Les Cahiers du GERAD ISSN: 0711–2440

User-based relocation strategies for free-floating car sharing systems

M. Schiffer,
F. Rüdel, G. Walther

G–2020–03

January 2020

Citation suggérée : M. Schiffer, F. Rüdel, G. Walther (Janvier 2020). User-based relocation strategies for free-floating car sharing systems, Rapport technique, Les Cahiers du GERAD G–2020–03, GERAD, HEC Montréal, Canada.

Avant de citer ce rapport technique, veuillez visiter notre site Web (https://www.gerad.ca/fr/papers/G-2020-03) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

Suggested citation: M. Schiffer, F. Rüdel, G. Walther (January 2020). User-based relocation strategies for free-floating car sharing systems, Technical report, Les Cahiers du GERAD G–2020–03, GERAD, HEC Montréal, Canada.

Before citing this technical report, please visit our website (https://www.gerad.ca/en/papers/G-2020-03) to update your reference data, if it has been published in a scientific journal.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2020
– Bibliothèque et Archives Canada, 2020

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2020
– Library and Archives Canada, 2020

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

User-based relocation strategies for free-floating car sharing systems

Maximilian Schiffer [a,b]
Fabian Rüdel [c]
Grit Walther [c]

[a] *GERAD, Montréal (Québec), Canada*

[b] *TUM School of Management, Technical University of Munich, Munich, Germany*

[c] *RWTH Aachen University, Chair of Operations Management, Aachen D-52072, Germany*

schiffer@tum.de
fabian.ruedel@om.rwth-aachen.de
walther@om.rwth-aachen.de

January 2020
Les Cahiers du GERAD
G–2020–03

Abstract: Free-Floating Carsharing (FFCS) systems are a promising concept to reduce the traffic volume in cities. However, spatial and temporal mismatches of supply and demand require a relocation of rental cars in order to avoid low degrees of utilization. Here, especially user-based relocation strategies seem to be promising to increase utilization in a cost-efficient manner. However, a thorough optimization-based assessment of user-based relocation strategies for FFCS systems is still missing. In this paper, we introduce an integer program that optimizes the assignment of user-based relocation strategies in FFCS fleets. We develop a graph representation that allows to reformulate the problem as a k-disjoint shortest paths problem and propose an exact algorithm to solve large-size instances. Furthermore, we present a case study based on real-world data and derive managerial insights on user-based relocation strategies. Our results reveal an upper bound on the benefit of user-based relocation strategies and demonstrate that the employment of such strategies can increase the number of fulfilled rental requests by 40 %, while increasing the operator's profit by 10 %.

Keywords: free-floating car sharing, user-based relocation, polynomial time algorithm

1 Introduction

In recent years, car sharing services have been hyped as a sustainable complement to public transport to realize sustainable individual (urban) mobility and thus to mitigate its negative externalities. In this context, the concept of Free-Floating Carsharing (FFCS) systems has been vividly discussed and FFCS fleets have been deployed in most large cities all around the world, e.g., by BMW (DriveNow) and Daimler (car2go). In a FFCS system, users pay a usage fee to pick up a car to move from an origin to a destination, where they drop the car such that the next user can pick it up. Such systems offer flexible mobility services that increase a vehicle's utilization, lower the number of required parking spaces (Grazi and van den Bergh (2008)), and reduce traffic congestion (Button (2002)).

However, the operative effect of currently deployed FFCSs as well as their viability for operators fell short of expectations for the following reasons: although being designed as a flexible mobility service, the perceived flexibility for customers is limited due to spatial or temporal demand mismatches, i.e., a preceding customer does not necessarily drop a vehicle close to the succeeding customer's origin. To resolve this problem without increasing the vehicle fleet size, carsharing operators relocate vehicles to decrease the imbalance between vehicle availability and customer demand. Nowadays, these relocations are mostly performed operator-based, i.e., staff members relocate vehicles during low demand times or during the night. Still, applying such a relocation concept remains expensive and (time) inefficient because additional staff must be paid to relocate vehicles. Further, a vehicle cannot be used by a customer while it is being relocated.

Our recent discussions with major players in the FFCS business revealed that they consider user-based relocation strategies as a viable alternative to operator-based relocation strategies. In a user-based relocation strategy, the car sharing operator provides incentives to the customer to relocate the car, e.g., by offering a discounted fare in exchange for an adjustment of the origin (destination) or the start (arrival) time of a trip. However, it remains an open question if the potential of user-based relocation strategies is sufficient to remedy a significant share of the total demand mismatch such that the need for operator-based relocation becomes superfluous or is at least significantly reduced.

So far, only simulation-based approaches that focus on customer acceptance exist to answer this question. With this work, we close a remaining gap in this field by providing an optimization-based approach that allows to exploit the maximum potential a user-based relocation strategy can offer under perfect conditions. These results are beneficial for practitioners to get an upper bound on potential benefits and for researchers to benchmark heuristic principles used in simulation studies.

1.1 Aims and scope

The contribution of this paper is threefold. First, we develop a mathematical model that formalizes the operator's planning problem in order to exploit the benefit of user-based relocation strategies. We introduce the Car Sharing Relocation Problem with Flexible Drop-Offs (CSRP-FDO) which relocates vehicles by slightly modifying a customer's origin or destination, or start or arrival time in exchange for a monetary discount. Second, we develop a graph reformulation that reduces the complexity of the underlying optimization problem significantly. Based on this reformulation, we present an exact algorithm that allows to solve the CSRP-FDO in polynomial time. We present this algorithm in a generic way such that it can also be used for related problems, e.g., dispatching vehicles in a ride-hailing system. Third, we apply this algorithm to a case study for car2go in Vancouver, Canada, and derive managerial insights on the maximum improvement potential that can be leveraged with different user-based relocation strategies.

1.2 Organization

The remainder of this paper is organized as follows. Section 2 gives an overview of recent research on relocation strategies in car sharing systems. Section 3 details our methodology by introducing an integer problem formulation as well as a graph reformulation, which allows for a polynomial time algorithm. Section 4 describes the design of our case study. Finally, Section 5 presents the results of our experiments before Section 6 concludes the paper with a summary and an outlook on future research.

2 Literature review

In this section, we review related work. We first give a concise overview on work related to operator-based relocation strategies, before we discuss work on user-based relocation strategies. Finally, we summarize our discussion.

Operator-based relocation

Most papers published so far focus on operator-based relocation strategies. While early papers present different simulation models (Barth and Todd, 1999; Kek et al., 2006), later works allow to identify optimal relocation strategies using mathematical optimization with different extensions, e.g., allowing to integrate demand uncertainty (Fan et al., 2008; Nair and Miller-Hooks, 2010). Some papers consider constraints for electric vehicles (Bruglieri et al., 2014; Gambella et al., 2018), or aggregate relocation on a more strategic level (Boyacı et al., 2015). Most papers that analyze operator-based strategies focus on Station-Based Carsharing (SBCS). Only Paschke et al. (2017) study operator-based relocation strategies for FFCS based on an agent-based simulation within the MATSim framework.

Concluding, simulation and optimization models are well studied for operator-based relocation strategies. Herein, the more advanced optimization models allow to solve the problem optimally and account for complex side constraints, e.g., uncertainties. However, most of these studies focus on SBCS. For FFCS systems only few simulation-based models exist so far.

User-based relocation

Research on user-based relocation strategies is still scarce. Early papers analyzed trip-joining and trip-splitting strategies to balance demand and supply (Barth et al., 2004; Uesugi et al., 2007). While such concepts seem amenable for ride-hailing services, for car-sharing services barriers exist due to safety and security reasons as well as privacy and convenience preferences (Correia and Viegas, 2011; Chan and Shaheen, 2012; Jorge and Correia, 2013). Accordingly, more recent work focused on user-based relocation strategies for individual FFCS, exploiting the benefits of modifying temporal or spatial characteristics of individual trips via incentives. In this course, Cepolina and Farina (2012), Di Febbraro et al. (2012), Clemente et al. (2013) and Di Febbraro et al. (2018) focused on the adjustment of drop-off locations. Other approaches analyzed combined strategies, e.g., modifying drop-off location and arrival times (Clemente et al., 2013), or combined temporal and spatial adjustments of trips with trip-joining strategies and paid relocation (Schulte and Voß, 2015). First approaches focused on combined user-based and operator-based relocation strategies (Weikl and Bogenberger, 2013; Clemente et al., 2017)).

All of these approaches base on simulation models that utilize different methodologies, e.g., discrete event simulation (Di Febbraro et al., 2012; Clemente et al., 2017), timed Petri Nets (Clemente et al., 2013), or discrete-event simulation (Schulte and Voß, 2015). So far, these approaches do not consider optimization techniques at all or integrate them only as a reactive, heuristic controler within a simulation. A pure optimization-based approach that allows to identify the maximum savings potential under perfect conditions is missing so far.

Summary

Table 1 summarizes the key characteristics of related publications on user-based relocation concepts. To the best of the authors' knowledge, only few other papers focus on this concept so far. All of these publications use a simulation-based methodology and only three focus on FFCS. Furthermore, none of the recent papers exploits the full range of relocation strategies, i.e., adjusting the start time, the arrival time, the origin, and the destination of a trip.

Table 1: Overview of user-based relocation models.

	I	II	III	IV	V	VI	VII	VIII	IX
Station-Based Carsharing	✓	✓	✓	✓	✓				
Free-Floating Carsharing						✓	✓	✓	✓
adjusted start time									✓
adjusted arrival time				✓					✓
adjusted origin								✓	✓
adjusted destination			✓	✓	✓	✓	✓	✓	✓
simulation-based methodology	✓	✓	✓	✓	✓	✓	✓	✓	
optimization-based methodology									✓

Indices I - IX signify publications as follows: (I) Barth et al. (2004); (II) Uesugi et al. (2007); (III) Cepolina and Farina (2012); (IV) Clemente et al. (2013); (V) Clemente et al. (2017); (VI) Di Febbraro et al. (2012); (VII) Di Febbraro et al. (2018). (VIII) Schulte and Voß (2015); (IX) this paper.

To close this gap, we introduce an optimization-based approach for user-based relocation strategies in FFCS systems that allows to adjust spatial and temporal characteristics of customer requests. We then use this approach to analyze the impact of user-based relocation strategies under perfect conditions in order to determine their maximum improvement potential.

3 Methodology

In this section, we introduce the CSRP-FDO in order to maximize an FFCS operator's profit through user-based relocation strategies. First, we present an integer program for the CSRP-FDO in Section 3.1. Then, we show how this problem can be reformulated using a compact graph representation in Section 3.2. Based on this representation, we introduce a polynomial time algorithm to solve the CSRP-FDO in Section 3.3.

3.1 Problem formulation

The CSRP-FDO maximizes the profit of a car sharing fleet with homogeneous vehicles for a given time horizon by scheduling a (feasible) sequence of trips to each car. In a sequence, consecutive trips must be feasible, i.e., a preceding car's destination must match a succeeding car's origin in both space and time dimension. Note that a spatial match does not require two trips to start and end at the very same position. Instead, we require that the preceding trip's destination and the succeeding trip's origin are within a walking distance below a certain threshold δ. On a similar note, a temporal match is given if the preceding trip ends before the next trip starts.

If two trips do not match, the fleet operator can apply a user-based relocation strategy to (slightly) adjust a trip's start time (S), arrival time (A), origin (O), or destination (D). In return, the customer receives a discounted fare for the trip she requested in order to compensate the caused inconvenience. In personal communications with fleet operators, we found that the customer is not expected to accept more than one modification to her trip at a time. Accordingly, the operator can only apply one of the mentioned adjustments to each customer request, i.e., the corresponding trip. In this basic model, we consider a deterministic setting in which a customer always accepts the proposed trip modification.

We use the following notation as summarized in Table 2 to formalize this setting. Let $\mathcal{J}$ be the set of all trips. For each trip $j \in \mathcal{J}$, a quintuple $L_j = (o_j, d_j, s_j, a_j, p_j)$ states its origin o_j, its destination d_j, its start time s_j, its arrival time a_j, and its profit p_j. Furthermore, we separate trips into original trips out of subset $\mathcal{J}_o$ and modified trips out of subset $\mathcal{J}_m$. A modified trip $i \in \mathcal{J}_m$ results from changing an original trip $j \in \mathcal{J}_o$ according to a specific relocation strategy. We refer to the original

Table 2: Notation used for the CSRP-FDO.

$\mathcal{K}$	set of all cars
$\mathcal{J}$	set of all trips
$\mathcal{J}_o$	set of all original trips
$\mathcal{I}_j$	set of all modifications of trip j including j itself
$\mathcal{B}_j$	set of all predecessors of trip j
$\mathcal{R}_j$	set of all rivals of trip j
$\mathcal{S}_j$	set of all trips that overlap in time with trip j
x_{jk}	binary variable - 1 if car k covers trip j
p_j	profit earned when fulfilling trip j
o_j, d_j, s_j, a_j	origin, destination, start time, arrival time of trip j

trip j of the modified trip i as its father $f_i = j$, and indicate the father of a modified trip with a superscript on the corresponding quintuple $L_i^{f_i}$. Note that $\mathcal{J}_o$ and $\mathcal{J}_m$ are disjunct but collectively exhaustive, i.e., $\mathcal{J}_o \cup \mathcal{J}_m = \mathcal{J}$. Set $\mathcal{I}_j = \{j\} \cup \{i \in \mathcal{J}_m \mid L_i^{f_i} = L_i^j\}$ contains the original trip j and its possible modifications. Finally, we associate each trip j with a specific profit p_j. If $j \in \mathcal{J}_m$ is a modified trip, we consider the profit to be already decreased by the related discount.

With $\mathcal{B}_j = \{i \in \mathcal{J} \mid d_i = o_j \ \wedge \ a_i \leq s_j\}$, we keep track of predecessors of trip j. Multiple trips can arise at the same time such that a car cannot serve more than one trip out of a subset $\mathcal{R} \subseteq \mathcal{J}$. We refer to such trips as rivals and denote the corresponding set by $\mathcal{R}_j = \{i \in \mathcal{J} \mid o_i = o_j \wedge s_i \leq s_j, \ i \neq j\}$. Furthermore, $\mathcal{S}_j = \{i \in \mathcal{J} \mid s_j \leq s_i < a_j \ \vee \ s_j < a_i \leq a_j, \ i \neq j\}$ contains all trips $i \in \mathcal{J}$ that show a temporal overlap with trip j. Let $\mathcal{K}$ be the set of cars for a given fleet size $|\mathcal{K}|$. For each car $k \in \mathcal{K}$, o_k denotes its initial location. We use the binary variable x_{jk} to state if car $k \in \mathcal{K}$ covers trip $j \in \mathcal{J}$ ($x_{jk} = 1$) or not ($x_{jk} = 0$).

With this notation, the CSRP-FDO results as follows.

$$\max \qquad Z = \sum_{j \in \mathcal{J}, k \in \mathcal{K}} p_j x_{jk} \tag{1}$$

$$\sum_{i \in \mathcal{I}_j, k \in \mathcal{K}} x_{ik} \leq 1 \qquad \forall j \in \mathcal{J}_o \tag{2}$$

$$x_{jk} \leq \mathbb{1}_{o_k = o_j} + \sum_{i \in \mathcal{B}_j} x_{ik} - \sum_{l \in \mathcal{R}_j} x_{lk} \qquad \forall j \in \mathcal{J}, \ \forall k \in \mathcal{K} \tag{3}$$

$$\sum_{i \in \mathcal{S}_j} x_{ik} \leq M(1 - x_{jk}) \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \tag{4}$$

$$x_{jk} \in \{0, 1\} \qquad \forall j \in \mathcal{J}, \ \forall k \in \mathcal{K} \tag{5}$$

The Objective (1) maximizes the total profit of all served customer requests. Constraints (2) secure single assignment of demands. Constraints (3) ensure the feasibility of a sequence of scheduled trips for each car as it enforces each assigned demand to succeed one of its predecessors or to be the first trip. Constraints (4) secure the single assignment of vehicles to trips over time. Constraints (5) state the domain of x_{jk}.

A few comments on this modeling approach are in order. First, we consider a deterministic planning problem that assumes perfect information about future demands over the considered planning horizon. Although limiting, this assumption is in line with the scope of our studies which is to identify an upper bound on the improvement potential that can be reached with user-based relocation strategies. Further, recent works on forecasting mobility demand reveal a high accuracy (Tsao et al., 2018) such that our approach can still form the basis for a real-time receding horizon algorithm, which uses additional information from elaborate forecasts, in practice. Second, we do not consider that customers may reject an operator's offer to modify her trip. Again, this is in line with our objective of analyzing the theoretical maximum improvement potential. For further research, one could apply our algorithm in a simulation environment in a receding horizon fashion to see how customer acceptance rates influence the improvement potential.

3.2 Graph representation

In general, the CSRP-FDO resembles a vehicle dispatching problem which assigns trips to vehicles, similar to vehicle dispatching for ride-hailing and taxi fleets. Naturally, such problems contain an inherent combinatorial complexity such that even concise integer programs as presented in Section 3.1 stay hardly computationally tractable for large-scale instances. In the following, we develop a graph representation that resembles parts of this complexity as it allows to capture all information about precedence constraints and rivalry between trips in the graph itself.

We consider a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ with a set of vertices $\mathcal{V}$ and a set of arcs $\mathcal{A}$. Figure 1 shows an example of such a dispatching graph, which captures information about each car's initial location and precedence constraints between trips. The vertex set consists of different subsets $\mathcal{V} = \mathcal{V}_k \cup \mathcal{V}_o \cup \{o, n\}$, where $\mathcal{V}_k$ denotes a set of vertices representing each car's initial location at the beginning of the planning horizon, and each vertex in $\mathcal{V}_o$ represents an unmodified trip. Further, we add an artificial source vertex o and an artificial sink vertex n to $\mathcal{V}$. We use the arc set $\mathcal{A}$ to model precedence constraints between trips and to provide profit information. Accordingly, each arc $(u, v) \in \mathcal{A}$ connects two vertices u and v only if *i)* a car can reach a trip from its initial location in $t = 0$ or if *ii)* two trips are feasible, i.e., can be covered by the same car as they match in time and space. Moreover, we connect the artificial source to all initial car locations and all trips and all initial car locations to the artificial sink. Each arc (u, v) is associated with a weight w_{uv}, which reflects the profit of covering the trip denoted by vertex v such that $w_{uv} = p_v$. Then, considering all original trips in $\mathcal{V}_o$ yields a basic graph representation to model precedence constraints without using user-based relocation strategies.

To consider user-based relocations, we modify this basic graph by adding artificial vertices to increase the graph's connectivity: Figure 2 shows such a modification for all potential relocation strategies. For each relocation strategy $(S,\ A,\ O,\ D)$, we add nodes S_j, A_j, O_j, $D_j \subseteq \mathcal{V}'_o$ to trips $j \in \mathcal{J}_o$ that we want to modify. To model the discount that the operator offers to the customer, the weight of the incoming arcs of these nodes is negative ($w_{uv} = d_{\mathrm{v}} < 0$). Then, we add all additional arcs to $\mathcal{A}$ that allow for an additional connection between another trip and a synthetic vertex (i.e., a user-based relocation). We only add vertices that create at minimum one additional arc. Clearly, we can afterwards remove the artificial vertices and merge the remaining arcs to sparsen the graph by shrinking $|\mathcal{V}|$ and $|\mathcal{A}|$. Figure 3 shows an example of such an extension. Here, the dashed arc represents a connection results from modifying the origin of v_7 and adapting the profit p_7^o accordingly. Using this technique, we allow to model different combinations of user-based relocation strategies via subpaths in $\mathcal{G}$. between v_4 and v_7 that

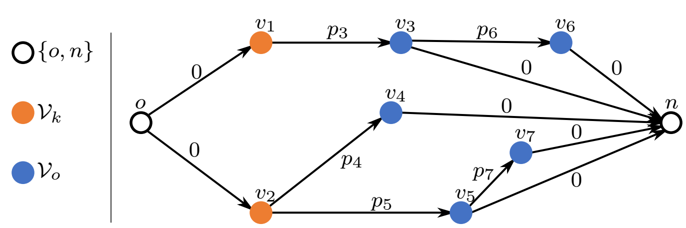


Figure 1: Dispatching graph representation of a CSRP-FDO instance with two vehicles and five trips.

3.3 Polynomial time algorithm

Using the graph representation introduced in Section 3.2, we note that maximizing the profit in the CSRP-FDO equals solving a k-disjoint Shortest Paths Problem on $\mathcal{G}'$ with $\mathcal{V}' = \mathcal{V}$, $\mathcal{A}' = \mathcal{A}$, and $w'_{uv} = -w_{uv},\ \forall (u, v) \in \mathcal{A}'$. By negating the arc weights and identifying k disjoint shortest paths, we consider the additive inverse of the original objective. To consider the FFCS system's fleet size, we

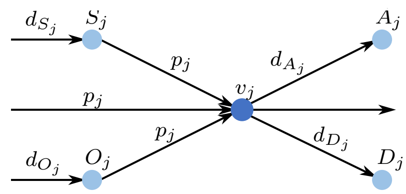


Figure 2: All possible modification vertices for a trip v_j.

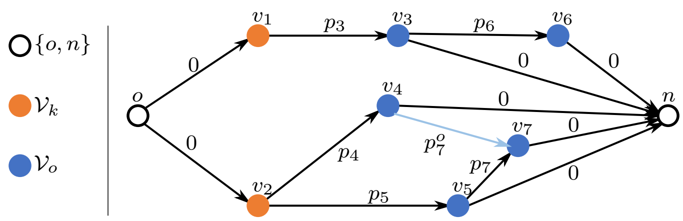


Figure 3: Graph representation of a CSRP-FDO instance with two cars and five trips.

limit the number of disjoint paths to $k = |\mathcal{K}|$. Based on the proof of Suurballe (1974) that shows that it is possible to increase the number of shortest disjoint paths on a graph G from i to $i+1$ by finding a shortest interlacing on a modified graph G', we present a polynomial time algorithm with a complexity of $O(k(|\mathcal{A}| + |\mathcal{V}|log|\mathcal{V}|))$ in the following.

Figure 4 shows the pseudo-code of this algorithm. The presented algorithm consists of two steps. First, we initialize the algorithm by finding a shortest path with the Bellman-Ford Algorithm. Afterwards, we iterate $i = 1, \ldots, k-1$ times to find the remaining disjoint shortest paths. Every iteration consists of three steps: i) modifying the graph, ii) computing the shortest path on the modified graph with a non-negative shortest path algorithm and iii) deriving the $i+1$ shortest disjoint paths.

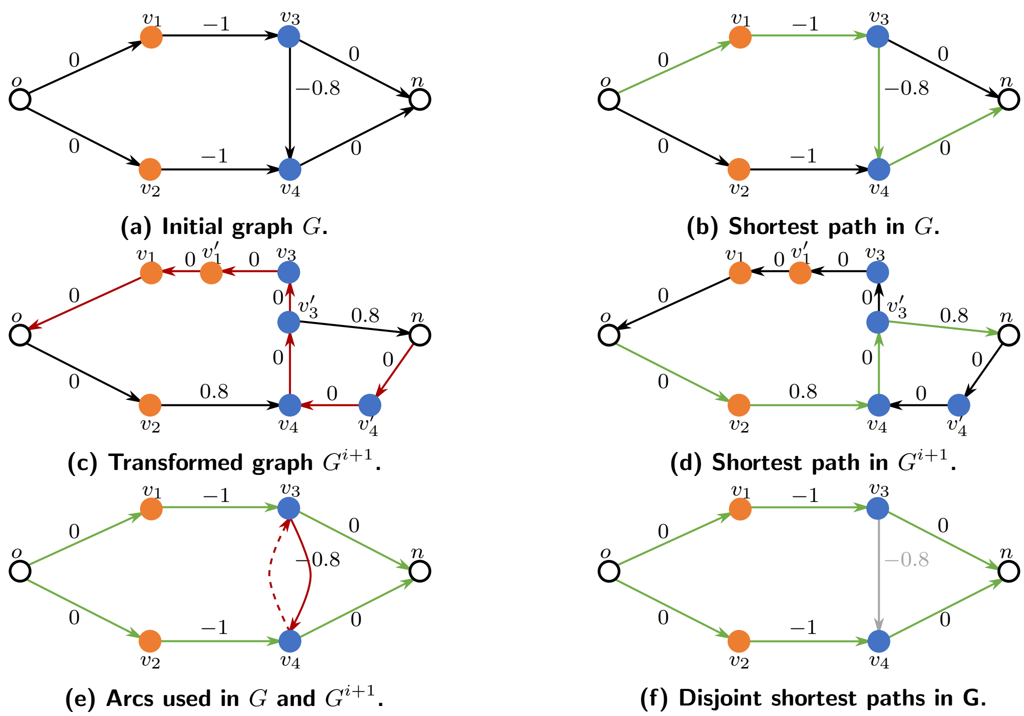
In the following, we explain this algorithm, and illustrate it with a simple example (see Figure 5a) in which we solve the problem for two cars and two trips. Figure 5b shows the graph with the marked first shortest path. From this shortest path, we obtain the distance label of each vertex ($e \in \mathcal{E}^1$) and its predecessor on the path ($o \in \mathcal{O}_{p_1}$).

With this information, we start iterating by updating the Graph G to a modified Graph G^{i+1} (Figure 5c). Herein, we first modify all path arcs and path nodes by calling the function UpdatePathArcs(). Afterwards, we complete the modification by modifying the non-path arcs with UpdateNonPathArcs(). The modification performed by these two functions consists of the following three steps. We revert all arcs that belong to existing paths, split the vertices into an incoming and an outgoing part, and update the arc weights with the reweighting function of the Johnson Algorithm (see Johnson, 1977).

After modifying the graph, we use a modification of a non-negative shortest path algorithm, e.g., the Dijkstra Algorithm, to find the next shortest path. Additionally, we calculate in this step the new

Figure 4: Algorithm for the CSRP-FDO.

$\mathcal{G}^1 \longleftarrow (\mathcal{V}, \mathcal{A}, \mathcal{C})$
$p_1,\ \mathcal{O}_{p_1},\ \mathcal{E}^1 \longleftarrow$ bellman ford*$(\mathcal{G}^1,\ 0,\ n)$
$\mathcal{P}^1 \longleftarrow \{p_1\}$
for $i = 1, \ldots, k-1$ **do**
  $\mathcal{G}^{i+1} \longleftarrow$ UpdatePathArcs$(\mathcal{G}^1,\ \mathcal{P}^i,\ \mathcal{E}^i)$
  $\mathcal{G}^{i+1} \longleftarrow$ UpdateNonPathArcs$(\mathcal{G}^{i+1},\ \mathcal{P}^i,\ \mathcal{E}^i)$
  $\hat{p}_{i+1},\ \mathcal{O}_{\hat{p}_{i+1}},\ \mathcal{E}^{i+1} \longleftarrow$ dijkstra*$(\mathcal{G}^{i+1},\ 0,\ n)$
  $\mathcal{P}^{i+1},\ \mathcal{O}_{\mathcal{P}^{i+1}} \longleftarrow$ construct paths$(\mathcal{P}^i,\ \mathcal{O}_{\mathcal{P}^i},\ \hat{p}_{i+1},\ \mathcal{O}_{\hat{p}_{i+1}})$



(a) **Initial graph G.** (b) **Shortest path in G.**

(c) **Transformed graph G^{i+1}.** (d) **Shortest path in G^{i+1}.**

(e) **Arcs used in G and G^{i+1}.** (f) **Disjoint shortest paths in G.**

Figure 5: Example with $|\mathcal{V}_k| = 2$ and $|\mathcal{V}_o| = 2$ for Algorithm 4.

distance labels and update the predecessor information. Figure 5d shows the shortest path on G^{i+1}. With function construct path(), we conduct the last step of the iteration and derive the $i + 1$ disjoint shortest paths of graph G. We first mark all arcs that are on a shortest path of either graph G or graph G^{i+1}. If there exists an arc from vertex i to vertex j in G^{i+1} that is the reversed arc of an arc in G (Figure 5e), we unmark both arcs. Afterwards, we derive the $i + 1$ disjoint shortest paths by using the predecessors to trace all marked arcs that reach the sink vertex n back to the source vertex 0. Figure 5f shows the solution of the example.

With this algorithm, we are able to solve all instances of the CSRP-FDO to optimality in polynomial time, while it is not possible to solve realistic instances with a standard desktop computer using a commercial solver due to the memory requirements of the corresponding IP-formulation.

4 Experimental design

We base our experiments on real-world data for car2go Vancouver, which was the largest fleet that car2go operated in terms of number of cars and members during the time the data was collected. Our data set covers a time span of 63 days during March 2015 and May 2015. Figure 6 shows the catchment area of this data set, which covers the fleet's main service and operations area in central Vancouver. The data set bases on idle times of cars, i.e., each data point contains information on the car's id, its position, and the time span when it idled there. Based on this data, we reconstructed a representative set of 164,445 trips for the analyzed time period.

Although the trip number appears to be sufficiently large for a statistically significant computational analysis, this is not the case if one looks at a daily resolution. Within the analyzed time period, the available data splits across 45 working days and 18 weekend days. Both, the imbalance between working and weekend days as well as the small number of days does not allow to base our study on a sufficient number of scenarios. To resolve this problem, we use conditional mass probability distributions (CMPDs) to sample a sufficiently large, yet realistic, set of scenarios for our studies.

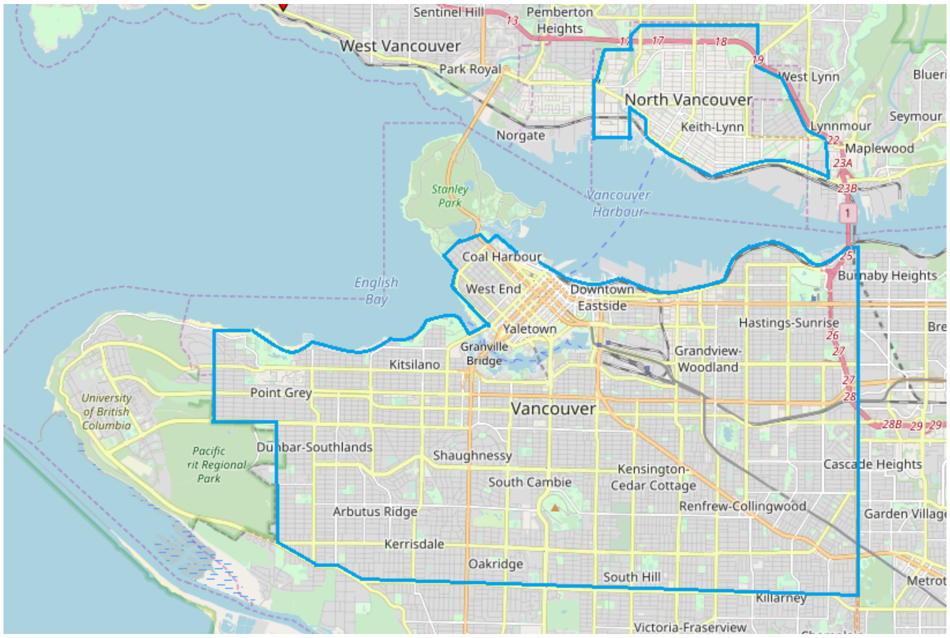


Figure 6: Catchment area of our case study data[1].

In the following, we first formalize the concept of CMPDs (Section 4.1), before we detail our scenario generation (Section 4.2) and discuss our sensitivity analyses (Section 4.3).

4.1 Conditional Mass Probability Distributions

In general, a CMPD is used in the field of statistics to describe a dependency between two jointly distributed random variables. Such a conditional probability depicts the contrary of the commonly known marginal probability, where a random variable is independent of other random variables. Formally, we consider two random variables X and Y. Then, the conditional probability that Y takes realization y when X takes realization x is

$$\mathbb{P}(Y = y \mid X = x) = \frac{\mathbb{P}(\{X = x\} \cap \{Y = y\})}{\mathbb{P}(X = x)}. \tag{6}$$

Further, we can express this relationship more generally by using a conditional probability mass function (CPMF) without specifying x such that

$$p_{Y|X}(y \mid x) \mathrel{\widehat{=}} \mathbb{P}(Y = y \mid X = x) = \frac{\mathbb{P}(\{X = x\} \cap \{Y = y\})}{\mathbb{P}(X = x)}. \tag{7}$$

Using the chain rule of probabilities, we can extend the CPMF definition to an arbitrary number of jointly distributed random variables $X_1, ..., X_n$ such that

$$p_{X_n,...,X_1}(x_n, ..., x_1) \mathrel{\widehat{=}} \mathbb{P}(X_n, ..., X_1) = \mathbb{P}(X_n)\mathbb{P}(X_{n-1}, ..., X_1), \tag{8}$$

which recursively yields

$$p_{X_n,...,X_1}(x_n, ..., x_1) \mathrel{\widehat{=}} \mathbb{P}(X_n, ..., X_1) = \prod_{i=1}^{n} \mathbb{P}(X_i \mid \bigcap_{j=1}^{n-1} X_j). \tag{9}$$

[1]The figure bases on OpenStreetMap, which is data licensed under the Open Data Commons Open Database License (ODbL) by the OpenStreetMap Foundation (OSMF). For more information see www.openstreetmap.org/copyright.

In our specific application case, we use this theory to fit a dedicated CPMF $p_{S,L,O,D,T}(s, l, o, d, t)$, which characterizes the probability of a specific car-sharing trip, dependent on five random variables (S,L,O,D,T), determining the start time s, the duration l, the origin o, the destination d of a trip, and the type of day t. We then use this CPMF to generate customer requests within our scenario generation.

4.2 Scenario generation

To create sufficiently dense data points, we discretize the spatial and the time dimension of our data before we generate our scenarios. Spatially, we discretize the catchment area into a board of 18x36 equally sized squares. This yields a resolution for which any pair of points located in two neighboring squares can be reached within less than 500 meters, which is the maximum distance a customer is willing to walk to a car sharing vehicle (cf. Herrmann et al., 2014). Temporally, we divide a day into 10-minute intervals, starting at 5:00am and ending at 12:00pm. We do not consider the time interval between 0:00am and 5:00am for our studies, because car2go conducted operator-based relocations during this period, which would bias the data set.

Based on this discretized data, we fitted a CPMF as discussed in Section 4.1, which allows us to generate trips for the whole time horizon. Then, we created scenarios in two steps. First, we picked an initial location for each vehicle based on a marginal probability distribution for the initial location of vehicles in the original data set at 5:00am. Second, we generated customer requests using the respective CPMF.

We set the number of vehicles to the average number of vehicles deployed during the days the data set was collected. Similarly, we generate n customer requests, with n being the average daily number of requests from the original data. To calculate the revenue of the car sharing operator, we assumed a revenue of 0.32 Canadian Dollars (CAD) per minute. Consumer studies showed that a vast majority of carsharing customers is willing to accept a more distant car for a price discount of about one third (Herrmann et al., 2014). Accordingly, we consider a discount of 33% in case a user-based relocation strategy is applied.

We limit the distance between origin and destination by the distance that a driver can travel within the time of his or her car rental ($(a_j - s_j)$ periods). The speed limit within Vancouver is 50 km/h ($\frac{50}{6}$ km/period). Therefore, a driver driving at the speed limit ($\frac{50}{6}$ km/period) for the whole trip duration ($(a_j - s_j)$ periods) can travel at most $\frac{50}{6}(a_j - s_j)$ km.

4.3 Sensitivity analyses

In our studies, we distinguish between working day and weekend day scenarios. For both we used an individual CPMF and created 20 scenarios each as a base case. Here, we set the number of vehicles to 626 and 623, which is the corresponding average number of vehicles from the real data for working and weekend days. Analogously, we accounted for 2,787 and 3,023 daily trips.

Besides, we create two additional sets of scenarios, one in which we vary the number of available vehicles and one in which we vary the number of daily customer requests. We vary the number of cars in between $[300; 630]$ with a step width of 10 and the number of daily customer requests in between $[2,500; 3,400]$ with a step width of 100. By so doing, we create in total 880 working day and 880 weekend day scenarios.

We analyze only pure user-based relocation strategies, i.e., strategies where the operator can change a single characteristic of a trip, since our discussions with practitioners indicated that a customer is expected to react much more reluctant to accept changes of multiple characteristics of her trip. Table 3 summarizes all possible relocation strategies, which consists of modifying either the start time (S), the arrival time (A), the origin (O) or the destination (D) of a customer request.

Table 3: Overview of relocation strategies.

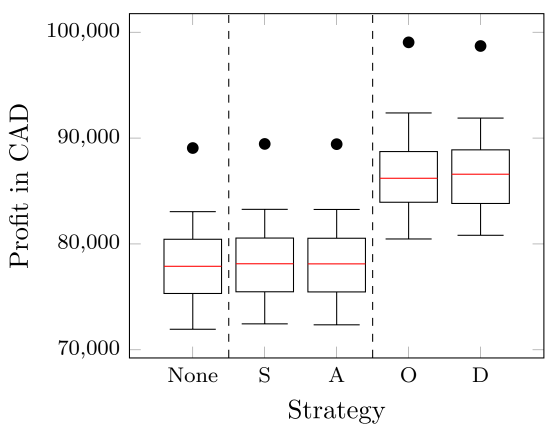
name	modified property	extend of modification	discount
S	start time	1 period later	33%
A	arrival time	1 period earlier	33%
O	origin	shift into a neighboring zone	33%
D	destination	shift into a neighboring zone	33%

5 Results

In this section, we discuss our results. First, we discuss the results for our base case, before we detail the findings of our sensitivity analysis. For all discussions, we assume that the system is operated under perfect conditions, i.e., complete (temporal) information and full customer cooperation in terms of accepting a user-based relocation. This assumption is in line with the goal of our studies, which is to analyze the maximum savings potential of different user-based relocation strategies.

5.1 Base case

Figure 7 shows a Box-Whisker-Plot that details the total profit for the base case working day scenarios and each strategy. As can be seen, a temporal user-based relocation strategy that modifies either the start time (S) or the arrival time (A) of a trip does not significantly increase the operator's profit compared to a fleet operation without user-based relocation (None). On average, both temporal strategies increase the profit by only 0.3%. Contrary, spatial user-based relocation strategies can increase the profit substantially by on average 10.9% (modifying the origin) or 11.1% (modifying the destination). Notably, the profit distribution when applying a destination modification shows a slightly more narrow distribution, yielding a higher minimum value, but also a lower maximum value.

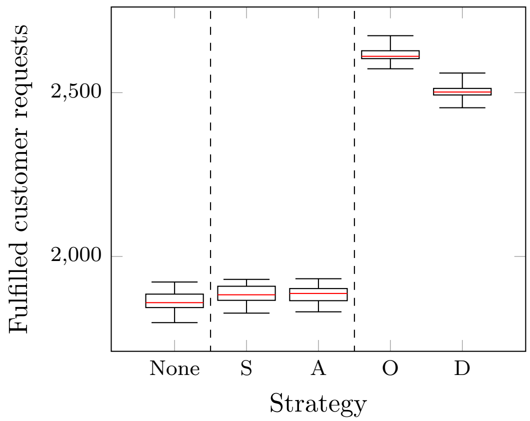


	None	S	A	O	D
Max	89,1	89,4	89,4	99,0	98,7
3. quartile	80,4	80,6	80,5	88,7	88,9
Median	77,9	78,1	78,1	86,2	86,6
1. quartile	75,3	75,5	75,5	83,9	83,8
Min	71,9	72,5	72,4	80,5	80,8

Figure 7: Profit generated during a working day.

Figure 8 details a distribution of the number of fulfilled trip requests for each working day scenario and strategy. As can be seen, the results for the temporal relocation strategies resemble the profit analysis of Figure 7 and show only minor improvements of on average 1.2%. Interestingly, the results for spatial relocation strategies do not resemble the profit discussion. As can be seen, a spatial relocation strategy that modifies a trip's origin yields more fulfilled customer requests compared to a strategy that modifies a trip's destination. On average, modifying trip origins yields an increase in fulfilled trips of 40.5% (752 trips) compared to a solution without relocation, whereas modifying trip destinations yields an average increase of 35% (643 trips).

These results allow for the following additional insights. Apparently, one may yield similar results by applying either of the temporal relocation strategies, because both modifying a trip's start time



	None	S	A	O	D
Max	1,922	1,930	1,932	2,674	2,560
3. quartile	1,885	1,909	1,902	2,628	2,513
Median	1,859	1,883	1,887	2,611	2,502
1. quartile	1,844	1,866	1,865	2,604	2,493
Min	1,798	1,827	1,831	2,573	2,454

Figure 8: Fulfilled customer requests during a weekday.

or modifying a trip's arrival time can be used to shift a trip in time analogously. Focusing on spatial relocation strategies, modifying trip origins yields a higher impact than modifying trip destinations, because modifying a trips origin yields an additional degree of freedom for each trip sequence when modifying the first trip. However, this additional degree of freedom only helps to increase the number of served customer requests, which does not necessarily corellate with an increase in the operator's profit. Both, the origin-modification and the destination-modification strategy yield an average profit increase of about 11% but the destination-modification strategy shows a 5% lower increase in terms of fulfilled customer requests.

Figures 9&10 complement our analysis by showing the average number of occupied cars for each time step (Figure 9) and the average number of modified trips for each time step (Figure 10), aggregated over all working day scenarios. As can be seen in Figure 9, the number of occupied cars remains always 30% below the fleet size, even during peak times when applying the most successful relocation strategies. Figure 10 shows that only few modifications during the evening peak are possible with temporal relocation strategies, which resembles our previous findings on a more granular time resolution. Focusing on spatial relocation strategies, one can see that modifications occur during the whole day. Although, the largest share of modifications occurs during the morning and evening peaks, both strategies keep a constant level of relocations during off-peak hours. Especially, the origin-modification strategy preserves a higher constant level of relocations during off-peak hours, which finally enables more relocations during peak hours.

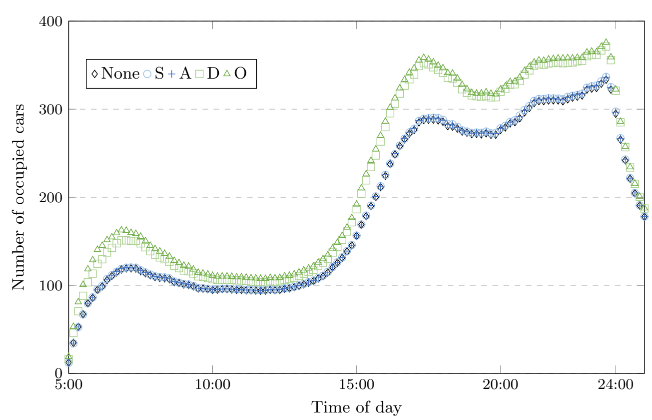


Figure 9: Average number of occupied cars for each time step on a working day.

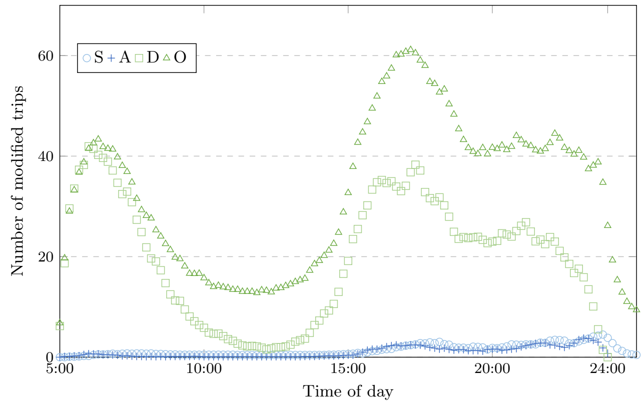


Figure 10: Average number of modified trips for each time step on a working day.

Figure 11 shows the average number of unfulfilled rental requests for working day scenarios for each time step. As can be seen, with spatial relocation strategies, the number of unsatisfied rental requests can be kept (well) below 10 requests per time step; especially when modifying the trips' origins. This clarifies that the higher effectiveness of the spatial relocation strategies results from its capability of mitigating the number of unfulfilled rental requests during peak times.

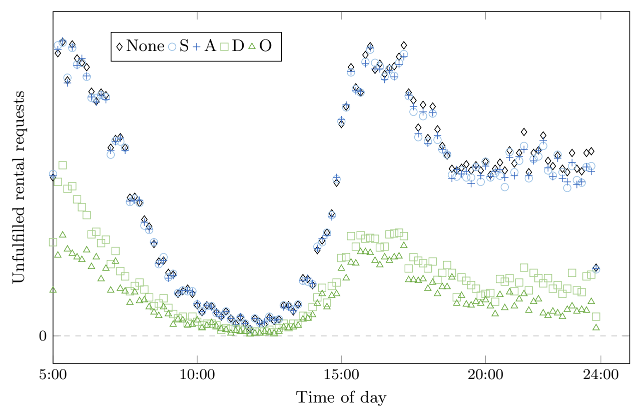


Figure 11: Average number of unfulfilled rental request on a working day.

We observe similar effects as discussed above for the weekend day scenarios. Differences result solely in the magnitude of the solution values. Table 4 summarizes the discussed effects for different key performance indicators (KPIs). It shows for each relocation strategy including the non-relocated

Table 4: Average KPIs for different relocation strategies.

	no modification			start time			arrival time			origin			destination		
	wo	we	Δ	wo	we	Δ	wo	we	Δ	wo	we	Δ	wo	we	Δ
P	78224	84073	7.5	78447	84073	7.2	78447	84321	7.5	86753	92196	6.3	86885	92370	6.3
ft	1862	2033	9.2	1885	2054	9.0	1884	2052	8.9	2615	2845	8.8	2505	2757	10.1
R	-	-	-	24	26	8.3	23	26	13.0	733	805	9.8	676	760	12.4
U	6.57	7.09	7.9	6.59	7.12	8.0	6.59	7.13	8.2	7.57	8.06	6.5	7.45	7.95	6.7
U^I	7.23	7.54	4.3	7.05	7.54	7.0	7.01	7.41	5.7	7.63	8.09	6.0	7.91	8.28	4.7
I	57.65	7.54	-86.9	39.5	35.65	-9.7	37.35	24.3	-34.9	3.85	2.65	-31.2	36.05	23.8	-34.0

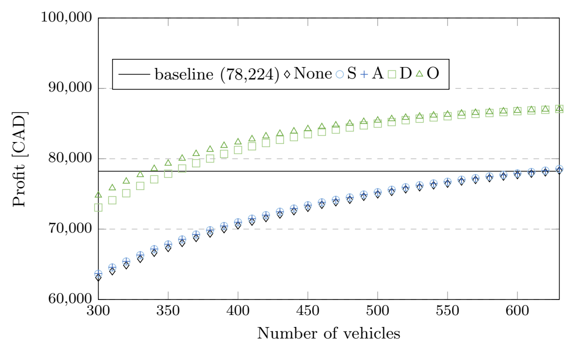
All reported values are average values for the respective scenario set. Abbreviations hold as follows: P–profit; T–fulfilled trips per day; R–number of relocations per day; U–average utilization per car [h]; U^I–average utilization per car [h] excluding idling cars; I–idling cars per day; wo–set of working day scenarios; we–set of weekend day scenarios; Δ–increase from wo to we [%].

scenario, the average profit (P), the number of fulfilled trips per day (T), the number of relocations per day (R), the average utilization (U), the average utilization excluding idling cars (U^I), and the number of idling cars, i.e., the number of cars that are not used during a whole planning period (I), for both the set of working day scenarios (wo) and the set of weekend day scenarios (we). As can be seen, the order of the strategies in terms of their impact remains the same for all KPIs for both working day and weekend day scenarios. In general, all KPIs but the number of idling cars increase on a weekend day compared to a working day, because the weekend day scenarios show a larger number of total requests. Intuitively, the number of idling cars decreases accordingly. Interestingly, the origin-modification strategy shows a significantly lower number of idling cars compared to all other strategies for both scenario sets. This highlights that the origin-modification strategy exploits the relocation potential most exhaustively, which allows for the best overall performance.

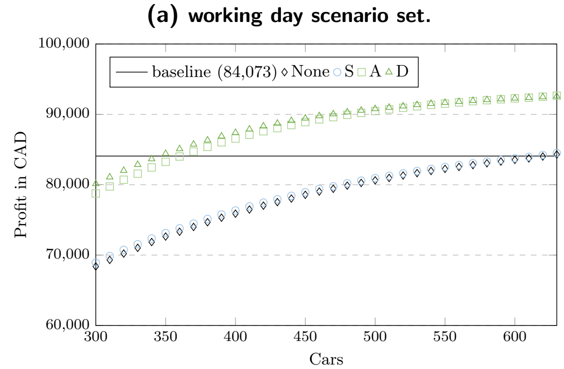
5.2 Sensitivity analysis

Table 4 suggests that the different relocation strategies show a robust behavior for different scenarios with increasing rental requests. However, such a conclusion cannot be generalized solely based on Table 4. To verify this hypothesis, we conduct additional sensitivity analyses, analyzing *i*) the impact of a varying fleet size for stable demand scenarios, and *ii*) the impact of varying demands for a stable fleet size.

Figure 12 shows the impact of a varying fleet size for both the working day and the weekend day scenario set by comparing the mean profit for each relocation strategy to the scenario without relocations. We vary the fleet size from 300 to 630 cars with a step width of 10. As can be seen, all relocation strategies show a robust behavior and an amplitude offset between the working day and weekend day scenarios, as already suggested in Table 4. Remarkably, when utilizing a spatial relocation strategy, one can decrease the original fleet size from 630 to 350 vehicles without lowering the resulting profit.



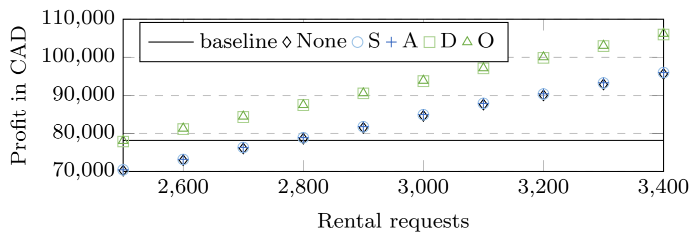
(a) working day scenario set.



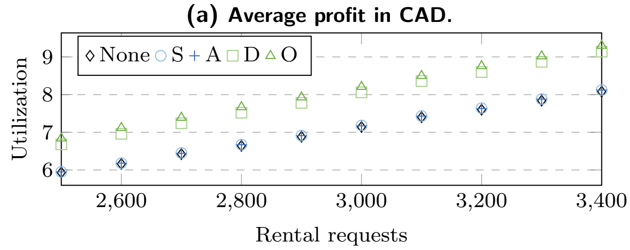
(b) weekend day scenario set.

Figure 12: Mean profit for each relocation strategy depending on the fleet size.

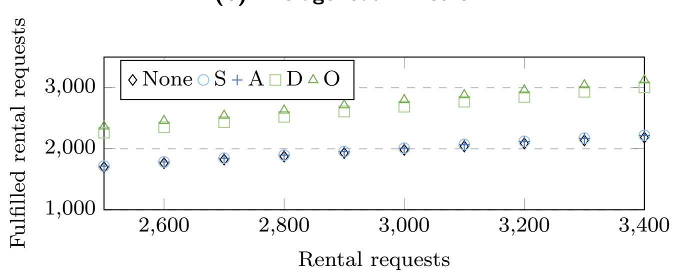
Figure 13 and Figure 14 detail the volatility of the KPIs as discussed in Table 4 for the original fleet size of 630 cars and a varying number of requests for both the working day and the weekend day scenarios. To this end, we vary the number of requests in between 2,500 and 3,400 requests, using a step width of 100. Again, we report the average values over all scenarios. As can be seen, the increase of a KPI correlates with an increase in rental requests in all cases. Moreover, the figures validate the hierarchy between spatial and temporal relocation strategies, the spatial relocation strategies outperform the temporal strategies for all KPIs.



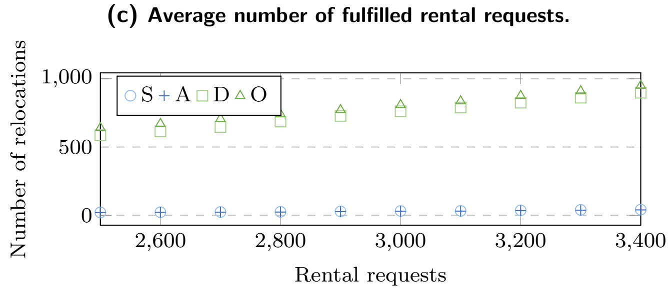
(a) Average profit in CAD.



(b) Average load in hours.

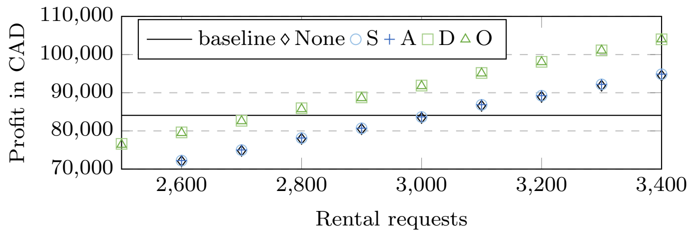


(c) Average number of fulfilled rental requests.

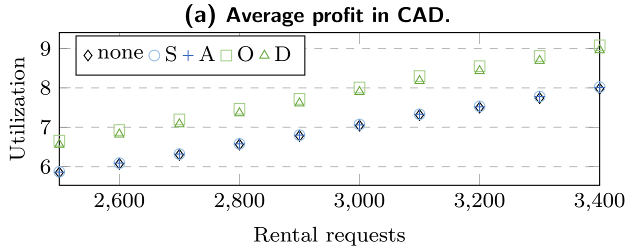


(d) Average number of strategy applications.

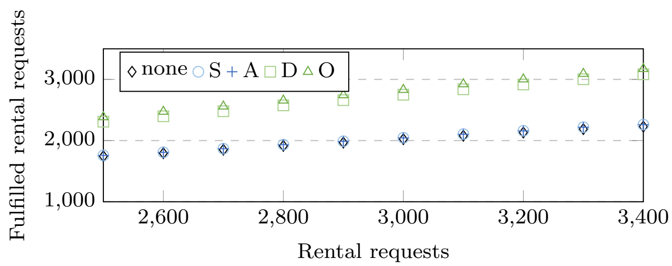
Figure 13: Working day scenario set KPIs depending on the number of trips.



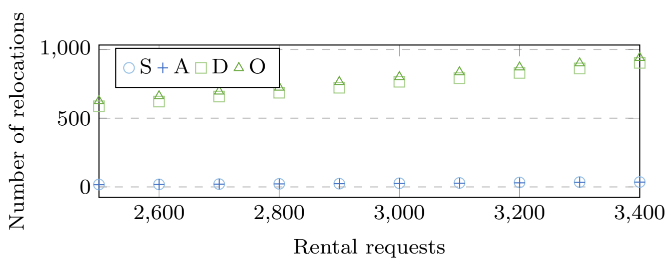
(a) Average profit in CAD.



(b) Average load in hours.



(c) Average number of fulfilled rental requests.



(d) Average number of strategy applications.

Figure 14: Weekend day scenario set KPIs depending on the number of trips.

6 Conclusion and outlook

In this paper, we studied the impact of user-based relocation strategies for FFCSs. We formalized the underlying planning problem as an integer program. We then developed a graph reformulation, which allows to solve this planning problem optimally as a k-disjoint shortest path problem in polynomial time. This algorithm provides a good algorithmic performance for large-scale problems. Moreover, we derived a case study based on real-world data from car2go Vancouver. We applied our algorithm to this case study, assuming perfect information. By so doing, we determined an upper bound on the improvement potential of user-based relocation strategies and derived the following two main insights: first, user-based relocation strategies may improve the utilization of a car sharing fleet significantly, yielding an up to 40% increase in covered trips, accompanied with a 10% increase in the operators profit. Second, spatial relocation strategies, i.e., modifying a customer's origin or destination, significantly outperform temporal relocation strategies. These findings are robust across working day and weekend day scenarios, for varying vehicle fleet sizes, and for a varying number of customer demands.

This work assumed a perfect information setting, thus focusing on the maximum improvement potential that user-based relocation strategies can create. This assumption opens the field for further research. Based on the analyzed results, it appears promising to lift the proposed algorithmic framework from its deterministic offline setting to its (stochastic) online counterpart, which makes the algorithm applicable in practice. Here, (stochastic) receding horizon approaches or model predictive control algorithms provide a good starting point to develop new algorithms. In addition, one may consider to incorporate the user acceptance behavior into such an online algorithm, e.g., by additional stochastic modeling or via reinforcement learning. Apart from these methodological avenues, applying our algorithm to additional case studies may reveal further managerial insights.

References

Matthew Barth and Michael Todd. Simulation model performance analysis of a multiple station shared vehicle system. Transportation Research Part C: Emerging Technologies, 7(4):237–259, 1999.

Matthew Barth, Michael Todd, and Lei Xue. User-based vehicle relocation techniques for multiple-station shared-use vehicle systems: Transportation research board 80th annual meeting. In Proc. Transportation Research Board Annual Meeting, CD-ROM, Washington DC, 2004.

Burak Boyacı, Konstantinos G. Zografos, and Nikolas Geroliminis. An optimization framework for the development of efficient one-way car-sharing systems. European Journal of Operational Research, 240(3):718–733, 2015.

Maurizio Bruglieri, Alberto Colorni, and Alessandro Luè. The vehicle relocation problem for the one-way electric vehicle sharing: An application to the milan case. Procedia - Social and Behavioral Sciences, 111: 18–27, 2014.

Kenneth Button. City management and urban environmental indicators. Ecological Economics, 40(2):217–233, 2002.

E. M. Cepolina and A. Farina. Urban car sharing: An overview of relocation strategies. WIT Transactions on The Built Environment, 128:419–431, 2012.

Nelson D. Chan and Susan A. Shaheen. Ridesharing in north america: Past, present, and future. Transport Reviews, 32(1):93–112, 2012.

M. Clemente, M. P. Fanti, G. Iacobellis, M. Nolich, and W. Ukovich. A decision support system for user-based vehicle relocation in car sharing systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems, PP(99):1–14, 2017.

Monica Clemente, Maria Pia Fanti, Agostino M. Mangini, and Walter Ukovich. The vehicle relocation problem in car sharing systems: Modeling and simulation in a petri net framework. In José-Manuel Colom and Jörg Desel, editors, Application and Theory of Petri Nets and Concurrency, pages 250–269, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-38697-8.

Gonçalo Correia and José Manuel Viegas. Carpooling and carpool clubs: Clarifying concepts and assessing value enhancement possibilities through a stated preference web survey in lisbon, portugal. Transportation Research Part A: Policy and Practice, 45(2):81–90, 2011.

Angela Di Febbraro, Nicola Sacco, and Mahnam Saeednia. One-way carsharing. Transportation Research Record: Journal of the Transportation Research Board, 2319:113–120, 2012.

Angela Di Febbraro, Nicola Sacco, and Mahnam Saeednia. One-way car-sharing profit maximization by means of user-based vehicle relocation. IEEE Transactions on Intelligent Transportation Systems, pages 1–14, 2018.

Wei Fan, Randy Machemehl, and Nicholas Lownes. Carsharing: Dynamic decision-making problem for vehicle allocation. Transportation Research Record: Journal of the Transportation Research Board, 2063:97–104, 2008.

Claudio Gambella, Enrico Malaguti, Filippo Masini, and Daniele Vigo. Optimizing relocation operations in electric car-sharing. Omega, 81:234–245, 2018.

Fabio Grazi and Jeroen C. J. M. van den Bergh. Spatial organization, transport, and climate change: Comparing instruments of spatial planning and policy. Ecological Economics, 67(4):630–639, 2008.

S. Herrmann, F. Schulte, and S. Voß. Increasing acceptance of free-floating car sharing systems using smart relocation strategies: A survey based study of car2go hamburg. In Rosa G. González-Ramírez, F. Schulte, S. Voß, and J. A. Ceroni Díaz, editors, Computational Logistics, pages 151–162, Cham, 2014. Springer International Publishing. ISBN 978-3-319-11421-7.

Donald B Johnson. Efficient algorithms for shortest paths in sparse networks. Journal of the Association for Computing Machinery, 24(1):1–13, 1977.

Diana Jorge and Gonçalo Correia. Carsharing systems demand estimation and defined operations: a literature review. European Journal of Transport and Infrastructure Research, 13(3):201–220, 2013.

Alvina Kek, Ruey Cheu, and Miaw Chor. Relocation simulation model for multiple-station shared-use vehicle systems. Transportation Research Record: Journal of the Transportation Research Board, 1986:81–88, 2006.

Rahul Nair and Elise Miller-Hooks. Fleet management for vehicle sharing operations. Transportation Science, 45(4):524–540, 2010.

Stefan Paschke, Milos Balac, and Francesco Ciari. Implementation of vehicle relocation for carsharing services in the multi-agent transport simulation matsim. In TRB 96th Annual Meeting Compendium of Papers, 2017.

Frederik Schulte and Stefan Voß. Decision support for environmental-friendly vehicle relocations in free- floating car sharing systems: The case of car2go. Procedia CIRP, 30:275–280, 2015.

J. W. Suurballe. Disjoint paths in a network. Networks, 4(2):125–145, 1974.

M. Tsao, R. Iglesias, and M. Pavone. Stochastic model predictive control for autonomous mobility on demand. In Proc. IEEE Int. Conf. on Intelligent Transportation Systems, Maui, Hawaii, November 2018. URL https://arxiv.org/pdf/1804.11074.pdf. In Press. Extended Version, Available at https://arxiv.org/pdf/1804.11074.

Kentaro Uesugi, Naoto Mukai, and Toyohide Watanabe. Optimization of vehicle assignment for car sharing system. In Knowledge-Based Intelligent Information and Engineering Systems: 11th Int. Conf. KES 2007, XVII Italian Workshop on Neural Networks, Vietri sul Mare, Italy, September 12-14, 2007. Proc. Part II, 2007.

S. Weikl and K. Bogenberger. Relocation strategies and algorithms for free-floating car sharing systems. IEEE Intelligent Transportation Systems Magazine, 5(4):100–111, 2013.