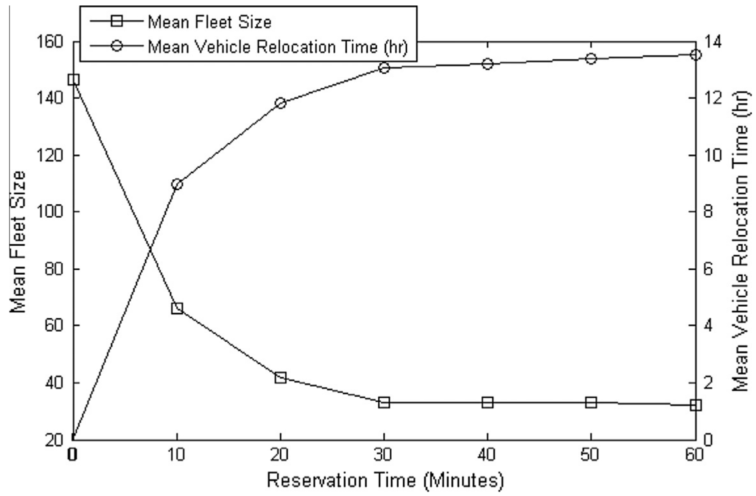

Fig. 8. Benchmark model mean fleet size and relocation time for different reservation time policies with a demand of 200 users/day.

with the following exceptions and additions. The initial location of each vehicle is randomly chosen from the 209 parking stations. No two vehicles are initially located at the same parking station. The parking capacity of each parking station is set to 1 with $\gamma^1 = 0$ and $\gamma^2 > 0$. The population size of the PSO for solving the PIO is set to 10, the number of iterations to 8, the initial inertia weight to 1.2, the final inertia weight to 0.4, and C_1 and C_2 values to 2.

Pr and *Cr* highly impact the results of the dynamic model. If *Cr* is noticeably higher than *Pr*, the CSO is not willing to relocate its vehicles. Conversely, with low *Cr* values, the CSO is indifferent to any number of relocation operations. Table 3 presents various measures of effectiveness at different *Pr* values with a demand of 200 users/day when $Cr = 1$. We only consider the ratio of *Pr* to *Cr* in the analysis because in objective function (2a) the least common multiples can be removed. Each scenario is run 30 times. Results show that hours of relocation, revenue, and reliability increase with higher *Pr* values. Revenue hours in Table 3 is the sum of total travel time of the users without multiplying by *Pr*. All three mentioned measures of effectiveness grow faster at lower *Pr* values and reach relatively constant values at higher *Pr* values. This shows that higher *Pr* values lead to serving users who require high relocation costs but offer less revenue to the system. The fairly constant Fleet Utilization index shows that all vehicles are used at every *Pr* Scenario. Note that Fleet Utilization is a measure that shows whether a company owns more vehicles than it really needs. For example a Fleet Utilization value of 0.94 in Table 3 implies that 16% of the vehicles are extra. However, this measure also considers vehicles that have been used for a very short time (say 5 min). Therefore, the summation of the Relocation hours and Revenue hours would show the utilization hours of the vehicles.

Table 3
Measures of effectiveness at various ratios of *Pr* to $Cr = 1$ where demand is 200 users/day.

Pr/*Cr*	Relocation hours	Revenue hours	Reliability	Fleet utilization
0.5	3.42	9.46	0.80	0.94
1.0	7.53	11.29	0.86	0.98
1.5	9.89	14.04	0.93	1.00
2.0	11.34	15.49	0.95	1.00
2.5	12.57	16.23	0.96	1.00
3.0	13.39	16.75	0.97	1.00

Table 4
Relocation hours of the benchmark model (left) and the dynamic model with $Pr = 3$, $Cr = 1$ (right).

	Benchmark model				Dynamic model			
	Demand (users/day)				Demand (users/day)			
	50	100	150	200	50	100	150	200
Maximum	6.89	11.59	15.28	17.51	7.21	12.35	16.11	18.62
Mean	4.45	7.45	10.33	12.78	4.78	8.01	10.91	13.39
Minimum	2.58	4.56	6.99	8.46	2.82	4.69	7.31	9.14
Variance	0.70	1.39	2.12	3.05	1.04	2.03	2.82	3.34

Test of hypothesis do not imply that benchmark and the dynamic models are significantly different with 90% level of confidence. The *t*-stat values for the four scenarios are 1.37, 1.65, 1.42, and 1.32, respectively.

Table 4 compares the relocation hours in the benchmark model and the dynamic model for a *Pr* value of 3 and a demand of 200 users/day. The highest *Pr*/*Cr* value ensures that all users have to serviced and therefore mainly minimizes the relocation costs. We choose this ratio because the dynamic model becomes more like the benchmark model at higher *Pr* to *Cr* ratios where Fleet Utilization becomes 1 and Reliability gets closer to 1. Furthermore, note that the comparisons in Table 4 are made for a special case when (i) the number of vehicles in both the VRO model and the benchmark model are the same and (ii) the *Pr*/*Cr* ratio is relatively high in the VRO model. These two conditions ensure that total relocation time comparison is valid between the two models.

The number of vehicles in the dynamic model is set to the average number of vehicles in the benchmark model shown in Table 2 for each demand scenario. Table 4 shows that the relocation hours for the benchmark model are lower than those of the dynamic model for all levels of user demand. The better performance of the benchmark model is expected due to the availability of complete information in the model. However, the differences are not significantly different with 90% confidence.

The dynamic model has higher variances because many relocation decisions are made iteratively whereas the benchmark model makes relocation decisions in only one iteration. The larger number of iterations makes the dynamic model more dependent on the user request patterns. Table 4 shows a lower percentage difference in the mean of the hours of relocation as demand increases because there are better relocation opportunities within higher demands.

The benchmark and the dynamic model can easily be operationalized with low computation times. The models were coded in CPLEX using Matlab 7.12 running on a dual core 2.40 GHz laptop computer with 8 GB RAM. The benchmark model with a demand of 200 users/day runs in 12 s. The dynamic model, with the same demand, runs in 34 s. The dynamic one-way system with 400 users, however, runs in 3.41 min. Therefore, although the model performs efficiently for the proposed case study, meta-heuristics can be used to reduce the computational burden for larger problems.

7. Conclusions

This paper proposes a benchmark and a dynamic model for the carsharing operations problem and evaluates the performance of the one-way system through various measures of effectiveness. The benchmark model is set up with complete knowledge of user requests whereas the more realistic dynamic model receives user information only when they make a request. The dynamic model performs satisfactorily when compared to the benchmark model. The presented dynamic model can benefit CSOs in building robust decision support systems and can lead to more successful one-way carsharing systems. Furthermore, the dynamic model is more practical than the conventional aggregate models in the literature because it is flexible with different demand scenarios (request patterns) and it can adapt to daily circumstances such as absence of a vehicle due to maintenance.

This study also shows the importance of various policies such as extension of the required reservation time on reducing the fleet size of the one-way carsharing systems. Increasing the reservation time from zero (open-ended system) to 30 min can reduce the fleet size by 86%. However, although any increase in the reservation time can lead to a lower fleet size and lower costs it can also make the system less attractive to the users and eventually lead to lower demands. This, however, is more complicated to model and needs further research.

Results show a tradeoff between vehicle relocation hours and fleet size. A higher fleet size requires less total relocation hours and vice versa. Both relocation hours and fleet size, however, increase with higher demands. In addition to the value of demand, both the fleet size and relocation hours also depend on the schedules of the users (request patterns). Fleet size decreases when requests are spread out in time.

The benefit of the dynamic model depends on the ratio of price charged per unit time (*Pr*) and cost of relocation per unit time (*Cr*). In cases where *Pr* is relatively higher than *Cr*, it is more beneficial to service more customers. This leads to higher reliability of the system which leads to customer satisfaction. Therefore, it is critical to cut down unit relocation costs as much as possible. For future research, we believe that consideration of different fleet types (e.g., electric), their cost, and their environmental impact is valuable to CSOs. Presentation of actual monetary costs can also provide more insight. Furthermore, accounting for flexibility of the CSO is assigning each user to a specific station from a set of stations that are in the vicinity of that user can help reduce relocation costs even more. This topics is also of interest to the carsharing community.

Acknowledgements

The authors wish to acknowledge the Canadian Automobile Association and National Science and Engineering Research Council for financial support on this project. We also thank Autoshare Car Sharing Network Inc. for providing data that were used in support of the project.

References

Abraham, A., Guo, H., Lio, H., 2006. Swarm intelligence: foundations, perspectives and applications. In: Nedjah, N., Mourelle, L.M. (Eds.), Swarm Intelligent Systems. Springer, Netherlands, pp. 18–25.

Barth, M., Todd, M., 1999. Simulation model performance analysis of a multiple station shared vehicle system. Transport. Res. C: Emerg. Technol. 7 (4), 237–259.

Celsor, C., Millard-Ball, A., 2007. Where does carsharing work? Using geographic information systems to assess market potential. Transport. Res. Rec.: J. Transport. Res. Board, No. 1992, 61–69, Transportation Research Board of the National Academies.

Cervero, R., Golob, A., Nee, B., 2007. City CarShare: longer term travel demand and car ownership impact. Transp. Res. Rec. 1992, 70–80.

City Carshare, 2013. City Carshare 2013 Fact Sheet. <https://www.citycarshare.org/wp-content/uploads/2013/04/1071_CCS_Fact_Sheet_v2-1.pdf> (accessed 10.05.13).

Communauto, 2013. Communauto launches Auto-mobile: first 100% electric one-way carsharing pilot project in Canada. <http://actualites.communauto.com/en/2013/06/16/communauto-launches-auto-mobile/> (accessed 10.05.13).

Constain, C., Ardon, C., Habib, K.N., 2012. Synopsis of users' behavior of a carsharing program: a case study in Toronto. Transport. Res. A: Policy Pract. 46, 421–434.

Correia, G., Antunes, H.A.P, 2012. Optimization approach to depot location and trip selection in one-way carsharing systems. Transport. Res. E: Logist. Transport. Rev. 48, 233–247.

Correia, G.H., Jorge, D., 2013. The added-value of accounting for user's flexibility and information on the potential of a station-based one-way carsharing system: an application in Lisbon, Portugal. J. Intell. Transport. Syst.: Technol. Plann. Oper. (in press).

Eberhart, R.C., Kennedy, J., 1995. A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science. IEEE Press, Piscataway, NJ, pp. 39–43.

Eberhart, R.C., Shi, Y., 2000. Comparing inertia weights and constriction factors in particle swarm optimization. Proceedings of the IEEE International Congress on Evolutionary Computation, vol. 1. IEEE Press, Piscataway, NJ, pp. 84–88.

El Fassi, A., Awasthi, A., Viviani, M., 2012. Evaluation of carsharing network's growth strategies through discrete event simulation. Expert Syst. Appl. 39, 6692–6705.

Fan, W., Machemehl, R.B., Lownes, N.E., 2008. Carsharing dynamic decision-making problem for vehicle allocation. Transp. Res. Rec. 2063, 97–104.

Firnkorn, J., Muller, M., 2011. What will be the environmental effects of new free-floating car-sharing systems? The case of car2go in Ulm. Ecol. Econ. 70, 1519–1528.

Habib, K.M.N., Morency, C., Islam, T., Grasset, V., 2009. Modelling user's behaviour of a carsharing program in Montreal: application of zero inflated dynamic ordered probit model. In: Presented at 12th International Association of Travel Behaviour Research (IATBR) Conference, Jaipur, India.

Habib, K.M.N., Morency, C., Islam, T., Grasset, V., 2012. Modelling user's behaviour of a carsharing program in Montreal: application of zero inflated dynamic ordered probit model. Transp. Res. A 46, 241–254.

Jorge, D., Correia, G.H., 2013. Carsharing systems demand estimation and defined operations: a literature review. Eur. J. Transport Infrastruct. Res. 13 (3), 201–220.

Jorge, D., Correia, G.H., Barnhart, C., 2012. Testing the validity of MIP approach for locating carsharing stations in one-way systems. Procedia – Soc. Behav. Sci. (54), 138–148.

Kara, I., Bektas, T., 2006. Integer linear programming formulations of multiple salesman problems and its variations. Eur. J. Oper. Res. 3, 1449–1458.

Katzev, R., 2003. Car sharing: a new approach to urban transportation problems. Anal. Soc. Issues Public Policy 3 (1), 65–86, <www.asap-spssi.org/pdf/katzev.pdf> (accessed 01.03.13).

Kek, A.G.H., Cheu, R.L., Meng, Q., Fung, C.H., 2009. A decision support system for vehicle relocation operations in carsharing systems. Transportation Research Part E: Logistics and Transportation Review 45 (1), 149–158.

Lee, B. Kwon, Kim, K., 2010. Optimizing the block size in container yards. Transport. Res. E: Logist. Transport. Rev. 46 (1), 120–135.

Morency, C., Habib, K.M.N., Grasset, V., Islam, T., 2010. Understanding members' carsharing (activity) persistency by using econometric model. J. Adv. Transport. http://dx.doi.org/10.1002/atr.142.

Shaheen, S.A., Cohen, A.P., 2007. Growth in worldwide carsharing – an international comparison. Transp. Res. Rec. 1992, 81–89.

Shaheen, S.A., Cohen, A.P., Roberts, J.D., 2006. Carsharing in North America – market growth, current developments, and future potential. Transp. Res. Rec. 1986, 116–124.

Soltani, R., Sadjadi, S.J., 2010. Scheduling trucks in cross-docking systems: a robust meta-heuristics approach. Transport. Res. E: Logist. Transport. Rev. 46 (5), 650–666.

Steininger, K., Vogl, C., Zettl, R., 1996. Car-sharing organizations: the size of the market segment and revealed change in mobility behavior. Transp. Policy 3 (4), 177–185.

Stillwater, T., Mokhtarian, P.L., Shaheen, S.A., 2009. Carsharing and the built environment: geographic information system-based study on one US operator. Transp. Res. Rec. 2110, 27–34.

Tu, W., Li, Q., Shaw, S., Chen, B., 2013. A bi-level Voronoi diagram-based metaheuristic for a large-scale multi-depot vehicle routing problem. Transport. Res. E: Logist. Transport. Rev. 61, 84–97.

Vine, S., Sivakumar, A., Polak, J., Lee-Gosselin, M., 2013. The market and impacts of new types of carsharing systems: case study of Greater London. Transportation Research Board Annual Meeting 2013, Paper.

Xu, J., Yan, F., Li, S., 2011. Vehicle routing optimization with soft time windows in a fuzzy random environment. Transport. Res. E: Logist. Transport. Rev. 47 (6), 1075–1091.