A Computational Approach to Quantify the Benefits of Ridesharing for Policy Makers and Travellers

Filippo Bistaffa*, Christian Blum*, Jesús Cerquides*, Alessandro Farinelli†, and Juan Rodríguez-Aguilar*

*IIIA-CSIC, 08193 Cerdanyola, Spain

†Department of Computer Science, University of Verona, 37134 Verona, Italy

Emails: *{filippo.bistaffa,christian.blum,cerquide,jar}@iiia.csic.es, †alessandro.farinelli@univr.it

Abstract—Peer-to-peer ridesharing enables people to arrange one-time rides with their own private cars, without the involvement of professional drivers. It is a prominent *collective intelligence* application producing significant benefits both for individuals (reduced costs) and for the entire community (reduced pollution and traffic). Despite these very promising potential advantages, the percentage of users who currently adopt ridesharing solutions is very low, well below the *adoption rate* required to achieve said benefits. One of the reasons of this insufficient engagement by the public is the lack of effective incentive policies by regulatory authorities, who are not able to estimate the costs and the benefits of a given ridesharing adoption policy. Here we address these issues by (i) developing a novel algorithm that makes large-scale, real-time peer-to-peer ridesharing technologically feasible; and (ii) exhaustively quantifying the impact of different ridesharing scenarios in terms of *environmental benefits* (i.e., reduction of CO_2 emissions, noise pollution, and traffic congestion) and *quality of service* for the users. Our analysis on a real-world dataset shows that major societal benefits are expected from deploying peer-to-peer ridesharing depending on the trade-off between environmental benefits and quality of service. Results on a real-world dataset show that our approach can produce reductions up to a 70.78% in CO_2 emissions and up to 80.08% in traffic congestion.

Index Terms—ridesharing, collective intelligence, environmental benefits, policy making, smart cities, online stochastic combinatorial optimisation, integer linear programming

I. INTRODUCTION

With the growing popularity of the shared economy, ridesharing (carpooling) services are called to transform urban mobility [1]. First, as argued by Alonso *et al.* [1] and Fontaras *et al.* [2], shared mobility is expected to have major environmental and economic impacts by reducing pollution (e.g., CO_2 emissions and noise pollution), traffic congestion, and energy consumption. Second, as noted by Ostrovsky *et al.* [3], ridesharing is said to become even much more attractive in a future world of self-driving cars, and spur a transition from solo driving to mass transit. Thus, the advent of autonomous vehicles in transportation is further encouraging the study of ridesharing solutions, such as *shared autonomous vehicles* (SAVs) [3], [4], [5]. Along these lines, the growing efforts in research and technology with respect to *mobility on demand* (MoD) systems, together with novel incentive policies for shared mobility [6] and efficient market designs [3], will play a crucial role for the exploitation of the full potential of ridesharing, and for the revolution of the current practice in personal transportation.

Despite its major potential benefits, ridesharing is nowadays still far from being widely used (according to the US Census Bureau, only around 9.3% of commuters in the US carpooled to work, compared to 76.4% who drove alone [7], [8]). As argued by Chan *et al.* [8], because of the growing concerns about climate change, congestion, and oil dependency, further research is needed to understand the benefits of ridesharing, while also considering time and privacy requirements of users [9], [10], [11]. Such understanding is fundamental for policy making in the context of shared mobility. Furthermore, Chan *et al.* [8] notice that an important aspect is to find out the *adoption rate* that allows to achieve such target societal benefits. Finally, it is crucial for such analysis tools to be able to deal with complex, large-scale, dynamic scenarios.

Not surprisingly, the concept of ridesharing has recently been receiving significant attention, both in the transportation industry (thanks to companies such as *UberPool*, *Lyft*, and *Maramoja*) and in academia [1], [9], [12], [13], [14], [15]. Despite the abundant literature on fleet management for MoD systems, the problem of servicing multiple rides with a single trip in large-scale dynamic scenarios has only very recently been considered. The approaches by Alonso *et al.* [1] and Pelzer *et al.* [12] introduce a carpooling matching algorithm, whose objective is to carpool passengers with the aim of maximising *quality of service* (QoS), i.e, minimising the delay experienced by the users. Riley *et al.* [15] recently improved upon the work of Alonso *et al.* [1] by proposing a column-generation algorithm to minimise wait times while guaranteeing short travel times and service for each customer. Bicocchi *et al.* [14] pursue the same objective, but they propose a recommendation-based approach that exploits historical data to compute potential matchings and minimise the inconvenience for the users. On the other hand, Rigby *et al.* [13] propose to shift from the usual discrete representation to a continuous space-time representation of vehicle accessibility to provide a client with a more realistic choice set. Despite being a significant contribution for *business-to-consumer ridesharing*, these works do not take into account the societal and environmental benefits of ridesharing, and, hence, they do not help policy makers in making the aforementioned assessments in the context of incentives design. Finally, Santi *et al.* [9] evaluate the impact of ridesharing in terms of trade-off between the benefit and the passenger discomfort. However, such an approach is computationally limited to two riders (three with heuristics) and lacks the scalability needed for large-scale ridesharing [1].

Against this background, here we devise and study a different type of ridesharing approach, with the objective of overcoming the mentioned drawbacks of the state of the art. Our algorithm allows to carpool passengers considering not only QoS, which is the common objective in the state of the art, but also *environmental benefits*. To the best of our knowledge, our ridesharing algorithm is the first one that allows to customise the trade-off of those two objectives when forming shared rides. Instead of a matching ridesharing service, we introduce an algorithm for online, city-scale, *peer-to-peer ridesharing*, which enables people to arrange one-time rides at short notice with their own private cars, without the involvement of professional drivers, hence providing an alternative to traditional taxi services. Along these lines, our work follows the definition of ridesharing established by Delhomme and Gheorghiu [16], i.e., "an arrangement where two or more people [...] share the use of a *privately owned car* for a trip." Thus, our algorithm allows the grouping of carpooling partners into cars, instead of matching passengers to cars. We show that our peer-to-peer ridesharing problem can be cast as a particular type of *online stochastic combinatorial optimisation* problem, thus being formally different from the matching problem described by the above mentioned works [1], [12], [13], [14], [15]. Our empirical results on the New York City taxi dataset [17] show that despite being challenged with exploring a large decision space, our algorithm is capable of computing solutions in a very limited amount of time (per minute) that are very close to optimality. In conclusion, our algorithm makes large-scale, real-time peer-to-peer ridesharing technologically feasible. Furthermore, we also conducted an extensive evaluation of the computational aspects of our solution approach. Results show that our online approach computes solutions with an average *competitive ratio* of 88.3% (i.e., *wrt* the optimal offline solution assuming complete knowledge of the future) for problem sizes that allow to compute such an optimal solution in a feasible amount of time. On large-scale instances (i.e., when the optimal cannot be computed), our online approach provides solutions whose quality is comparable with those of our approximate offline approach.

The rest of the paper is organised as follows: Section II formally defines the peer-to-peer ridesharing problem, while Section III discusses our solution approach. Section IV presents our experimental evaluation on real-world data. Finally, Section V concludes the paper and outlines future work.

II. THE PEER-TO-PEER RIDESHARING PROBLEM

We define the *peer-to-peer ridesharing* (P2P-RS) problem as an *online stochastic scheduling* problem [18] over a discrete time horizon $H = [1, h]$. Our problem takes place in an area divided in n zones, i.e., $Z = \{1, \ldots, n\}$.[1] At each step $t \in H$, the system receives a (possibly empty) set of requests R_t. Each request $r \in R_t$ is a tuple $\langle i, j, d, \delta \rangle$ characterised by a starting zone $i \in Z$ and a destination zone $j \in Z$, a Boolean value

$d \in \mathbb{B}$ indicating whether the corresponding commuter has a car or not (i.e., whether it is a driver or not), and the maximum time $\delta \in H$ the commuter is willing to wait to be assigned to a car. Formally, $r \in \mathcal{R} = Z \times Z \times \mathbb{B} \times H$. Hence, the sequence $\langle R_1, \ldots, R_t, \ldots, R_h \rangle$ represents the input of the problem.

We consider the following hard constraints in order to characterise if a set of requests $S \subseteq \mathcal{R}$ is a *feasible car*. First, we assume our cars have k seats including the driver.[2] S is said to be *feasible* when (i) it does not contain more than k requests, (ii) the requests are time-compatible (that is, there is a time t where every request in the car is active)[3] and (iii) there is at least one driver per car. Formally,

$$F(S) = |S| \leq k \ \wedge \ \min_{r_\alpha \in S}(t_\alpha + \delta_\alpha) \geq \max_{r_\beta \in S} t_\beta \ \wedge \ \bigvee_{r_\gamma \in S} d_\gamma,$$

where t_α (resp. t_β) is the time step when r_α (resp. r_β) was accepted. Given $R \subseteq \mathcal{R}$, the set of all feasible cars that can be generated from R is $\mathcal{F}(R) = \left\{ S \in 2^R \mid F(S) \right\}$.

Within our model, we measure the performance of a given car S both in terms of *environmental benefit* and *quality of service*. In this paper, we consider three environmental costs, i.e., CO_2 emissions, noise pollution, and traffic congestion. Following Fontaras *et al.* [2] and van Blokland and Peeters [19], we consider CO_2 emissions and noise pollution associated to a given car to be *proportional* to its *travelled distance*. The relative benefit of forming a car S is given by

$$E_{CO_2}(S) = E_{\text{noise}}(S) = |S| \cdot \frac{\sum_{r \in S} d(\{r\}) - d(S)}{\sum_{r \in S} d(\{r\})}, \quad (1)$$

where $d : 2^\mathcal{R} \to \mathbb{R}_{\geq 0}$ denotes the distance travelled by a given car while driving through its optimal route, as detailed in section II-A. On the other hand, we consider the traffic congestion associated to a given car as a positive constant. Thus, by forming a car with $|S|$ passengers we spare $|S| - 1$ cars compared to the case with no ridesharing (where $|S|$ cars are formed). Consequently we define

$$E_{\text{traffic}}(S) = |S| - 1. \quad (2)$$

Furthermore, we define $Q : 2^\mathcal{R} \to \mathbb{R}_{\leq 0}$, namely the *quality of service* associated to a given car S, as

$$Q(S) = - \sum_{r \in S} \overbrace{\frac{t_r - t_r^*}{t_r}}^{\text{in-car delay}}, \quad (3)$$

where t_r and t_r^* are the travel times associated to request r with and without ridesharing, respectively. Since t_r^* is the optimal travel time associated to $r = \langle i, j, d, \delta \rangle$ (i.e., obtained by driving through the shortest path from zone i to zone j), then $0 \leq \frac{t_r - t_r^*}{t_r} \leq 1$, hence $-|S| \leq Q(S) \leq 0$.

Notice that the absolute values of all the above benefits are in the range $[0, |S|]$, since they are all measured *relative to the case where no ridesharing is employed*.

---

[1] This choice is motivated by the fact that—for privacy reasons—the dataset we consider only reports starting and destination zones (i.e., neighbourhoods) of requests, instead of the actual coordinates.

[2] Extending our model to consider a heterogeneous fleet of cars is straightforward and it is subject of future work.

[3] In other words, requests in S are time-compatible if the first request in S to expire is still active when the latest request in S enters the system.

Finally, we define $V : 2^{\mathcal{R}} \to \mathbb{R}$ as the total value (i.e., considering environmental benefits and the quality of service) associated to a given car S as

$$V(S) = \rho_{\text{CO}_2} \cdot E_{\text{CO}_2}(S) + \rho_{\text{noise}} \cdot E_{\text{noise}}(S) + $$
$$\rho_{\text{traffic}} \cdot E_{\text{traffic}}(S) + \rho_{\text{QoS}} \cdot Q(S), \quad (4)$$

where ρ_{CO_2}, ρ_{noise}, and ρ_{traffic} measure the importance of the corresponding environmental benefit and ρ_{QoS} that of the QoS.

A. Distance Computation

We assume the existence of a n-by-n matrix D, such that $D_{i,j}$ represents the length of the shortest path from zone i to zone j. We then consider the set $\mathcal{L}_S$ of all sequences of pick-up and drop-off points of the passengers of S that satisfy the following feasibility constraints [20]:

- the first (resp. the last) point of $L \in \mathcal{L}_S$ is the pick-up (resp. drop-off) point of the currently designated driver;
- for each request $r \in S$, the pick-up point of r precedes the drop-off point of r in $L \in \mathcal{L}_S$.

The optimal route for S, namely L_S^*, is then defined as the one that minimises the sum of the lengths $D_{i,j}$ for every couple (i,j) of subsequent points in L, for every route $L \in \mathcal{L}_S$, and considering all the possible drivers in S. Finally, $d(S)$ is defined as the sum of such lengths for the optimal route L_S^*.

B. Online Problem Definition

We now define the online version of the problem, where the set of requests R_t is not known a priori but is revealed online as time proceeds. We first define the set All_t of all the requests received up to time step t as $\text{All}_t = \bigcup_{1 \leq \tau \leq t} R_\tau$. Our task is to arrange, at each time step t, a (possibly empty) set $\mathcal{S}_t$ of non-overlapping cars among the current set of active requests:

$$\mathcal{S}_t \subseteq \mathcal{F}(\text{Pool}_t), \quad \text{where } \text{Pool}_t = \text{All}_t \setminus \overbrace{\bigcup_{1 \leq \tau < t} \bigcup_{S \in \mathcal{S}_\tau} S}^{\text{past cars (serviced)}}, \quad (5)$$

such that, for each couple of cars $S, S' \in \mathcal{S}_t$, $S \cap S' = \emptyset$.

Our definition of the online problem follows the one by Hentenryck and Bent [18], where each set of requests R_t is drawn from a distribution over sets of requests $\mathcal{I}_t$. For simplicity we assume that at each time step t, each sample drawn from $\mathcal{I}_t$ contains the same number of requests, which we refer to as the request rate. Thus, for a fixed request rate ω, each R_t sampled from $\mathcal{I}_t$ returns a set with ω requests, where each individual request $r \in R_t$ can be seen as the realisation of a random variable ξ_t whose distribution is specified by $\mathcal{J}_t$. Formally, $\mathcal{J}_t : \mathcal{R} \to [0,1]$ is a joint probability mass function among $\mathcal{Z}_t : Z \times Z \to [0,1]$ such that $\mathcal{Z}_t(i,j)$ is the frequency of the requests departing from zone i and arriving at zone j at step t in the historical data [17], a Bernoulli distribution $\mathcal{B}_t : \mathbb{B} \to [0,1]$, and a probability mass function $\mathcal{H}_t : H \to [0,1]$. The P2P-RS problem requires to compute the optimal sequence of sets of cars $\langle \mathcal{S}_1^*, \ldots, \mathcal{S}_t^*, \ldots, \mathcal{S}_h^* \rangle$ where each $\mathcal{S}_t^* \subseteq \mathcal{F}(\text{Pool}_t)$ is a subset of the feasible cars that can be formed from Pool_t selected so as to maximise the expected benefit assuming that unknown future sets of requests are sampled from $\mathcal{I}_{t+1}, \ldots, \mathcal{I}_h$.

III. SOLUTION APPROACH

Let us assume, for a moment, that the sequence of requests $\langle R_1, \ldots, R_t, \ldots, R_h \rangle$ is fully known a priori for the whole planning horizon. Moreover, let $R^{\cup} := \bigcup_{t=1}^h R_t$. Then, the P2P-RS problem can be expressed in terms of an integer linear program (ILP) in the following way. For each $S \in F(R^{\cup})$ we make use of a binary variable x_S. The P2P-RS is then modelled as a *weighted set packing* problem [21] as follows:

$$\max \sum_{S \in \mathcal{F}(R^{\cup})} x_S \cdot V(S) \quad (6)$$
$$\text{such that } x_S + x_{S'} \leq 1 \quad \forall \mathcal{F}(R^{\cup}) : S \cap S' \neq \emptyset.$$

Assuming that we know R_t ($t = 1, \ldots, h$) beforehand is, of course, not realistic. Moreover, given the magnitude of the problem that we consider—that is, approx. 400 requests per minute—it would not even be possible to generate the corresponding ILP, let alone solving it.

Nevertheless, it is possible to compute an approximate solution for the P2P-RS by means of the following *approximate offline* approach. In the field of combinatorial optimisation, a well-known technique for approximately solving large-scale binary ILPs consists in (1) removing those variables from the model that, a priori, do not seem to help for the generation of good solutions, and (2) passing the reduced ILP to a solver in order to get the best solution possible in the available computation time. Such techniques were used, for example, in [22], [23], [24], [25]. In the specific case of our approximate offline approach, we iteratively apply a probabilistic greedy heuristic to generate feasible cars. All the cars found in these solutions are added to an initially empty set $\hat{\mathcal{F}}$. Once this process finishes, the ILP stated above is solved on the basis of $\hat{\mathcal{F}}$ instead of $\mathcal{F}(R^{\cup})$. Given a total time budget of TB CPU seconds, our approach allows $TB \cdot (1 - \gamma)$ seconds for the probabilistic generation of feasible cars, and the rest of the time—that is, $TB \cdot \gamma$ seconds—for solving the resulting ILP model. Hereby, $0 < \gamma < 1$ is a parameter of the approach.

The natural greedy approach for the P2P-RS problem starts from an initially empty set of cars, i.e., an empty solution. Then, at each step, the following three actions are carried out:

1) Select the car $S^* \in \mathcal{F}(R^{\cup})$ with the highest $V(\cdot)$-value.
2) Car S^* is added to the current solution.
3) Car S^* and all those cars that have at least one request in common with S^* are removed from $\mathcal{F}(R^{\cup})$.

The disadvantage of this greedy approach is that $\mathcal{F}(R^{\cup})$ must be enumerated, which is—as already mentioned above—not feasible due to the massive number of possible cars. Therefore, the identification of the car with the highest $V(\cdot)$-value (see list point 1 above) is replaced with the following mechanism for generating a car from the remaining set of requests $\overline{R}^{\cup}$, i.e., requests that are not already covered by cars in the solution under construction. Note that, when the current solution is empty, it holds that $R^{\cup} = \overline{R}^{\cup}$. For the generation of a car, first all couples of requests from $\overline{R}^{\cup}$ are evaluated. Then, with probability $d_{\text{rate}} < 1$, the best couple according to the $V(\cdot)$-value is selected. Otherwise, one of the best l_{size} couples is chosen uniformly at random. While the $V(\cdot)$-value of the currently generated car can be

Algorithm 1 $\text{PG}^2(R, d_{\text{rate}}, l_{\text{size}})$

Input: A set of requests R
Output: A set of cars $\mathcal{S}$
1: $\mathcal{S} \leftarrow \emptyset$
2: **repeat**
3:      $\mathcal{C} \leftarrow \{\{r_i, r_j\} \mid r_i, r_j \in R, r_i \neq r_j\}$ {Couples in R}
4:      $S^* \leftarrow \text{PROBABILISTICBEST}(\mathcal{C}, d_{\text{rate}}, l_{\text{size}})$
5:      **if** $V(S^*) > 0$ **then**
6:          **for all** $i \in [3, k]$ **do** {For all sizes up to k}
7:              $\mathcal{C} \leftarrow \{S^* \cup \{r\} \mid r \in R\}$ {Enlarge best car}
8:              $S \leftarrow \text{PROBABILISTICBEST}(\mathcal{C}, d_{\text{rate}}, l_{\text{size}})$
9:              **if** $V(S) > V(S^*)$ **then**
10:                 $S^* \leftarrow S$
11:             **else**
12:                 **break**
13:         $\mathcal{S} \leftarrow \mathcal{S} \cup \{S^*\}$
14:         $R \leftarrow R \setminus S^*$
15: **until** $V(S^*) \leq 0$
16: **return** $\mathcal{S}$

Algorithm 2 $\text{PROBABILISTICBEST}(\mathcal{C}, d_{\text{rate}}, l_{\text{size}})$

Input: A set of cars $\mathcal{C}$
Input: Determinism rate d_{rate}
Input: Candidate list size l_{size}
Output: Best car in $\mathcal{C}$ with probability d_{rate}, random car among the l_{size} best ones with probability $1 - d_{\text{rate}}$
1: $c \sim \text{Bern}(d_{\text{rate}})$ {Deterministic or probabilistic?}
2: **if** $c = 1$ **then**
3:      $S^* \leftarrow \arg\max_{S \in \mathcal{C}} V(S)$ {Select best car}
4: **else**
5:      $\mathcal{X} \leftarrow$ list of l_{size} cars $\in \mathcal{C}$ with the highest value
6:      $S^* \leftarrow$ random car $\in \mathcal{X}$
7: **return** S^*

---

increased by adding some request, such requests are added one-by-one, in the same random-proportional way. Once this process stops, the generated car is added to the solution under construction, and the corresponding requests are removed from $\overline{R}^{\cup}$. This simplified greedy heuristic is henceforth called PG^2 (Algorithm 1). Remember that after the generation of a solution with PG^2, the corresponding cars are added to $\hat{\mathcal{F}}$.[4] Finally, notice that each application of PG^2 can be safely run in parallel with other applications of PG^2. Hence, it is possible to run μ applications in parallel, with μ being the maximum number of threads supported by the architecture executing our approach. Figure 1 shows an overview of our offline approach.

A. Online Approach

As it is unrealistic to assume that the requests R_t ($t = 1, \ldots, h$) are known beforehand, we tackle the P2P-RS problem by solving a *sequence* of offline problems by means of our approximate offline approach. Note that this is similar to the procedure proposed in [18]. However, the approximate offline approach is applied with two differences. The first one concerns the input. Instead of providing $R^{\cup}$ as input—that is, the whole set of requests concerning the complete planning horizon—at each time step $t = 1, \ldots, h$, only the currently active set of requests (i.e., Pool_t, see Equation 5) is provided as input. Let $\mathcal{S}_t$ be the set of cars returned by the approximate offline approach when given Pool_t as input. The

---

[4]In our experiments, $\hat{\mathcal{F}}$ contains contains less than 0.01% of the variables of $\mathcal{F}$ (i.e., the set of variables corresponding to the optimal ILP formulation).
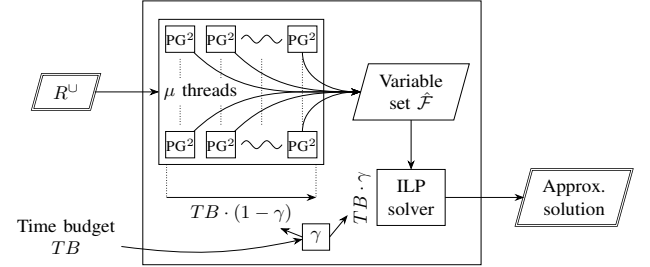

Fig. 1: Overview of our approximate *offline* approach. Double line indicates the input and the output of the algorithm.

second change consists in filtering out immediately profitable cars whose formation could hinder the formation of even better cars in the near future by applying the following *look-ahead reasoning* to $\mathcal{S}_t$: all those cars that do not contain at least one request which would expire in the following time step are removed from $\mathcal{S}_t$. Note that no possible harm is done by such a decision, because the same car might actually be generated again in time step $t + 1$. As a result, we open the possibility to use the requests from discarded cars to form in the near future cars with a larger value, as also noted by Santi *et al.* [9]. A graphical overview of our online approach is in Figure 2.

IV. EXPERIMENTAL ANALYSIS

We evaluate the performance of our approach in terms of environmental benefits and QoS using real data from the publicly available dataset of taxi trips in Manhattan, New York City [17]. This dataset contains for each day the time and location of all of the pick-ups and drop-offs associated to each trip request, in our analysis we select an arbitrarily chosen representative weekend day, i.e., 3 December 2016. From these data, we extract all of the requests (origin and destination within Manhattan) and consider the time of request equal to the time of pick-up. Notice that the dataset—for privacy reasons—only reports the *zone* in which the pick-up and drop-off locations are situated. The shortest paths and travel times between all zones are precomputed with Google Maps and stored in a look-up table. All experiments have been run on a cluster whose computing nodes have two 4-cores 2.26GHz CPUs and 32GB of RAM, using CPLEX 12.5 as ILP solver.[5]

A. Environmental Benefits and QoS

We analyse several metrics to evaluate the effectiveness of our approach in terms of environmental benefits (i.e., average occupancy, distance reduction—which directly translates into CO_2 emissions and noise pollution reduction—and number of trips reduction—which produces a reduction of traffic congestion) and QoS (i.e., average in-car delay) when considering different *adoption rates*. To provide a comprehensive evaluation of our approach, we also measure the average *total* delay experienced by the users, i.e., the sum of the *to-be-assigned* delay, the *to-be-picked-up* delay, and the *in-car* delay. Given an adoption rate $a_{\text{rate}} \in [0, 1]$, we assume that, at each time step, a number of requests proportional

---

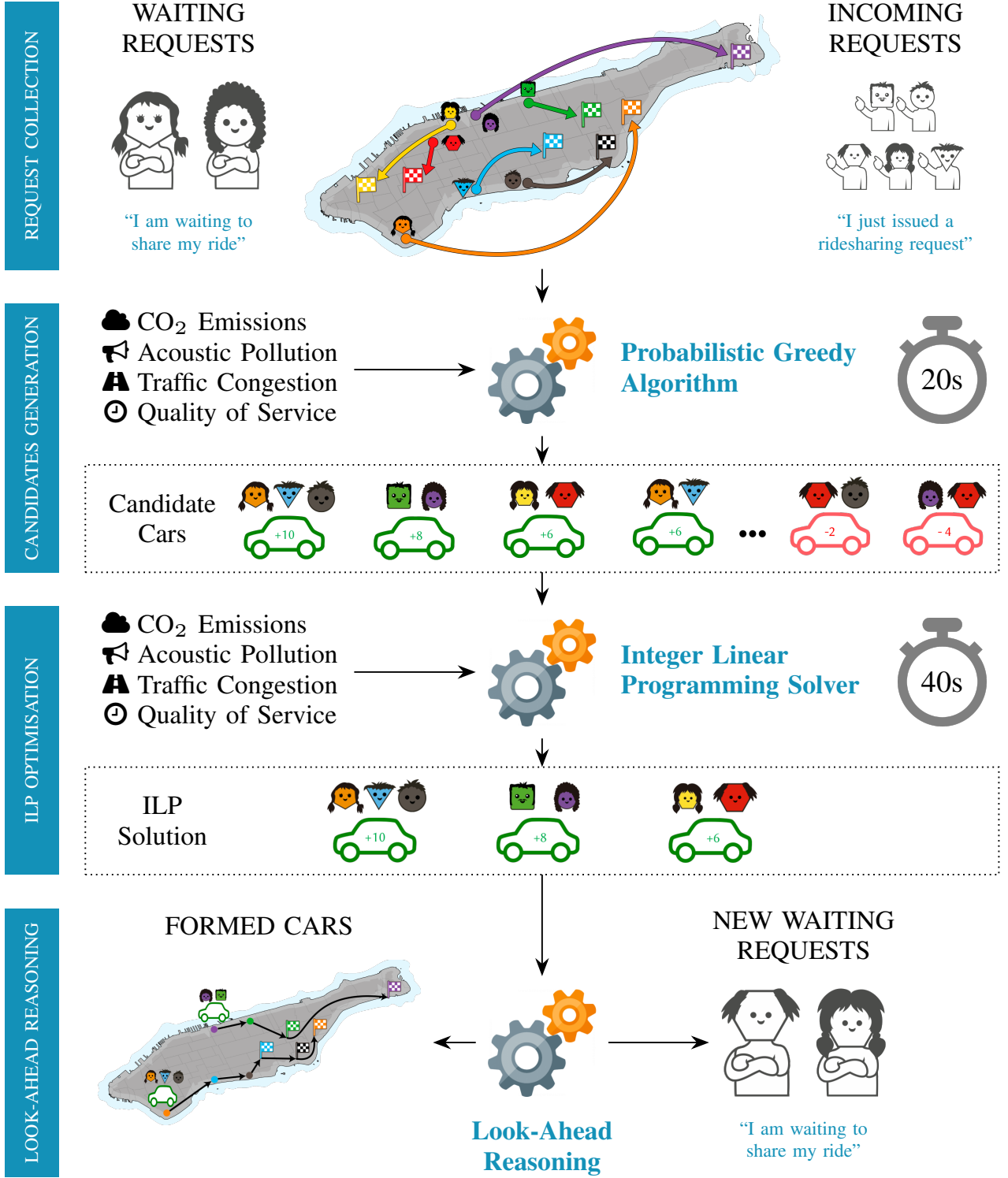[5]Source code available at https://github.com/filippobistaffa/P2P-RS.

Fig. 2: [Best viewed in colour] Graphical overview of our approach to solve the peer-to-peer ridesharing problem. **Request collection (1):** The system receives 5 new request at the current time step, which are added to the current pool of 2 waiting requests. **Candidate car generation (2):** On the basis of such a pool of requests, our probabilistic greedy algorithm generates a set of good candidate cars, aiming at maximising the considered value in terms of environmental benefits and quality of service. **Ridesharing optimisation (3):** Our ILP-based algorithm computes the optimal set of non-overlapping cars on the basis of the previously computed candidates. Notice that both phase 2 and 3 must be completed within the given time budget (20 seconds + 40 seconds = 1 minute, in this example), as we are considering an online problem with very strict time constraints. **Look-ahead reasoning (4):** Finally, our look-ahead reasoning determines which cars will be formed at the current time step by selecting only the ones with end-of-life requests (orange and purple in this example). The remaining requests are re-introduced in the pool for the next time step.
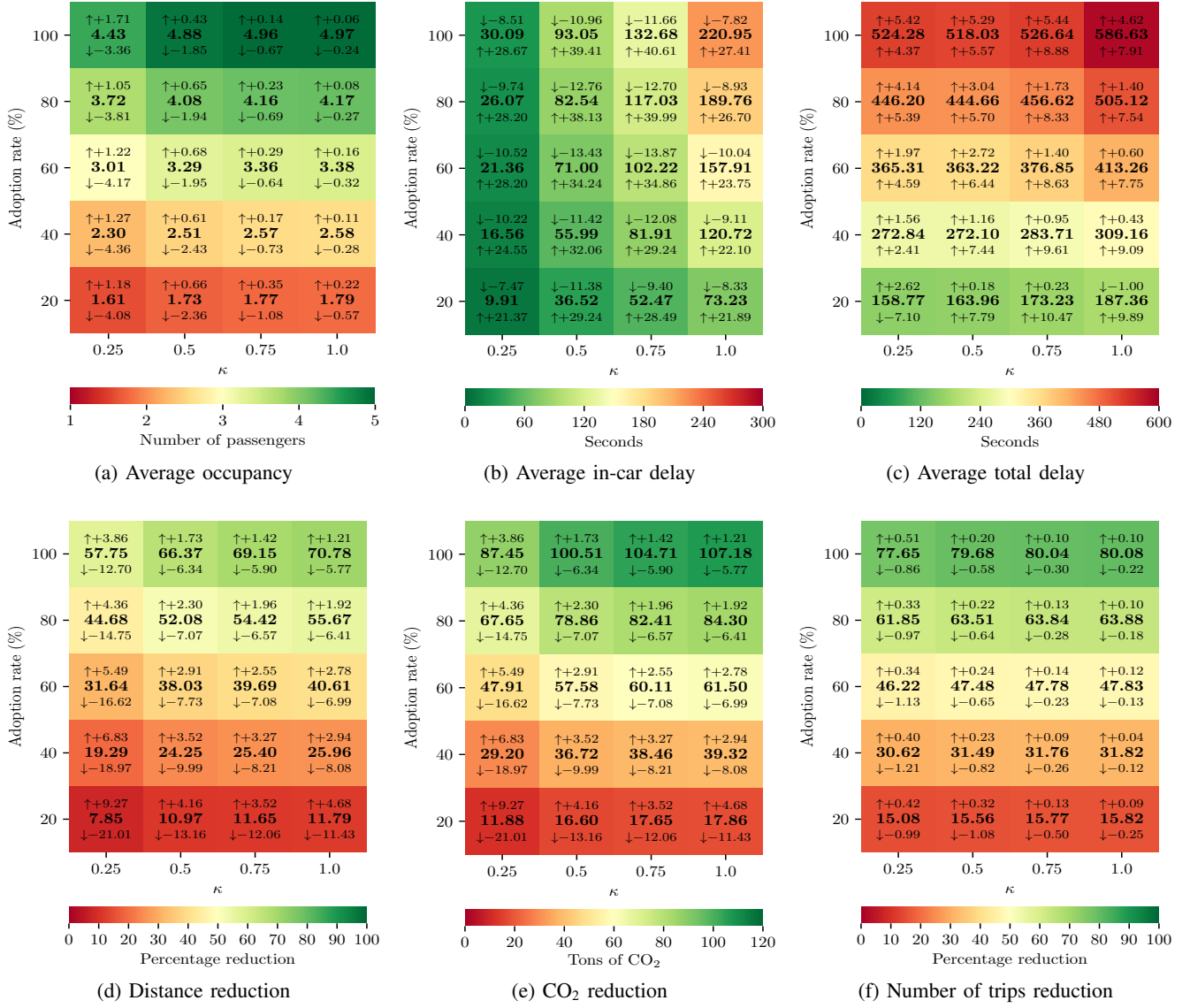
(a) Average occupancy

(b) Average in-car delay

(c) Average total delay

(d) Distance reduction

(e) CO$_2$ reduction

(f) Number of trips reduction

Fig. 3: [Best viewed in colour] Average occupancy (3a), average in-car delay (3b), average total delay (3c), distance reduction (3d), CO$_2$ reduction (3e), and number of trips reduction (3f) in the selected weekend day (03/12/2016), for different values for κ (on the x axis) and different adoption rates (on the y axis). Within each cell, the central **bold** value (which determines the colour of the cell) refers to the case with a maximum waiting time δ of 5 minutes for each request. The value above (resp. below) refers to the case with $\delta = 7$ minutes (resp. 2 minutes), showing the percentage difference *wrt* the central value.

to a_{rate} adopts our ridesharing solution, while the remaining requests do not adopt ridesharing (i.e., each user commutes from its pick-ups to its drop-off zone by driving through the shortest path). In order to simplify the analysis, we introduce a single parameter $\kappa \in [0, 1]$ representing the trade-off between environmental benefits and quality of service. In particular, we set $\rho_{\text{CO}_2} = \rho_{\text{noise}} = \rho_{\text{traffic}} = \kappa$ and $\rho_{\text{QoS}} = 1 - \kappa$. The higher κ the more relevant the environmental benefits to the optimisation. We consider different environmental benefit importances $\kappa \in \{0.25, 0.5, 0.75, 1\}$ and maximum traveller waiting times $\delta \in \{2, 5, 7\}$ (in minutes).

Figure 3 details how the adoption rate, κ, and δ impact on the above mentioned metrics. As expected, our results show a clear *trade-off between environmental benefits and QoS*, i.e., the higher the adoption rate and the environmental benefit

importance κ, the lower the QoS (i.e., the larger the average in-car delay). In other words, a lower adoption rate results in a QoS closer to the optimal one (i.e., no in-car delay) as a greater portion of the population commutes from its pick-up zone to its drop-off zone by driving through the shortest path without ridesharing. For instance, with 100% adoption and grouping commuters to mainly foster environmental benefits ($\kappa = 1$), our approach achieves a reduction of 70.78% on the total travelled distance (corresponding to a reduction of CO$_2$ emissions of 107.18 Tons per day), a reduction of 80.08% on the number of trips, and full occupancy (i.e., an average of 4.97 passengers per car). However, these extraordinary results require that, on average, users are willing to tolerate an in-car delay of 220.95 seconds, and a total delay of 583.61 seconds (less than 10 minutes). On the other hand, when

we mainly focus on QoS (i.e, with 20% adoption rate and $\kappa = 0.25$), the average in-car delay is negligible (less than 10 seconds on average) and the average total delay is less than 2 minutes, but we can still achieve some environmental benefits (a reduction of 7.85% of CO_2 emissions, and a 15.08% reduction concerning the number of trips), showing that our approach can still provide benefits even without impacting on the QoS. The significance of our study can be best understood when considering *intermediate* cases, hence addressing the trade-off between environmental benefits and QoS mentioned above. In fact, our results provide policy makers with an important tool to design *action plans*, where target values are set as goals and some plan is defined to achieve them *within some time frame*. For instance, if a public administration can incentivise users so to have an adoption rate of 40% our results show that it is possible to achieve a reduction of almost 40 Tons of CO_2 per day, corresponding to a 25.96% reduction on the travelled distance, and a 31.82% reduction of the number of trips. On the other hand, if the administration has the ambitious objective of reducing CO_2 emissions (and, hence, the total travelled distance) by 50-60%, it is necessary to design an adequate incentive mechanism for commuters so as to achieve an adoption rate of 80-100%.

To better observe the impact of the adoption rate on the behaviour of our approach, in Figure 4 we report the average in-car delay and the distance reduction for different adoption rates when environmental benefits and QoS are equally important, i.e., $\kappa = 0.5$. Results show that both the considered measures increase when increasing the adoption rate, further highlighting the trade-off between environmental benefits and QoS *wrt* the adoption rate (i.e., we aim at maximising the environmental benefits while minimising the delay). Results also show that the distance reduction increases linearly (red line in Figure 4), while the in-car delay increases less-than-linearly (blue line in Figure 4). This behaviour is desirable: the marginal impact of increasing the adoption rate on the QoS is progressively less significant for higher percentages, while the gain in terms of environmental benefits grows constantly.

Furthermore, we analyse the influence of different waiting times $\delta \in \{2, 5, 7\}$ on the considered performance measures. Specifically, each cell in Figure 3 reports the considered value for $\delta = 5$ in the middle in bold, as it represents our main case of study. The value above (resp. below) refers to the case with $\delta = 7$ minutes (resp. 2 minutes), showing the percentage difference *wrt* the central value. Results show that, in general, a larger δ results in higher environmental benefits and lower average in-car delays. This behaviour is expected, since with a larger δ requests can stay in the pool for a slightly longer period, allowing us to have a wider range of optimisation options and, hence, produce solutions of higher quality. Notice that, once again, we observe a *trade-off*, as better solutions can be provided if users are willing to tolerate a longer initial waiting time. The same behaviour can also be seen in Figure 4, where we represent the three different data series (i.e., $\delta \in \{2, 5, 7\}$) as a ribbon plot (see caption of the figure).

The same experimental evaluation discussed here has been conducted on a working day (i.e., 1 December 2016) obtaining similar results, as discussed in Appendix A.
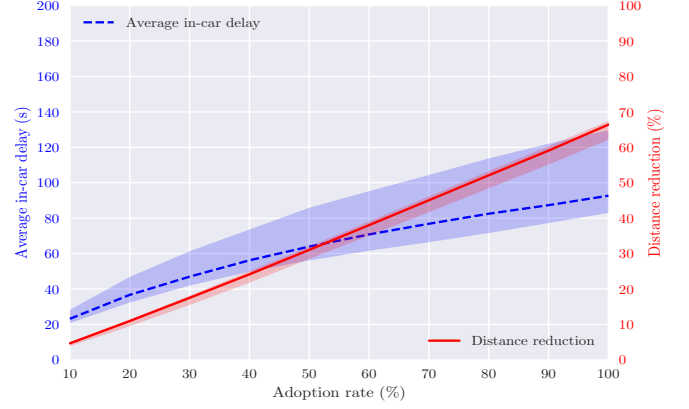

Fig. 4: Average in-car delay and distance reduction achieved in the selected weekend day (3 December 2016) for $\kappa = 0.5$. The line refers to the case with $\delta = 5$ minutes, while, in the distance reduction plot, the shaded area above (resp. below) the line refers to the case with $\delta = 7$ minutes (resp. 2 minutes). In the average in-car delay plot these cases are inverted (i.e., 2 minutes above and 7 minutes below), because a higher (resp. lower) δ results in a lower (resp. higher) delay.

B. Computational Analysis

We now present some additional experimental results focusing on the computational aspects of our approach. Specifically, our main goals are: *(i)* to evaluate the performance and the quality of the solutions of our *offline* approach *wrt* the optimal approach; and *(ii)* to evaluate the performance and the quality of the solutions of our *online* approach *wrt* the offline approach with complete future information.

1) Experimental methodology: We assume a maximum car capacity of $k = 5$, i.e., the number of seats of an average car. Moreover, we assume that each commuter is a driver, in order to account for the most computationally intensive scenario.[6] For the online approach, we consider a time step of 1 minute (which also represents the time budget TB for the offline approximate approach for each time step), and a time horizon h of 10 minutes, unless otherwise stated. Each request is associated to a maximum waiting time of 5 minutes, unless otherwise stated.[7] Since here we are interested in evaluating the behaviour of our algorithmic approach on different problem sizes (i.e., with different request rates ω), we generate realistic synthetic instances based on the real-world dataset detailed above. Specifically, we generate such instances by sampling ω requests at time step t from the joint probability mass function $\mathcal{J}_t$ discussed in Section II. For the small-case scenario, we consider a request rate of 10 and 50 requests per minute. For the large-scale scenario, we consider 100 and 500 requests per minute.[8] We set the parameters of our approximate offline algorithm as follows:

---

[6]If all commuters are not drivers, some cars are not feasible (it is not possible to form a car without a driver), and, hence, the total number of possible cars is smaller.

[7]Our dataset refers to taxi trips, hence it does not contain information about maximum waiting times for the users. We consider 5 minutes as a reasonable maximum waiting time for a ridesharing service.

[8]The actual average request rate in the considered time period is ∼400 requests per minute (with peaks of ∼500).

	Offline *vs* Optimal		Online *vs* Offline	
	Offline	Optimal	Online	Offline
Time horizon (h)	10 time steps	10 time steps	10 time steps	10 time steps
Request rate (ω)	10	10	$\{10, 50, 100, 500\}$	$\{10, 50, 100, 500\}$
Time budget (TB)	$TB^* \cdot \{5 \cdot 10^{-5}, \ldots, 0.1\}$	$+\infty$	1 minute / time step	2 hours
Gener. budget ($1 - \gamma$)	0.74	-	0.74	0.51
Solution budget (γ)	0.26	-	0.26	0.49
Determ. rate (d_{rate})	0.62	-	0.62	0.79
Cand. list size (l_{size})	6	-	6	9

TABLE I: Parameter values used in our experimental methodology. TB^* denotes the runtime required by the optimal approach.

- $\gamma = 0.26$
- $d_{\text{rate}} = 0.62$
- $l_{\text{size}} = 6$

These parameter values were determined by using the `irace` package [26], a state of the art tool for hyper-parameter optimisation. We ran `irace` on a set of 100 random instances different from the instances used for testing, in order to ensure a meaningful evaluation of our approach. We summarise all the parameters used in our experimental methodology in Table I. All data points report the average and the standard deviation over 20 random instances generated as discussed above.

We remark that our approximate offline approach benefits from parallelism both in the generation and in the solution of the ILP. On the one hand, as represented graphically in Figure 1, multiple runs of PG^2 can be safely executed in parallel, hence it is possible to run up to μ different threads (where μ is the maximum number of threads supported by the architecture running the algorithm), each executing one instance of the algorithm. On the other hand, modern ILP solvers automatically exploit the degree of parallelism available on the architecture in order to speed-up the computation of the ILP solution.

2) Offline quality wrt *optimality:* In our first set of experiments, we evaluate the quality of the solutions produced by our offline approach in very tightly time-constrained scenarios *wrt* the optimal solutions computed using the ILP formalisation considering all candidate cars (Equation 6). Since we are interested in measuring the quality *wrt* optimality, we consider small-case instances with a rate of 10 requests per minute, since computing the optimal solution is not feasible for larger problems. Then, for each problem instance we measure the runtime required by the optimal approach (namely, TB^*), we run our approximate offline approach with a total time budget (including both generation and solution) $TB \in \{5 \cdot 10^{-5} \cdot TB^*, 10^{-4} \cdot TB^*, \ldots, 10^{-1} \cdot TB^*\}$, and we measure the resulting solution quality.

Figure 5 shows that our approximate offline approach produces solutions with 95.4% quality in 3 orders of magnitude less time than the optimal approach, 97.2% quality in 2 orders of magnitude less time than the optimal approach, and 98.1% quality in 1 order of magnitude less time than the optimal approach, confirming that our approach produces solutions of very good quality while operating on a very tight time budget.

3) Online quality wrt *offline:* We now benchmark our online approach, both in small and large-scale P2P-RS scenarios. To this end, we measure the quality of solutions produced
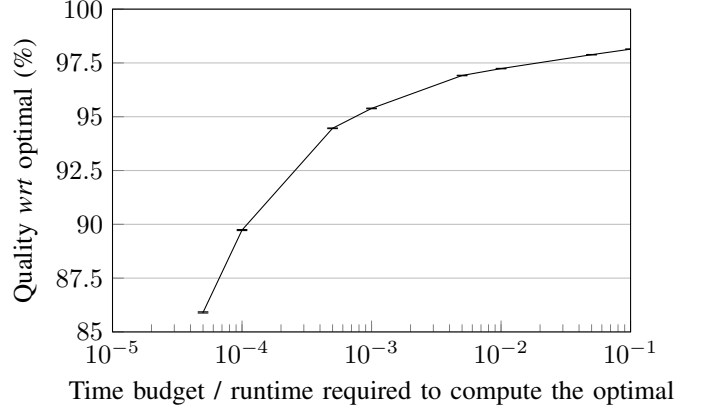

Fig. 5: Offline quality with different time budgets.

by our online approach *wrt* offline solutions computed with complete knowledge about the future (i.e., when the input stream of requests if fully known a priori). By doing so, we aim at evaluating our capability of making online decisions that accurately take into account the future. Specifically, for small-scale scenarios (with a rate of 10 requests per minute) we consider the optimal solution as a reference, which can be computed in a feasible amount of time. On the other hand, in large-scale scenarios (with a rate of 100 and 500 requests per minute) we compare against the solution produced by our offline approach (since the optimal solution cannot be computed) when given a time budget of 2 hours. Here we also consider the additional medium-scale scenario of 50 requests per minute. Since we consider a time budget significantly larger that the general case (2 hours *vs* 1 minute), the parameters of the offline approach have been re-tuned with `irace` specifically for this experiment, obtaining:

- $\gamma = 0.49$
- $d_{\text{rate}} = 0.79$
- $l_{\text{size}} = 9$

Here the γ parameter (i.e., the portion of time budget devoted to the solution of the generated ILP) is significantly larger than the previous one (i.e., $\gamma = 0.26$). This indicates that, when we increase the total time budget, it is not profitable to devote a large portion of it to the generation of the ILP model, which would be unnecessarily large (i.e., with many variables corresponding to poor candidate cars) for CPLEX to solve.

Figure 6 shows that our approach achieves a quality of at least 88.3% (with 10 requests per minute). Since this quality is
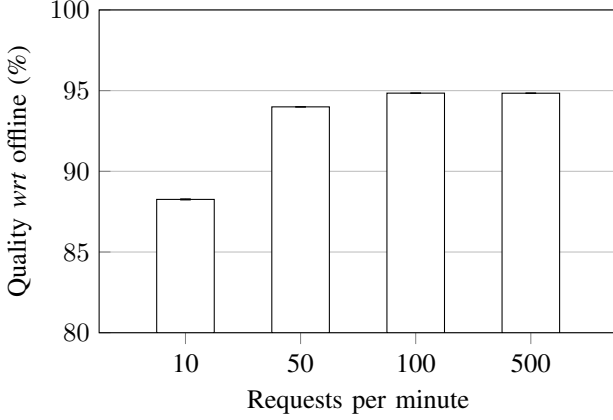
Fig. 6: Online quality *wrt* offline.

compared to the one achieved by the optimal offline approach with complete future information, this is the *competitive ratio* [27] of our approach.

Interestingly, the quality progressively increases as the problem size increases, ranging from 94.0% (50 requests per minute) to 94.8% (500 requests per minute). Although a bit surprising (as we are solving a larger, and, hence, more difficult problem), these results can be explained as follows. When the rate of requests (and hence, the number of requests in the pool at each time step) grows, the *relative* impact of each suboptimal decision taken during the online algorithm decreases, as the overall number of formed cars is larger.

Overall, these results show that our approach can successfully deal with large-scale P2P-RS problems, while producing solutions of very good quality.

V. DISCUSSION & CONCLUSIONS

In this paper we propose a novel approach that makes real-time *peer-to-peer ridesharing technologically feasible*. Indeed, our approach is capable of handling hundreds of trip requests being continuously issued by a large population of travellers (like NYC) to group them into shared rides in very short time (less than one minute).

Furthermore, for the first time we empirically measure the societal benefits of ridesharing when considering different ridesharing adoption rates. Although there have been studies trying to estimate the potential societal benefits of business-to-consumer ridesharing [28], and even of peer-to-peer carsharing [29], [30], the study in this paper is the first to focus on peer-to-peer ridesharing. *Our analysis on a real-world dataset shows that major societal benefits are expected from deploying peer-to-peer ridesharing* depending on the trade-off between environmental benefits and QoS. Thus, when environmental benefits are prioritised, our approach can yield a 70.78% reduction in CO_2 emissions (corresponding to up to 107.18 Tons per day) and a 80.08% reduction in traffic congestion, while achieving full occupancy on shared rides. On the other hand, when QoS is fully privileged (i.e., minimising delays experienced by the users), we can still achieve environmental benefits (7.85% reduction in CO_2 emissions, and 15.08% in the number of trips), showing that our approach can still lead to

benefits even without impacting on the QoS. Our experimental evaluation shows that New York City offers ample opportunities for trip sharing with minimal passenger discomfort, without the need of more computationally expensive strategies in which already started trips would be rerouted [9].

Notice that our approach also stands as valuable computational tool for policy makers concerned with shared mobility. Our approach provides the means to estimate the expected societal benefits of peer-to-peer ridesharing from existing travelling data records (as we do with the NYC taxi dataset in this paper). Thus, policy makers can leverage on our approach in order to come up with *action plans* aimed at achieving target societal impacts within some time frame, which is the common practice in policy making (e.g., refer to the *New York State Energy Plan* [31]). Moreover, our study may also help policy makers conduct *what-if analysis*. For instance, considering that both environmental benefits and QoS are equally important, achieving a reduction of 16.60 Tons of CO_2 (10.97% reduction) per day would require reaching a 20% ridesharing adoption rate, namely doubling the current ridesharing adoption figures in the US [7]. Far more ambitious, and less feasible, appears to achieve a reduction of 78.86 Tons of CO_2 (52.08% reduction) per day, since these benefits require an 80% adoption rate.

Our work opens up several interesting research lines, mainly regarding the design of incentive plans and pricing mechanisms (e.g., along the lines of [32]) that foster the adoption of ridesharing, increase the adoption rate, and the extension of our approach to incorporate the users' preferences as a further optimisation criterion (in addition to environmental benefits and QoS). Furthermore, we plan to extend our solution approach to other large-scale online stochastic combinatorial optimisation problems, e.g., online team formation [33] and on-demand crowd-sourcing [34], in virtue of its generality.

APPENDIX A
WORKING DAY RESULTS

Here we report the results of the experiments conducted on a working day (i.e., 1 December 2016) with the same methodology as discussed in the main paper. The considered working day contains 308187 requests, as opposed to the 343363 requests (+11.41%) reported during the considered weekend day (i.e., 3 December 2016).

The results reported in Figure 7 show that all the measures of *relative* nature (i.e., average occupancy, average in-car delay, average total delay, distance percentage reduction, and number of trips percentage reduction) show negligible differences *wrt* the weekend day, since, despite the greater number of requests during the weekend, the distribution of such requests is comparable, hence producing a comparable relative behaviour. On the other hand, CO_2 reduction, being an *absolute* measure depending on the total travelled distance, is significantly larger during the weekend, in virtue of the greater number of requests and, hence, the greater total travelled distance. Specifically, considering the main case of $\delta = 5$, during the weekend day the reduction of CO_2 emission was up to 16.73 Tons larger (9.06 Tons larger on average) than that achieved during a working day.
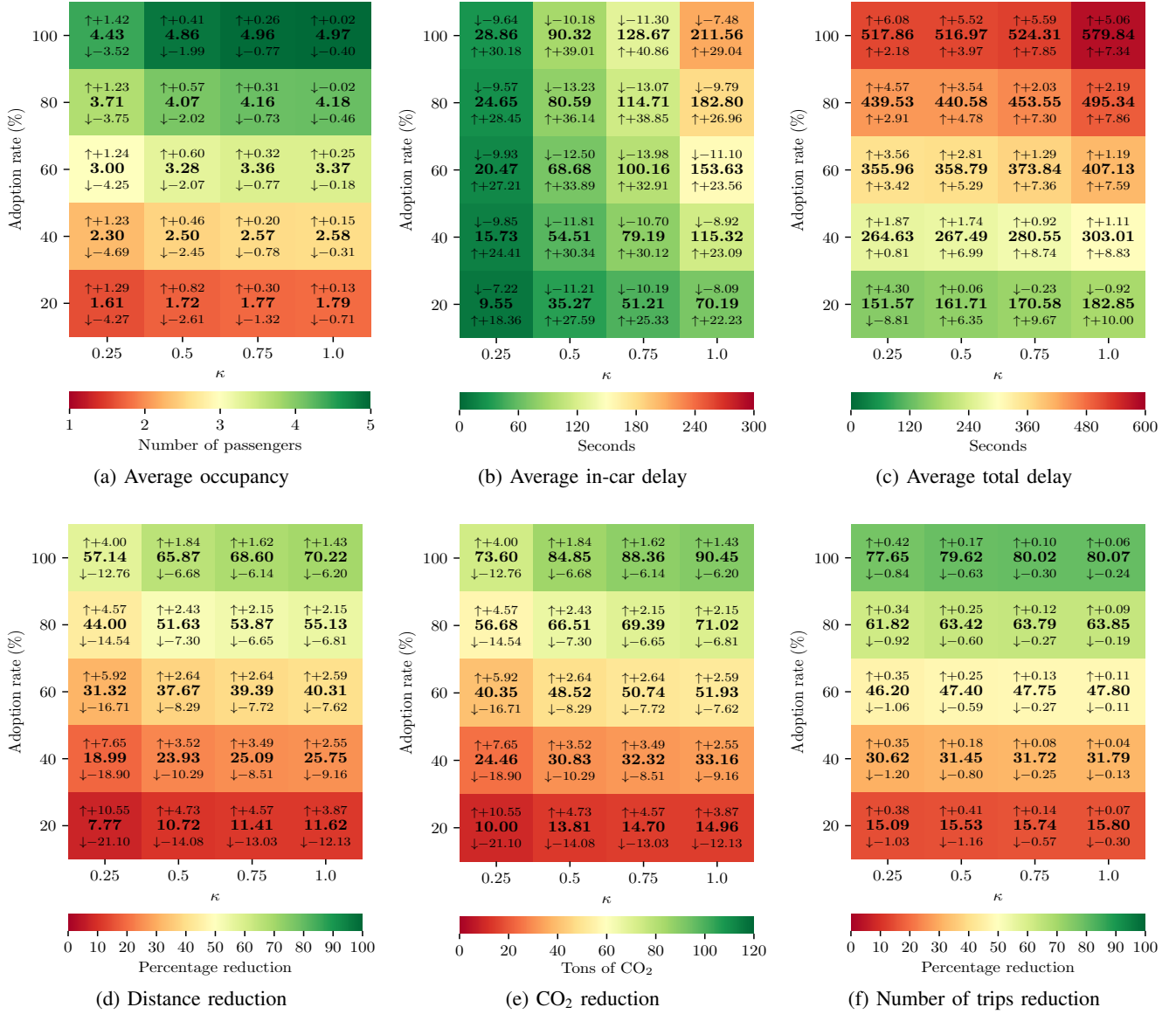
Fig. 7: [Best viewed in colour] Average occupancy (7a), average in-car delay (7b), average total delay (7c), distance reduction (7d), CO_2 reduction (7e), and number of trips reduction (7f) in the selected working day (01/12/2016), for different values for κ (on the x axis) and different adoption rates (on the y axis). Within each cell, the central **bold** value (which determines the colour of the cell) refers to the case with a maximum waiting time δ of 5 minutes for each request. The value above (resp. below) refers to the case with $\delta = 7$ minutes (resp. 2 minutes), showing the percentage difference *wrt* $\delta = 5$.

APPENDIX B
WEB-BASED SIMULATION AND ANALYSIS TOOL

We provide a simulation tool that demonstrates the approach discussed in this paper. Our tool is implemented by a web-based interface available at `https://ridesharing.iiia.csic.es`, and it offers two functionalities: *simulation* (Figure 8a) and *analysis* (Figure 8b). On the one hand, the user can select several *simulation* parameters, e.g., the adoption rate, the environmental benefit importance, the number of simulation ticks, the number of requests per tick, and the maximum waiting time for each request. Our interface then provides statistics for the selected scenario in terms of environmental benefits and QoS, and displays the result of the simulation over the map of the considered urban area (i.e.,

Manhattan in our case). Furthermore, the user can modify the value of the adoption rate and the κ parameter, and directly see the impact of such changes on the results of the simulation.

On the other hand, by means of the *analysis* functionality we offer the possibility of setting an objective in terms of environmental benefits and QoS, and to determine the necessary ridesharing adoption policy to achieve such an objective.
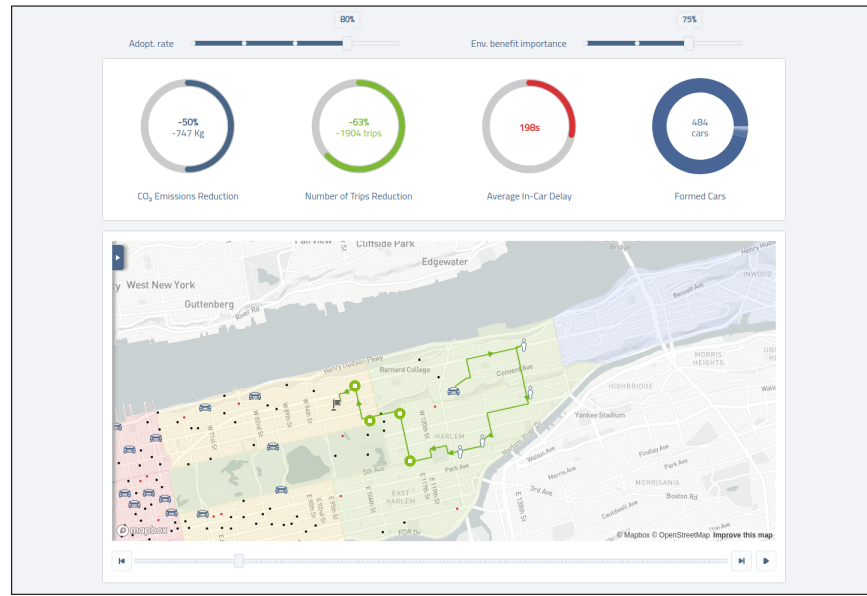
(a) *Simulation* interface (results refer to the default parameters).

(b) *Analysis* interface showing 3 example adoption scenarios.

Fig. 8: [Best viewed in colour] Screenshots of our web-based simulation tool: *simulation* and *analysis*.

REFERENCES

[1] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," *Proc. of the National Academy of Sciences*, vol. 114, no. 3, pp. 462–467, 2017.

[2] G. Fontaras, N.-G. Zacharof, and B. Ciuffo, "Fuel consumption and CO2 emissions from passenger cars in Europe–laboratory versus real-world emissions," *Progress in Energy and Combustion Science*, vol. 60, pp. 97–131, 2017.

[3] M. Ostrovsky and M. Schwarz, "Carpooling and the economics of self-driving cars," *National Bureau of Economic Research*, 2018, working paper 24349.

[4] D. J. Fagnant and K. M. Kockelman, "The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios," *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 1–13, 2014.

[5] ——, "Dynamic ride-sharing and optimal fleet sizing for a system of shared autonomous vehicles," in *Proc. of the Transportation Research Board Annual Meeting*, no. 15-1962, 2015.

[6] G. Santos, "Incentives to encourage shared mobility," Centre on Regulation in Europe, Tech. Rep., 2017.

[7] US Census Bureau. (2016) Commuting survey. Online at https://factfinder.census.gov/bkmk/table/1.0/en/ACS/16_5YR/S0801/0100000US.

[8] N. D. Chan and S. A. Shaheen, "Ridesharing in North America: Past, present, and future," *Transport Reviews*, vol. 32, no. 1, pp. 93–112, 2012.

[9] P. Santi, G. Resta, M. Szell, S. Sobolevsky, S. H. Strogatz, and C. Ratti, "Quantifying the benefits of vehicle pooling with shareability networks," *Proc. of the National Academy of Sciences*, vol. 111, no. 37, pp. 13 290–13 294, 2014.

[10] K. J. Dueker, B. O. Bair, and I. P. Levin, "Ride-sharing: psychological factors," *Transportation engineering journal of the American Society of Civil Engineers*, vol. 103, no. 6, pp. 685–692, 1977.

[11] R. F. Teal, "Carpooling: who, how and why," *Transportation Research*

Part A: General, vol. 21, no. 3, pp. 203–214, 1987.

[12] D. Pelzer, J. Xiao, D. Zehe, M. H. Lees, A. C. Knoll, and H. Aydt, "A partition-based match making algorithm for dynamic ridesharing," *IEEE Trans. on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2587–2598, 2015.

[13] M. Rigby, S. Winter, and A. Krüger, "A continuous representation of ad hoc ridesharing potential," *IEEE Trans. on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2832–2842, 2016.

[14] N. Bicocchi, M. Mamei, A. Sassi, and F. Zambonelli, "On recommending opportunistic rides," *IEEE Trans. on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3328–3338, 2017.

[15] C. Riley, A. Legrain, and P. Van Hentenryck, "Column generation for real-time ride-sharing operations," in *Proc. of CPAIOR 2019 – Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, ser. Lecture Notes in Computer Science, L.-M. Rousseau and K. Stergiou, Eds., vol. 11494, 2019, pp. 472–487.

[16] P. Delhomme and A. Gheorghiu, "Comparing French carpoolers and non-carpoolers: which factors contribute the most to carpooling?" *Transportation Research Part D: Transport and Environment*, vol. 42, pp. 1–15, 2016.

[17] NYC Taxi and Limousine Commission. (2017) Trip record data. Online at https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.

[18] P. Van Hentenryck and R. Bent, *Online stochastic combinatorial optimization*. The MIT Press, 2009.

[19] G. van Blokland and B. Peeters, "Modeling the noise emission of road vehicles and results of recent experiments," in *Innovations in Practical Noise Control, Inter-Noise*, 2009.

[20] F. Bistaffa, A. Farinelli, G. Chalkiadakis, and S. D. Ramchurn, "A cooperative game-theoretic approach to the social ridesharing problem," *Artificial Intelligence*, vol. 246, pp. 86–117, 2017.

[21] M. M. Halldórsson, "Approximations of weighted independent set and hereditary subset problems," *Journal of Graph Algorithms and Applications*, vol. 4, no. 1, pp. 1–16, 2000.

[22] D. Clements, J. Crawford, D. Joslin, G. Nemhauser, M. Puttlitz, and M. Savelsbergh, "Heuristic optimization: A hybrid AI/OR approach," in *Proc. of the Workshop on Industrial Constraint-Directed Scheduling*, 1997.

[23] W. Cook and P. Seymour, "Tour merging via branch-decomposition," *INFORMS Journal on Computing*, vol. 15, no. 3, pp. 233–248, 2003.

[24] G. W. Klau, I. Ljubić, A. Moser, P. Mutzel, P. Neuner, U. Pferschy, G. Raidl, and R. Weiskircher, "Combining a memetic algorithm with integer programming to solve the prize-collecting Steiner tree problem," in *Genetic and Evolutionary Computation Conference*, 2004, pp. 1304–1315.

[25] C. Blum, P. Pinacho, M. López-Ibáñez, and J. A. Lozano, "Construct, Merge, Solve & Adapt: A new general algorithm for combinatorial optimization," *Computers & Operations Research*, vol. 68, pp. 75–88, 2016.

[26] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, T. Stützle, and M. Birattari, "The irace package: Iterated racing for automatic algorithm configuration," *Operations Research Perspectives*, vol. 3, pp. 43–58, 2016.

[27] A. Borodin and R. El-Yaniv, *Online computation and competitive analysis*. Cambridge University Press, 2005.

[28] J. Viegas and L. Martinez, "Transition to shared mobility: How large cities can deliver inclusive transport services," Corporate Partnership Board Report, International Transport Forum, Tech. Rep., 2017, online at https://www.itf-oecd.org/transition-shared-mobility.

[29] T. D. Chen and K. M. Kockelman, "Carsharing's life-cycle impacts on energy use and greenhouse gas emissions," *Transportation Research Part D: Transport and Environment*, vol. 47, pp. 276–284, 2016.

[30] N. Rabbitt and B. Ghosh, "Economic and environmental impacts of organised car sharing services: A case study of Ireland," *Research in Transportation Economics*, vol. 57, pp. 3–12, 2016.

[31] New York State. (2015) Energy plan. Online at https://energyplan.ny.gov/-/media/nysenergyplan/2015-overview.pdf.

[32] J. Zhang, D. Wen, and S. Zeng, "A discounted trade reduction mechanism for dynamic ridesharing pricing," *IEEE Trans. on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1586–1595, 2016.

[33] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, "Online team formation in social networks," in *Proc. of the International Conference on World Wide Web*, 2012, pp. 839–848.

[34] A. Kittur, J. V. Nickerson, M. Bernstein, E. Gerber, A. Shaw, J. Zimmerman, M. Lease, and J. Horton, "The future of crowd work," in *Proc. of the Conference on Computer Supported Cooperative Work*, 2013, pp. 1301–1318.

Filippo Bistaffa received the Ph.D. in Computer Science from the University of Verona in 2016. He is currently a Marie Skłodowska-Curie Fellow at the Artificial Intelligence Research Institute (IIIA-CSIC), Bellaterra, Spain. His research interests comprise combinatorial optimisation problems for realistic applications (such as ridesharing and team formation) and GPU computing.


Christian Blum received the Ph.D. degree in Applied Sciences from the Free University of Brussels, Belgium, in 2004. He currently holds the position of a Senior Research Scientist at the Artificial Intelligence Research Institute (IIIA-CSIC), Bellaterra, Spain. His research interests include solving difficult optimisation problems using swarm intelligence techniques or using combinations of metaheuristics and exact techniques.

Jesús Cerquides is a tenured scientist at the Artificial Intelligence Research Institute (IIIA-CSIC), Bellaterra, Spain. He is actively conducting research in the areas of probabilistic graphical models and multi-agent systems. Formerly he was associate professor in the University of Barcelona, where he founded the Volume Visualisation and Artificial Intelligence Research Group. Prior to that he was head of iSOCOLab, Chief Technology Officer, co-founder and member of the board at Intelligent Software Components and Associate Director at UBS IT Research Lab. Prof. Cerquides has lead several industrial research projects and participated in many academic research projects in the areas of machine learning and multi-agent systems.

Alessandro Farinelli received the Ph.D. degree from the University of Rome La Sapienza in 2005. He is an associate professor at University of Verona, Department of Computer Science. His research interests focus on developing novel Artificial Intelligence methodologies applied to Robotics and Cyber-Physical Systems. In particular, he focuses on multi-agent coordination, decentralised optimisation, reinforcement learning and data analysis for Cyber-Physical Systems.
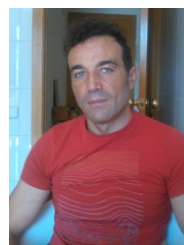
Juan A. Rodriguez-Aguilar received a Ph.D. degree in Computer Science from the Autonomous University of Barcelona in 2001. He is a research scientist at the Artificial Intelligence Research Institute (IIIA-CSIC), Bellaterra, Spain. His research interests encompass multiagent optimisation, artificial intelligence and ethics, industrial applications of artificial intelligence, and artificial social systems. He is a fellow of the European Association for Artificial Intelligence.