Incentivization Schemes for Vehicle Allocation in One-Way Vehicle Sharing Systems

Alexandros Angelopoulos
School of Electrical & Computer Engineering
National Technical University of Athens, and
Computer Technology Institute & Press "Diophantus"
Patras, Greece

Damianos Gavalas
Dept. of Cultural Technology and Communication
University of Aegean, and
Computer Technology Institute & Press
"Diophantus", Patras, Greece

Charalampos Konstantopoulos
Dept. of Informatics
University of Piraeus, and
Computer Technology Institute & Press
"Diophantus", Patras, Greece

Damianos Kypriadis
Dept. of Informatics & Telecommunications
University of Athens and
Computer Technology Institute & Press
"Diophantus", Patras, Greece

Grammati Pantziou
Dept. of Informatics, Technological
Educational Institute of Athens, and
Computer Technology Institute & Press
"Diophantus", Patras, Greece

***Abstract*—The major operational problem in one-way vehicle sharing systems is the vehicle stock imbalance problem. In this paper, we address this problem by proposing a new approach for dynamically allocating vehicles to users based on "incentivization" schemes which use reservations to coordinate supply-and-demand mismatches and price incentives for rewarding users, if they accept to pick up their vehicle from an oversupplied station and/or to drop off it to an under-supplied station. The system incentivizes users based on the priorities of vehicle relocations from station to station, taking into account the fluctuating demand for vehicles and parking places at different stations over time. We present two different schemes for incentivizing users to act in favour of the system. Both schemes consider budget constraints and are truthful and budget-feasible. We extensively evaluated our approach through simulations and observed significant improvements in the number completed trips and system revenue.**

I. Introduction

Vehicle sharing is a model of short-term vehicle rental, particularly attractive to citizens who make only occasional use of a vehicle, enabling the benefits of private vehicles without the costs and responsibilities of ownership [12]. Replacing private automobiles with shared ones directly reduces demand for parking spaces and decreases traffic congestion at peak hours, thereby supporting the vision of sustainable transportation. Vehicle sharing first appeared in Europe in 1940's [15], [16]. Vehicle Sharing Systems (VSSs) are commonly classified into two-way and one-way systems. Two-way systems require users to pick up the vehicles from and return the vehicles to the same station while one-way systems permit users to return the vehicles to a different station. The major operational problem in one-way systems which does not appear in two-way systems, is the vehicle stock imbalance taking place when a large number of vehicles is gathered to certain stations of the system while high demand stations remain without the appropriate number of vehicles to satisfy user requests.

Significant research work has been conducted from an optimization perspective to devise methods to overcome the vehicle stock imbalance problem in VSSs [7]. A very common approach proposed in the literature to address the problem is vehicle relocation, i.e., to employ drivers (or staffed trucks in the case of bicycles) to relocate vehicles to high demand stations. The proposed vehicle relocation approaches are separated into static and dynamic ones. In the static approaches, repositioning is taking place when the system is not operating (e.g. during the night) [14]. Although such an off-line approach gives the possibility of employing optimal low cost routing methods for transferring vehicles among stations, it cannot react to unexpected variations in the demand pattern arising during the operation of the system. In the dynamic approaches, relocation is taking place while the system is in full operation and therefore, reacts on-line to unforeseen changes in the demand pattern [13].

Vehicle relocation methods address successfully the vehicle stock imbalance problem but increase significantly the operational cost of the VSSs as they have to account for the relocation cost. Note that the relocation cost consists of the vehicle cost related to the distance driven to relocate the vehicles and the labour cost of the drivers used to relocate the vehicles. Three important alternative approaches to address vehicle stock imbalance have been proposed in the literature. The first approach is a user-based relocation strategy which employs price incentives for grouping users (trip-joining strategy) if they are travelling from a station with a shortage of vehicles or grouping users (trip-splitting strategy) if they are travelling from a station with a surplus of vehicles [1]. The second approach is based on trip selection, i.e., vehicles are allocated to satisfy user requests only if this is advantageous to the system from the profit point of view taking into consideration the vehicle relocation costs [3], [4].

The third approach employs pricing policies to encourage users to pick up their vehicle from an oversupplied station and/or to drop off it to an under-supplied station. Regarding this approach different methods have been proposed. Specifically, in [5] a mean-field technique is used to analyse the effect of simple incentive schemes on the service level provided by the sharing system. It is shown that incentives to return bikes to the least loaded station among two, may significantly improve the system's performance even if only a small proportion of the users accept to do so. In [18] the VSS is modelled as a closed queuing network with infinite buffer capacity and Markovian demands. The goal is to maximize utilization by setting prices and incentives for each possible trip. In [10] a MINLP model is proposed that considers demand as a function of price and searches for the prices that maximize the profit. In [13] a mechanism is given to compute dynamic price incentives encouraging users to choose another drop-off station, thereby reducing the expected cost of vehicle relocation using dedicated staff. The mechanism is based on a predictive model of the expected near-future evolution of the system state. In [17] the authors design a dynamic pricing mechanism using the approach of regret minimization in online learning.

In this paper we propose a new approach for dynamically allocating vehicles to users based on "incentivization" schemes which use (i) reservations to coordinate supply-and-demand mismatches, reduce uncertainty and allow better forecast of the VSS's future state, and (ii) price incentives for rewarding users if they are travelling from a station with a surplus of vehicles to a station with a shortage of vehicles. Briefly, our approach proceeds as follows. Upon receiving a set of user requests, for each request asking to move from a start location to an end location, the system first determines a set of alternative trips with different vehicle pick-up station and/or a drop-off station (the set of *feasible trips*) than the requested trip. These are the trips the user is willing to accept, as long as they participate in the incentivization scheme and according to their declared tolerances. Then each feasible trip is broken down into exact vehicle relocations, each consisting of an available vehicle at the pick-up station, an available parking place at the drop-off station and the corresponding pick-up and drop-off times. At this point, there is a set of candidate relocations V to materialize the set of requests, from which the system may select the subset which maximizes its payoff.

To accomplish this, we assign a weight (*relocation priority*) to each potential relocation to indicate the urgency in performing it. The relocation priority is a function of the pick-up and drop-off times and stations, as well as of the exact pair of available vehicle and available parking place to be used. In the sequel, we propose a method for computing an optimal set of trip suggestions that maximizes the profit of the system, i.e., the number of accepted requests taking into account the relocation priorities, the fluctuating demand for vehicles over time, the system budget constraints, and also the strategic behaviour of the users potentially aiming at maximizing their profit. We present two effective incentivization schemes to associate trip suggestions with price incentives aiming at encouraging users to accept the suggested trips. The first scheme is not taking any user cost associated with the change of his route, while the second scheme considers this cost. Both schemes are truthful and budget-feasible.

Unlike previous related works, our incentivization schemes examine the user requests not individually but at batches and assign priorities to the requests by considering occupancies of pairs of stations instead of individual occupancies of the pick-up and drop-off stations together with a detailed view of vehicle distribution over VSS. Thus, our schemes are able to take informed decisions and hence derive effective user incentives. The rest of the paper is organized as follows. Section II presents the incentivization schemes for vehicle allocation and Section III discusses how the relocation priorities are set. Section IV presents the experimental evaluation results.

II. THE INCENTIVIZATION SCHEMES FOR VEHICLE ALLOCATION

Let $S = \{s_1, ..., s_n\}$ be the set of the stations of the VSS. We associate each station $s_i \in S$ with the following variables:

- c_i: s_i's capacity, i.e., the number of parking spaces at s_i;
- $o_i(t)$: s_i's anticipated (planned) occupancy at time t; the value of $o_i(t)$ can be computed at a time $t' < t$ using the system's information about the confirmed vehicle reservations concerning s_i up to t';
- $w_i(t)$: s_i's target (intended) occupancy at time t. It is mainly computed using historical data for vehicle reservations over long periods of time that may concern different hours of a day, different days of the week or different seasons (see [6], [9], [11] for possible approaches).

In the sequel we shall use the notation, $s_i^{\alpha}(t)$, $\alpha \in \{1, ..., o_i(t)\}$ to refer to a vehicle available at station s_i and time t, and $s_i^{\epsilon}(t)$, $\epsilon \in \{1, ..., c_i - o_i(t)\}$ to refer to a free parking space at station s_i and time t, for $i \in \{1, ..., n\}$.

We consider that each user request for vehicle reservation R_i comprises the following attributes:

- $sl(R_i)$: the exact start trip location;
- $el(R_i)$: the exact end trip location;
- $st(R_i)$: the user start trip time;
- $dt(R_i)$: the rental time duration (optional);
- $icv(R_i)$: indication of willingness in participating in the incentivization scheme (optional);
- $delay(R_i)$: the maximum additional time the user is willing to spent for an alternative trip with respect to the user's best trip induced by R_i (optional);
- $tol(R_i)$: the maximum acceptable (tolerable) time to be spent for the non-vehicle-use part of the user's trip (i.e., the maximum walking time the user may tolerate and/or the maximum time the user is willing to spent using public transport) (optional).

Note that the values of the last four attributes are kept at the user's profile. A request contains values to these attributes only when the user wants to differentiate from her profile kept by the VSS.

A. Handling user requests

Upon receiving a user request R_i, the VSS first determines a set of alternative trips called the set of *feasible trips* T^{R_i}.

This set is constructed taking into account walking time and/or multimodal public transportation data for estimating the elapsed time to reach a vehicle pick-up station from $sl(R_i)$ and to reach $el(R_i)$ from a vehicle drop-off station. Specifically, for each R_i the set of *reachable vehicle station pairs* B^{R_i} is computed which consists of potential pairs of vehicle pick-up and drop-off stations $(s_p, s_d) \in S \times S$ that support the feasibility of the request.

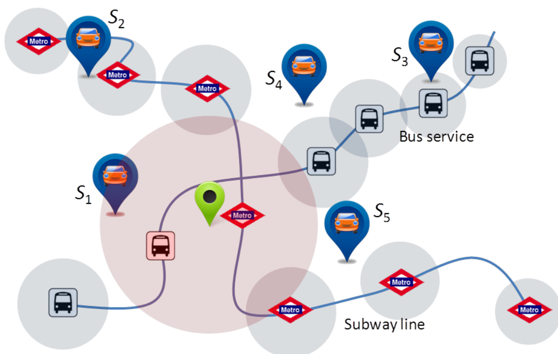

Fig. 1. Set of reachable stations from $sl(R_i)$; the red circle area is within walking distance from $sl(R_i)$, while the blue circle areas are reachable using public transport

Notice that the user's total trip time $\tau_p^d(R_i)$ for materializing request R_i using a vehicle from station s_p to station s_d can be broken down into three parts:

$$\tau_p^d(R_i) = \tau\big(sl(R_i) \to s_p\big) + \tau\big(s_p \to s_d\big) + \tau\big(s_d \to el(R_i)\big),$$

that is the time $\tau\big(sl(R_i) \to s_p\big)$ needed to move from $sl(R_i)$ to station s_p, the time $\tau\big(s_p \to s_d\big)$ needed to drive from s_p to s_d, and finally, the time $\tau\big(s_d \to el(R_i)\big)$ needed to reach $el(R_i)$ from s_d. Therefore, the set of *reachable vehicle station pairs* B^{R_i} consists of the pairs (s_p, s_d) which satisfy the following:

$$\tau\big(sl(R_i) \to s_p\big) + \tau\big(s_d \to el(R_i)\big) \leq tol(R_i)$$
$$\tau_p^d(R_i) \leq \tau_{p^*}^{d^*}(R_i) + delay(R_i),$$

where p^* is the index of the station s_{p^*} which is closer to $sl(R_i)$ than any other station $s_x \in S$, and d^* is the index of the station s_{d^*} which is closer to $el(R_i)$ than any other station $s_y \in S$ (see Figure 2). In the sequel, we refer to the pair (s_{p^*}, s_{d^*}) as the *user's best* pair.

Each R_i corresponds to a set of *feasible trips* $T^{R_i} = \{tr(s_p, s_d, t_p, t_d) \mid (s_p, s_d) \in B^{R_i}\}$, where each trip $tr(s_p, s_d, t_p, t_d) \in T^{R_i}$ is *uniquely characterized by* the pair of stations (s_p, s_d) between which the vehicle relocation takes place and the timestamps t_p, t_d when the vehicle is taken from s_p and left at s_d respectively. If a user declares unwilling to participate to the incentivization scheme, then the set B^{R_i} consists only of the user's best pair (s_{p^*}, s_{d^*}) and thus, the request corresponds to only one feasible trip called the *user's best* trip from s_{p^*} to s_{d^*}.

B. The vehicle allocation mechanism

Given a set of requests $R = \{R_1, R_2, ..., R_k\}$ submitted to the VSS during a predetermined time window, for each $R_i \in R$ we compute the corresponding set of feasible trips $T^{R_i} = \{tr(s_p, s_d, t_p, t_d) \mid (s_p, s_d) \in B^{R_i}\}$, $i = 1, .., k$. We aim at computing an optimal set of trip suggestions M that maximizes the profit of the system, i.e., the number of accepted user requests taking into account the priorities of the vehicle relocations realized by the user requests as well as the budget the system can afford to spend for rewarding users if they accept the suggested trips, and the strategic behaviour of the users potentially aiming at maximizing their profit. .

To formulate the problem, we proceed as follows. For each $R_i \in R$ and feasible trip $tr(s_p, s_d, t_p, t_d)$ in T^{R_i} if $s_p^\alpha(t_p)$ is a vehicle available at s_p at time t_p and $s_d^\epsilon(t_d)$ is an empty parking place at s_d at time t_d, then $\phi = \big(R_i, s_p^\alpha(t_p), s_d^\epsilon(t_d)\big)$ is considered as a *candidate trip suggestion* for serving R_i. Then $\phi\big(R_i, s_p^\alpha(t_p), s_d^\epsilon(t_d)\big)$ is assigned a *weight*, which is equal to the relocation priority π of the vehicle $s_p^\alpha(t_p)$ to the parking place $s_d^\epsilon(t_d)$, calculated as a function of s_p, s_d, t_p, t_d (see Section III).

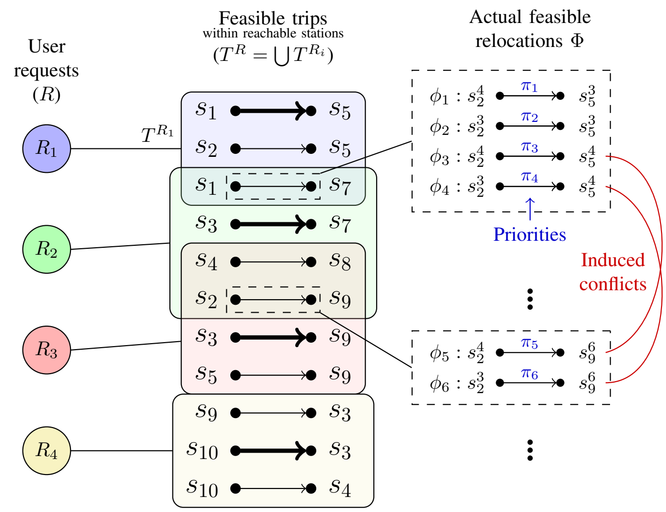

Fig. 2. The vehicle allocation problem

Now, for each $R_i \in R$, let C_i be the set of all candidate trip suggestions $\big(R_i, s_p^\alpha(t_p), s_d^\epsilon(t_d)\big)$ each weighted as described previously. Let also V be the union of the sets C_i over all $R_i \in R$. Then a solution to our problem is a subset M of trip suggestions in V which satisfy different requests, require non-conflicting resources and maximize the sum of the priorities of the corresponding feasible trips. Specifically, note that for each C_i only one of the candidate suggestions may be included in the problem solution as they all correspond to the same user request R_i. Similarly, although the same vehicle (or the same parking place) may appear to more than one candidate suggestion in V, it can be used to satisfy only one request (see Figure 3). Notice that our problem can be considered as a WEIGHTED 3-SET PACKING problem on the set V. By solving it we find a subset M of V consisting of mutually disjoint triples (i.e., for any two distinct triples $\big(R_i, s_p^\alpha(t_p), s_d^\epsilon(t_d)\big)$, $\big(R_i', s_{p'}^{\alpha'}(t_{p'}), s_{d'}^{\epsilon'}(t_{d'})\big) \in M$, we have $R_i \neq R_i'$ and $s_p^\alpha(t_p) \neq s_{p'}^{\alpha'}(t_{p'})$ and $s_d^\epsilon(t_d) \neq s_{d'}^{\epsilon'}(t_{d'})$), which maximizes the sum of the priorities of the corresponding feasible trips. The triples in the set M correspond to requests that can be concurrently served as they require non-conflicting resources.

In the sequel we employ a heuristic algorithm to solve the WEIGHTED 3-SET PACKING problem on the set V. The algorithm considers the intersection graph $G = (V, E)$ of the set system V containing a node for each triple (candidate suggestion) in V and an edge between any two nodes that represent intersecting sets. The weight of each node in G is set equal to the calculated priority of the corresponding triple in V. Actually, the WEIGHTED 3-SET PACKING problem is reduced to the MAXIMUM WEIGHT INDEPENDENT SET (MWIS) problem on $G = (V, E)$. Notice that G is a 4-claw free graph, since the sets are of size 3[1]. The *ComputeTripSuggestions* algorithm (Algorithm 1) given below, is a constant factor approximation algorithm for solving the MWIS problem in our vertex-weighted 4-claw free graph G. The algorithm first orders the nodes of V in decreasing order of their relocation priorities π and then employs a greedy approach for computing the MWIS of G, i.e., the set M of *trip suggestions* that can be simultaneously served by the VSS. In Algorithm 1, if A, B are subsets of V then $N(A, B) = \{v \in B : \exists u \in A \ \ such \ that \ \ \{v, u\} \in E \ \ or \ \ v = u\}$.

In the sequel, we need to associate each trip suggestion in M with a price incentive aiming at encouraging users to accept the suggested trips.

Algorithm 1 ComputeTripSuggestions

Input: Node-weighted 4-claw-free graph $G(V, E, \pi : V \to \mathbb{R})$
Output: The Set of Trip Suggestions M

```
Order nodes in V in decreasing order of π;
M ← ∅;
for all φ ∈ V in the order defined above do
    if φ ∈ V \ N(M, V) then
        M ← M ∪ {φ}
    end if
end for
```

C. Incentivizing the users

The design of an effective incentivization mechanism to decide the rewards that will be given to the users in case that they accept the system suggested trips (or *system-best* trips), requires a model of how the VSS as well as the users react. As already mentioned, the VSS needs to encourage user-based vehicle relocations in order to reduce the vehicle relocation cost by dedicated staff. Thus, it aims at incentivizing user requests based on the priorities of vehicle relocations from station to station taking into account the fluctuating demand for vehicles and parking slots at different stations over time. On the other hand, the users participating in an incentivization scheme, place a value on the additional cost/effort and/or time they should spend in order to change their original and/or destination station (i.e., their *user-best* trip). Therefore, the rewards that will be given should counterbalance this additional cost/time. Also, the incentivization scheme should take into account the potential strategic behaviour of the users aiming at maximizing their rewards.

We propose two different schemes for incentivizing users to act in favour of the VSS. Both schemes consider that there are budget constraints on how much the system can spend for incentives. In the first scheme, no user costs are taken into account. The rewards given to the users (incentives) are computed based on the relocation priorities of the suggested trips derived by the *ComputeTripSuggestions* algorithm. The second scheme employs a utility function associating each candidate trip suggestion in V with the priority of the corresponding vehicle relocation, and also considers user defined costs, each corresponding to the additional time and/or effort the user spends if he changes his original/destination station. Then, the set M of the suggested trips and the rewards to the users are computed based on the utility values and the user costs. Both schemes are truthful and budget-feasible: they give no reason to the user to report fake information (for example cost or start trip location) and respect the system's desired budget to be returned. The second scheme is also individually rational, i.e., no user gets as a reward less than his declared cost.

Scheme I. A priority-based incentivization scheme. The budget B (i.e. the maximum amount of money to be returned to the users as incentives) is determined as a fraction of the total amount of money the system is going to collect from the set R of submitted requests as follows. Let $M = \{\phi_1, ..., \phi_l\}$, $l \leq |R|$ be the output of the *ComputeTripSuggestions* algorithm, i.e., the set of the trip suggestions that can be simultaneously served by the VSS. Let $R' = \{R_{i_1}, ..., R_{i_l}\}$, $R' \subset R$, be the set of the distinct user requests such that R_{i_j} corresponds to the trip suggestion ϕ_j, $j \in \{1, ..., l\}$. For each $R_{i_j} \in R'$ let z_j be the cost the user is going to pay if he accepts the corresponding trip suggestion. This cost is determined by the VSS. For example, if the VSS's charging system is based on distance, z_j may be determined by the distance between the stations of R_{i_j}'s user's best trip. Then the budget B may be set equal to $A \cdot \sum_{R_{i_j} \in R'} z_j$ where A is a positive constant smaller than 1 defined by the VSS.

To determine the incentive given to each user or in other words, the fraction a_j of the initial cost z_j, $j \in \{1, ..., l\}$, the user is going to pay, we proceed as follows. We consider a parameter $b < 1$ indicating a minimum fraction of the cost to be paid by each user accepting a trip suggestions in M. For each $\phi_j \in M$, let π_j be the corresponding relocation priority. Then we introduce a, a decreasing function on the set of relocation priorities π as follows: $a : \mathbb{R}_+ \to [b, 1]$, $a(0) = 1$ and $a(\infty) = b$. For example, it could be $a(\pi) = b + (1-b)e^{-\pi}$. Alternatively, if $\pi_{\max}$ is the maximum of the priorities, then we may define $a : [0, \pi_{\max}] \to [b, 1]$, $a(\pi_{\max}) = b$ and $a(0) = 1$; it could be $a(\pi) = 1 - (1-b)(\frac{\pi}{\pi_{\max}})$, $0 \leq \pi \leq \pi_{\max}$.

Finally, we set the fraction a_j equal to $a(\pi_j)$, $j = 1, ..., l$

[1]A d-claw is an induced subgraph of G that consists of an independent set T_C of d nodes called talons, and the centre node that is connected to all talons; algorithms for solving the maximum independent set problem in d-claw free graphs (d constant) have constant approximation ratio [2].

and we determine the parameter b, by solving the following equation for b: $\sum_{R_{ij}\in R'} a_j z_j = \sum_{R_{ij}\in R'} z_j - B$.

It is easy to notice that the above incentivization scheme is budget-feasible, i.e., the total amount of the rewards given to the users does not exceed the budget B. As far it concerns the truthfulness of the scheme, note that although the user reports no cost, he may misreport his private data (i.e., his start trip location). This may change the set of reachable pairs and therefore, the set of candidate trip suggestions. It is easy to prove that by misreporting the start location, the user is offered either the same trip suggestion or a trip suggestion from which he derives "no value" because it corresponds to a pick-up station that is far away from the user's real start location. Therefore, the following lemma can be proved.

Lemma 1. The priority-based incentivization scheme is budget-feasible and truthful.

Scheme II. A user cost and priority-based incentivization scheme. Given a set of requests $R = \{R_1, R_2, ..., R_k\}$ and the corresponding set system V as defined in Section II-B, we associate each candidate trip suggestion $\phi = \big(R_i, s_p^\alpha(t_p), s_d^\epsilon(t_d)\big)$ in V with a utility u_ϕ corresponding to the relocation priority of vehicle $s_p^\alpha(t_p)$ to the parking place $s_d^\epsilon(t_d)$. Specifically, $u_\phi = D\cdot\pi_\phi$ where D is a constant and π_ϕ is the relocation priority. We also associate ϕ with a cost c_i corresponding to the value the user (who requested R_i) places on the additional cost/effort and/or time he spends in order to change his pick-up and/or drop-off station. Note that c_i is zero if the user is not participating in the incentivization scheme.

We hereby present an incentivization scheme (Algorithm *ComputeTrips&Rewards*) which, given the user costs $c_i, i = 1, ..., k$, the utilities u_ϕ, $\phi \in V$, and a budget B to be offered as rewards to the users by the VSS, decides the set of trip suggestions (requests that can be served) and the rewards to the respective users. The rewards are uniform in the sense that if a request R_i corresponding to a trip suggestion $\phi = \big(R_i, s_p^\alpha(t_p), s_d^\epsilon(t_d)\big)$ is accepted, then the user is paid a reward $r_i = r\cdot u_\phi$, where r is the same for all accepted requests. The total payment to the users must not exceed the budget B.

The incentivization scheme is inspired by the TM-Uniform mechanism presented in [8]. It considers the intersection graph $G = (V, E)$ of the set system V containing a node for each triple (candidate trip suggestion) $\phi = \big(R_i, s_p^\alpha(t_p), s_d^\epsilon(t_d)\big)$ in V. Each node ϕ in V is associated with the value $rate(\phi) = c_i/u_\phi$ corresponding to the cost the system should pay per unit of utility. Then the nodes in V ($|V| = K$) are sorted in decreasing order with respect to their rate value, i.e., if $\phi_1, \cdots, \phi_K$ is the sorted list of the nodes, then for $i < j$ we have that $rate(\phi_j) \le rate(\phi_i)$.

The scheme assumes a permutation σ of the nodes in V and consists of at most K iterations. It starts with the initial graph G while at each iteration G is modified. Specifically, at each iteration i an independent set M on the current graph G (Algorithm *FindIS*) as well as the utility $u(M) = \sum_{\phi\in M} u_\phi$ of M are computed. Then, if the product $rate(\phi_i)\cdot u(M)$ is greater than the budget B, the node ϕ_i is removed from the graph and the next iteration starts.

Algorithm 2 FindIS(G)

Input: Intersection Graph $G(V, E, u : V \to \mathbb{R})$, Permutation σ
Output: Independent Set M in G

$M \leftarrow \emptyset$;
$K = |V|$;
for $i \leftarrow 1$ to K **do**
  if $\sigma(i) = \big(R_i, s_p^\alpha(t_p), s_d^\epsilon(t_d)\big) \in V$ **then**
    find the triple $\phi' = \big(R_i, s_{p'}^{\alpha'}(t_{p'}), s_{d'}^{\epsilon'}(t_{d'})\big) \in V$ with maximum $u_{\phi'}$;
  end if
  $M \leftarrow M \cup \{\phi'\}$;
  $V \leftarrow V \setminus N(\{\phi'\}, V)$;
end for

Note that at any new iteration, the graph contains nodes with value smaller than or equal to $rate(\phi_{i-1})$. The algorithm stops when $rate(\phi_i)\cdot u(M)$ is less than or equal to B; r is set equal to $\min\{\frac{B}{u(M)}, rate(\phi_{i-1})\}$. The *FindIS* Algorithm takes as input a fixed permutation of the node set V of G. When a node in V corresponding to a candidate trip suggestion $\phi = \big(R_i, s_p^\alpha(t_p), s_d^\epsilon(t_d)\big)$ that concerns request R_i is considered, then the triple $\phi' = \big(R_i, s_{p'}^{\alpha'}(t_{p'}), s_{d'}^{\epsilon'}(t_{d'})\big) \in V$ which has the highest utility $u_{\phi'}$ among all the candidate trip suggestions concerning request R_i, is selected to be added to the independent set M. We will say that ϕ' is a suggested trip within M and we will use $u^{R_i}(M)$ to refer to the utility of this suggestion. The independent set M is produced after considering all the nodes in V and its utility is $u(M) = \sum_{\phi\in M} u^{R_i}(M)$. The pseudocodes of algorithms *FindIS* and *ComputeTrips&Rewards* follow.

Algorithm 3 ComputeTrips&Rewards

Input: Intersection Graph $G(V, E, rate : V \to \mathbb{R})$, Budget B, Permutation σ
Output: The Set of Trip Suggestions M; The Reward Rate r

$G'(V', E') \leftarrow G(V, E)$;
for $i \leftarrow 1$ to K **do**
  $M \leftarrow$ FindIS(G', σ);
  if $rate(\phi_i)\cdot u(M) \le B$ **then**
    $r \leftarrow min\{\frac{B}{u(M)}, rate(\phi_{i-1})\}$;
    break ;
  end if
  $V' \leftarrow V' \setminus \{\phi_i\}$;
end for

The budget B in the (*ComputeTrips & Rewards* Algorithm is determined as in the priority-based scheme but the *FindIS* algorithm is used instead of the *ComputeTripSuggestions*. The scheme is *budget-feasible*, i.e., the total amount of the rewards given to the users does not exceed B. To see this, notice that the reward for each user request R_i is $r\cdot u^{R_i}(M)$. Since $r \le B/u(M)$ the total reward is $\sum_{\phi\in M} r\cdot u^{R_i}(M) \le B$. The scheme is also *individually rational*, i.e., no user request R_i

gets as a reward less than the cost c_i corresponding to the value the user places on the additional cost/effort and/or time he spends in order to change his original and/or destination station. To prove this, let $G' = (V', E')$ be the graph when the mechanism stops. Then for each $\phi = \big(R_i, s_p^{\alpha}(t_p), s_d^{\epsilon}(t_d)\big) \in V'$ we have that $c_i/u_\phi \leq r$. If $\phi' = \big(R_i, s_{p'}^{\alpha'}(t_{p'}), s_{d'}^{\epsilon'}(t_{d'})\big)$ is a suggested trip within M then $c_i/u^{R_i}(M) \leq r$. Therefore, $c_i \leq r \cdot u^{R_i}(M) = r_i$. Finally, the scheme is *truthful* in the sense that a users has no incentive to report higher cost than his real cost because in this case either he will be suggested the same trip as if he had reported the real cost, or no trip at all. The proof of the truthfulness follows the same lines with the proof of Lemma 2 in [8] and it is omitted due to lack of space. Therefore, the following Lemma holds.

Lemma 2. The user cost and priority-based scheme is budget-feasible, individually rational and truthful.

III. Setting the priorities

We focus on a pair of stations s_p and s_d and consider the relocation of a vehicle from s_p to s_d with pick-up and drop-off times t_p and t_d, respectively. If $\delta_p(t) = o_p(t_p) - w_p(t_p)$, that is the excess (negative) of vehicles w.r.t. the target occupancy of the pick-up station and $\delta_d(t) = w_d(t_d) - o_d(t_d)$, the respective excess of empty parking spots at the drop-off station, we may define the relocation priority of moving *any* vehicle $\alpha \in \{1, ..., o_p(t_p)\}$ of s_p to *any* parking place $\epsilon \in \{1, ..., c_d - o_d(t_d)\}$ of s_d (*spot-to-spot* priorities). Therefore, notation $\delta_p^{\alpha}(t)$ is introduced to indicate the excess of vehicles if all vehicles $\{\alpha + 1, ..., o_p(t_p)\}$ have been already relocated from s_p at time t_p; $\delta_d^{\epsilon}(t)$ will be used analogously. The function $\pi = \pi(s_p, s_d, t_p, t_d, \alpha, \epsilon)$ should favour cases where both $\delta_p^{\alpha}(t)$ and $\delta_d^{\epsilon}(t)$ are positive and highly discourage a relocation with $\delta_p(t) < 0$ and $\delta_d(t) < 0$; if only one of the quantities is positive, intermediate values should be obtained.

The necessity of spot-to-spot priorities, rather than station-to-station priorities, is based on the observation that if a high priority is determined for a good amount of relocations from station A to station B, then any algorithm based on `MWIS` would easily select the whole set of the aforementioned relocations and lead to overshoot the target occupancy for A or B, while at the same time, other relocations which may balance the system could be discarded. This indicates the need for dampening any priority when getting closer to the target occupancies. All of the above are summed and expressed by our selection for the priority function $\pi\big(s_p^{\alpha}(t_p), s_d^{\epsilon}(t_d)\big)$=

$$= \begin{cases} 0.5\big(1 - \frac{\delta_p^{\alpha}(t_p)\delta_d^{\epsilon}(t_d)}{w_p(t_p)(c_d - w_d(t_d))}\big) & , \delta_p^{\alpha}(t_p) < 0 \text{ and } \delta_d^{\epsilon}(t_d) < 0 \\ 1.5 + \frac{\delta_p^{\alpha}(t_p)\epsilon}{w_p(t_p)(c_d - w_d(t_d))} & , \delta_p^{\alpha}(t_p) < 0 \text{ and } \delta_d^{\epsilon}(t_d) > 0 \\ 1.5 + \frac{\delta_d^{\epsilon}(t_d)(c_s - \alpha)}{(c_d - w_d(t_d))(c_p - w_p(t_p))} & , \delta_p^{\alpha}(t_p) > 0 \text{ and } \delta_d^{\epsilon}(t_d) < 0 \\ 2 + \frac{\delta_p^{\alpha}(t_p)\delta_d^{\epsilon}(t_d)}{(c_p - w_p(t_p))w_d(t_d)} & , \delta_p^{\alpha}(t_p) > 0 \text{ and } \delta_d^{\epsilon}(t_d) > 0 \end{cases}$$

Its range is $[0, 1.5] \cup (2, 3]$. For points of discontinuity, we may choose an intermediate value, i.e., $\delta_p^{\alpha}(t_p) = 0, \delta_d^{\epsilon}(t_d) = 0 \Rightarrow \pi(\cdot) = 1.75$.

IV. Experimental Evaluation

A. Dataset

We applied our incentivization scheme in an existing bike sharing system operating in the city of Washington DC, US, the "Capital Bikeshare" system. The system comprises 357 bike stations and approximately 2800 bikes. For our simulation, we used historical data which are publicly available[2]. These records include information about the pick-up and drop-off stations, the departure time, the duration and the distance of every trip completed. Trip data is categorized and archived according to the year of their occurrence. For the purpose of our work, we processed the data referring to the trips of the 3rd quarter of 2015, which account for over 1 million records. From these data we extracted information about the demand for bikes on every station, namely the arrival rate of the users at the stations per hour and per day.

B. Simulation

In order to assess the two *Schemes* described in Section II, we developed a Discrete Event Simulator (DES) using Java. Our DES is principally a user generator; the arrival rate of the users is based on the probabilistic model extracted from the above discussed historical data. The main attributes of users are: the station where they want to pick up a bike, the drop-off station, the departure time and their willingness to participate in the incentivization scheme. The actions pursued by a user are decided by a Finite State Machine (FSM); the FSM specifies a number of discrete states reached as a result of these actions. The instantiation of a user represents the initial state of the FSM, which is succeeded by another state upon the user issuing a reservation request for a bike. The request may be accepted or rejected by the system. The rejection of a request corresponds to a final state of the FSM, which implies the deletion of the user from the simulation procedure. In case of acceptance the user needs to confirm the reservation they have previously requested. Finally, upon the termination of the trip, the user's involvement in the simulation process completes, hence, they are removed by the simulator. Every message that is generated by a user and consequently by the DES, is formatted as JSON string and is delivered to a server through invoking an appropriate RESTful web service. The server and the web services have been developed in the context of an EU-funded program, the FP7 MOVESMART[3], for the purpose of a pilot program with electric scooters, in the city of Vitoria-Gasteiz, Spain, where the incentivization scheme will be validated in a real environment.

C. Results

We used *Scheme I* to draw the curves of the percentage of rejected trips, the altered average price-per-km users are asked to pay and the system's revenue, all as a function of the percentage of users willing to participate in our scheme.

[2]Available through the official site of the Capital Bikeshare System, https://www.capitalbikeshare.com

[3]http://www.movesmartfp7.eu/

The results are the mean of 50 iterations of each simulation for the 6-hour peak period of the day (9 a.m. to 3 p.m.) and verify the soundness of *Scheme I* (see Figure 3). The minor revenue reduction for the 25% and 50% mark is not disturbing, eventually it captures the diminishing of the average price-per-km as an initial "investment" of the system, amortized by the decreasing rejection rate, as more and more users are attracted by the scheme. The base of comparison for the price-per-km and revenue changes is their respective result for 0% participation in the incentivization scheme.

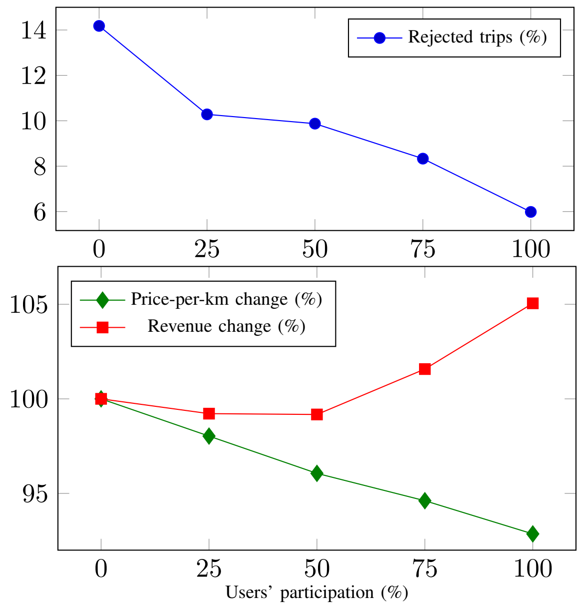

Fig. 3. *Scheme I*: Rejection rate (●), price-per-km change (♦) and revenue change (■)

Scheme II has been mostly used as a user behavior modeler. Here, all users participate in the scheme and self-declare the minimum discount they demand in order to accept a system-best trip option. The system's budget is set (approximately) to 20% of the cost of the maximum independent set of trips that can be simultaneously accepted; different biddings are simulated as Gaussian distributions around different mean values, namely 15%, 25%, 35% and 45% of the respective user-best trip cost, plus a uniform distribution over the interval $[0.1, 0.5]$.

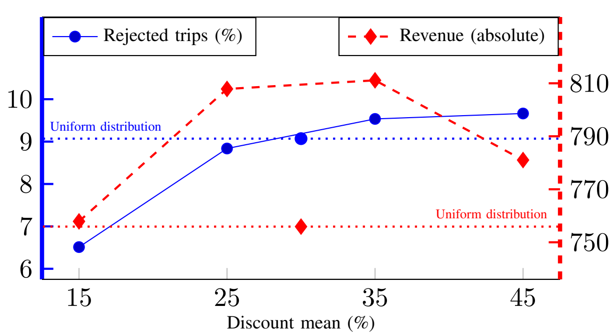

Fig. 4. *Scheme II*: Rejected requests (●) and revenue (■) vs. users' claimed discount

The simulation results match the expectation: the rejection rate increases as the mean claimed discount (including the mean 0.3 of the uniform distribution) increases. The revenue of the system, however, is maximized when the users ask for discounts of 20%–30%. Reduced claims indicate inefficient redistribution of the budget to nearly all users. Highly increased claims reveal their unfavourable coupling with the increased rejection rate.

Acknowledgement. This work has been supported by the EU FP7 Collaborative Project MOVESMART (Grant Agreement No 609026).

References

[1] M. Barth, M. Todd, and L. Xue. User-based vehicle relocation techniques for multiple-station shared-use vehicle systems. *Transportation Research Record*, 1887:137–144, 2004.

[2] P. Berman. A d/2 approximation for maximum weight independent set in d-claw free graphs. *Nord. J. Comput.*, 7(3):178–184, 2000.

[3] G. Correia and A. Antunes. Optimization approach to depot location and trip selection in one-way carsharing systems. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):233 – 247, 2012.

[4] W. Fan, R. Machemehl, and N. Lownes. Carsharing: Dynamic decision-making problem for vehicle allocation. *Transportation Research Record: Journal of the Transportation Research Board*, 2063(1):97–104, 2008.

[5] C. Fricker and N. Gast. Incentives and redistributions in homogeneous bike-sharing systems with stations of finite capacity. *European Journal on Tansportation and Logistics*, 2015.

[6] J. Froehlich, J. Neumann, and N. Oliver. Sensing and predicting the pulse of the city through shared bicycling. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 1420–1426, 2009.

[7] D. Gavalas, C. Konstantopoulos, and G. Pantziou. Design and management of vehicle sharing systems: A survey of algorithmic approaches. In *Smart Cities and Homes: Key Enabling Technologies", M.S. Obaidat and P. Nicopolitidis (Eds.) Elsevier Science*, 2015.

[8] G. Goel, A. Nikzad, and A. Singla. Matching workers expertise with tasks: Incentives in heterogeneous crowdsourcing markets. In *in NIPS Workshop on Crowdsourcing, NV, USA*.

[9] Y. Han, L. Oukhellou, and E. Côme. Towards bicycle demand prediction of large-scale bicycle sharing system. In *Proceedings of the Transportation Research Board 93rd Annual Meeting*, 2014.

[10] D. Jorge, G. Molnar, and G. Correia. Trip pricing of one-way station-based carsharing networks with zone and time of day price variations. *Transportation Research Part B*, 2015.

[11] A. Kaltenbrunner, R. Meza, J. Grivolla, J. Codina, and R. E. Banchs. Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system. *Pervasive and Mobile Computing*, 6(4):455–466, 2010.

[12] R. Katzev. Car sharing: A new approach to urban transportation problems. *Analyses of Social Issues and Public Policy*, 3(1):65–86, 2003.

[13] J. Pfrommer, J. Warrington, G. Schildbach, and M. Morari. Dynamic vehicle redistribution and online price incentives in shared mobility systems. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1567–1578, 2014.

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

[15] S. Shaheen and A. Cohen. Growth in worldwide carsharing - an international xomparisonl. *Transportation Research Record*, 1992:81–89, 2007.

[16] S. Shaheen, D. Sperling, and C. Wagner. A short history of carsharing in 90's. *Journal of World Transport Policy*, 5(3):16–37, 1999.

[17] A. Singla, M. Santoni, G. Bartok, P.Mukerji, M. Meenen, and A. Krause. Incentivizing users for balancing bike sharing systems. In *Association for the Advancement of Articial Intelligence (www.aaai.org)*, 2015.

[18] A. Waserhole and V. Jost. Pricing in vehicle sharing systems: Optimization in queuing networks with product forms. *European Journal on Transportation and Logistics*, 2015.