Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 23 (2017) 1000–1024

Transportation Research Procedia

www.elsevier.com/locate/procedia

22nd International Symposium on Traffic and Transportation Theory

Privacy-preserving MaaS fleet management

Francois Belletti [a*], Alexandre M. Bayen [a,b,c]

[a] Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, United States
[b] Department of Civil and Environmental Engineering, University of California, Berkeley, United States
[c] Institute of Transportation Studies University of California, Berkeley, United States

Abstract

On-demand traffic fleet optimization requires operating *Mobility as a Service* (MaaS) companies such as Uber, Lyft to locally match the offer of available vehicles with their expected number of requests referred to as demand (as well as to take into account other constraints such as driver's schedules and preferences). In the present article, we show that this problem can be encoded into a *Constrained Integer Quadratic Program* (CIQP) with block independent constraints that can then be relaxed in the form of a convex optimization program.We leverage this particular structure to yield a scalable distributed optimization algorithm corresponding to computing a gradient ascent in a dual space. This new framework does not require the drivers to share their availabilities with the operating company (as opposed to standard practice in today's mobility as a service companies). The resulting parallel algorithm can run on a distributed smartphone based platform.

Peer-review under responsibility of the scientific committee of the 22nd International Symposium on Transportation and Traffic Theory.

Keywords: Fleet Management, Privacy-preservation, Optimization

Open-sourcing *Mobility as a Service*. A key component in *Mobility as a Service* (MaaS) companies such as Uber and Lyft is market making and market matching. The famous price multiplier of the former incentivizes drivers to go to regions where many rides are requested whereas the latter takes into account the personal schedule of the drivers. To remain competitive, MaaS companies keep their optimization strategies a secret. In the present article, we present a convex optimization formulation and provide an open source algorithm that match drivers with demand in an optimal manner while taking into account the drivers' availability as a set of constraints. In addition, we show that using a dual splitting technique, the convex program can be parallelized with respect to the agents. This has two important consequences. It alleviates the need for a single machine with a large amount of memory. Indeed we leverage the fact that the schedules of the drivers can be considered as independent constraint sets. It also does not require drivers to share their schedules which naturally protects this privacy sensitive piece of information.

Background on fleet management. A sizable body of research focuses on optimizing fleet management, vehicle dispatching and multimodal transportation optimization. In order to assess the best optimization policies for taxi fleets, numerous simulators have been developed. For instance, in Cheng and Nguyen (2011), an agent-based simulation program was used in conjunction with a large dataset of GPS trajectories to learn and mimic the behavior of taxi drivers. In Maciejewski and Nagel (2013) researchers have modified gen-eral purpose agent-based traffic simulators such as MATSim Horni et al. (2016) and incorporated taxi ride simulation. Both these approaches aimed at providing taxi companies with tools and methods dedicated to better informing fleet management decisions with a focus on better answering demand modeled as queues and minimizing idling driver time. GPS tracking has been instrumental in calibrating taxi fleet simulators as well as real-time information services for large taxi fleets Balan et al. (2011); Liao (2001, 2003) which aimed at better informing optimization decisions in real time. More recently, the advent of massive electric vehicle

*Corresponding author:
Email Address: francois.belletti@berkeley.edu

2352-1465 © 2017 The Authors. Elsevier B.V. All rights reserved.
Peer-review under responsibility of the scientific committee of the 22nd International Symposium on Transportation and Traffic Theory.
10.1016/j.trpro.2017.05.055

fleets has given rise to a new class of problems. These analyze the structure of demand of rides and dispatch vehicles while taking into account the constraints inherent to the use of electric vehicles whose batteries provide a shorter range Gacias and Meunier (2015). Such problems have also been considered for the design of optimal charging station networks for such fleets Cai et al. (2014).
Integrating operational constraints in privacy preserving environments. The framework in which we develop the novel approaches presented in this work takes into account constraints that correspond to each agent in the system such as driving preferences, electromobility-induced constraints, etc. These constraints can encode a variety of factors, the most interesting one for the study being driver availability (which we want to protect for privacy reasons). In such a setting, the demand for rides can be studied in a statistical manner leveraging the important amount of data being collected by MaaS systems. Note also, as will appear later, that the present article does not require the assumption that the demand for rides is elastic as in Wong et al. (2001). Our focus is that of making MaaS open; this specifically takes the form of a peer-to-peer distributed algorithm that can run on smartphones and achieves an aim similar to that of Jayakrishnan (2015). As in Ma et al. (2013), a major concern is that the algorithm scales which typically becomes problematic when the number of constraints increases linearly with the number of drivers, a problem that we adress as well. For completeness, a large number of approaches for ride-sharing optimization can be found in Agatz et al. (2012).

Contributions of the article. Based on this study of the state of the art in optimization for MaaS services, our approach is novel in three key aspects which we delineate below.

- **Modeling and data analytics.** We show how openly accessible data can, using a coarse yet robust statistical estimation, yield reliable estimates of the locations and times corresponding to demand for mobility. We operate under the assumption that the only shared location by the rider is that of the departure of the trip. This corresponds to the reality of popular applications for MaaS today in which the destination of the rider can be modified until the start of the trip. We show how, once the demand for rides has been characterized using this method, the availability constraints of the drivers can be matched as a mobility offer in the form of an integer program. Assuming vehicles would directly be controlled by the managing company, this leads to a Contrained Quadratic Integer Program (CQIP). As the market making solution being offered by MaaS does not directly control the drivers, we focus on controlling a probability of presence of the drivers which corresponds to a convex relaxation of the integer program we formulate.
- **Privacy preservation by dual splitting.** A salient issue with the convex optimization program that needs to be solved is that its number of constraints scales linearly with the number of drivers. This makes the complexity of computations cubic in time and memory with respect to the number of vehicles for standard convex optimization solvers that use interior point methods or any solution based on inverting the matrix of linear constraints that characterizes the problem. We develop a dual splitting technique which splits the problem into as many subproblems as there are drivers involved in the service. We show how this results in a gradient descent based convergence method which requires each driver to solve a subproblem at each descent step for the gradient to be collectively computed. We prove that the convergence of the scheme is linear and the computational burden for each device is only proportional to the number of constraints of the driver it corresponds to as opposed to that of the whole fleet of vehicles. A key element is that the only information that needs sharing is a Lagrangian price multiplier which preserves the privacy of the drivers and mostly gives them an incentive to go where most rides are needed. Their personal availability constraints need not be shared in our system. We further show how their recommended trajectory can be obfuscated thereby further protecting their privacy. The algorithmic contribution relies on techniques different from the famous *Alternating Direction Method of Multipliers* (ADMM) of Boyd et al. (2011), distributed primal methods Nedic and Ozdaglar (2009); Goldfarb and Ma (2012) or primal-dual methods Chang et al. (2014). Indeed it is tailored to the block independent structure of the constraints and the partitioning of the data across smartphones. This section introduces a method that is privacy preserving and tractable computationally.
- **Theoretical convergence analysis and numerical experiments.** A thorough analysis of the convergence rates of the algorithm we present to solve the convex program is conducted. This analysis is general and takes place for a larger class of problems than the program we designed thereby making it useful for practitioners with different objective functions (i.e. not limited to the operational scenarios

outlined in the earlier parts of the article). We show that the scheme is robust to noise being injected in communications. After delineating two privacy preserving scalable algorithms, we prove that as the number of drivers increases, privacy can further be improved. We highlight a crowd obfuscation effect thanks to which an individual attacker cannot learn as fast as the group. We confirm the results obtained in this holistic theoretical analysis of the properties of the algorithm with numerical experiments conducted with actual openly accessible data in both the noiseless and noisy settings. The open source code will be made available through a hypertext link in the final public version of the present article. Section 3 proves that the algorithm we present is robust enough to allow for the obfuscation of the optimal program it converges to.

Organization of the article We first anchor a theoretical optimization program into the reality of mobility as a service applications, highlight how a convex relaxation corresponds to a realistic assumption which later on leads to a novel dual splitting technique and finally analyze the theoretical and numerical properties of the resulting distributed algorithm. The present article is therefore organized as follows. In Section 1, we first briefly describe the method used to characterize the spatial and temporal structure of demand for mobility as a service rides in New York. While this is not the core focus of the present article, it is briefly presented in Section 1 since the rest of the article leverages this spatio-temporal structure once it has been characterized. We then formulate a convex optimization problem in Section 2 that matches the probability of presence of drivers with the demand for rides. Finally, in Section 3, a theoretical analysis of the convergence properties of the distributed algorithm is conducted that proves its privacy preserving properties. Numerical experiments confirming these results are shown in Section 4.

1. General modeling of the fleet management problem

In this section, we show how to model the problem of matching demand for rides with the schedule of MaaS drivers and how this can be formulated as an optimization program. We first focus on the identification of the structure of the demand prior to formulating the resulting matching problem.

1.1. Characterization of demand as a seasonal time series

Setting. Demand for rides can be characterized as a multivariate time series $(D_t) \in \mathbb{R}^d$ where d is the number of cells considered in the spatial discretization grid (see Figure 1 for the case of NYC). These are sometimes referred to as "heatmaps" in MaaS companies. For each element of the grid i and each timestep t (typically 5 minutes), D_t^i is the sum of all ride requests for the service that have been observed in region i. Let $(U_t) \in \mathbb{R}^d$ the sum of vehicles available in a grid cell at time t. A way to encode optimal matching is to penalize the sum of squared distances between the vectors U_t and D_t at each timestamp to which we add a regularizing function over (U_t). This latter component of the objective represents the aversion of drivers to long distances and going to highly congested zones. This will enforce a vehicle allocation schedule matching the spatial density of ride requests with that of available mobility platforms. At the same time, it will penalize the fleet dispatch plan in order to discourage traveling long distances or go through regions with a high number of vehicles, therefore taking into the route preferences of the drivers Bogers et al. (2015).

Spatio-temporal structure of demand for MaaS. As an instantiation of the method exposed to identify the structure of demand for rides, further exploring the statistical facts exposed in Qian et al. (2013) for on-demand mobility in New York city, we show that the demand for rides (D_t) has strong spatial and temporal periodicity (which is illustrated in Figure 2 and Figure 3). A consequence of this fact is that we can use seasonality analysis Brillinger (1981); Brockwell and Davis (2013) as the basis for an optimization scheme taking into account the drivers' availability and schedules. The exact definition of privacy preservation will be given later in the article. The stochastic input demand profile Ak and Erera (2007) can of course be updated as new data is collected with a receding horizon approach Miao et al. (2015) and the demand forecast obtained by multivariate time series analysis Sun et al. (2003); Durango-Cohen (2007); Lütkepohl (2005) changes. Similar updates for traffic conditions can be taken into account in real time Ben-Akiva et al. (2001); May et al. (2003).

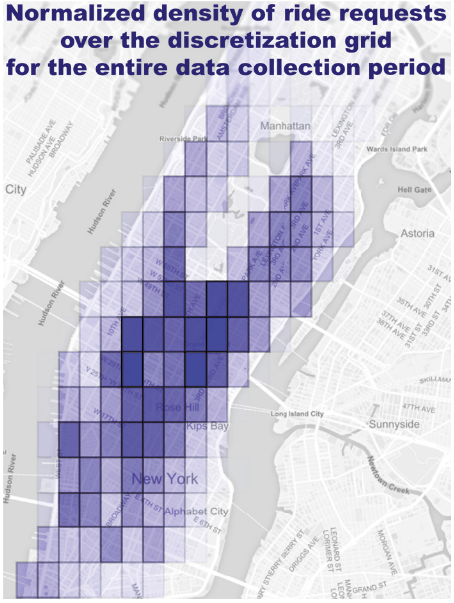

Fig. 1. Spatial quantization of demand for Uber rides in New York. Each element of the grid corresponds to aggregate ride requests in a tessellation region during each time interval (depicted at a given time in this figure). Empty cells correspond to regions with low or zero ride requests. This data is studied as a multivariate time series and its seasonal component studied. The mobility on-demand fleet distribution is then optimized based on this data with the novel algorithm we present.

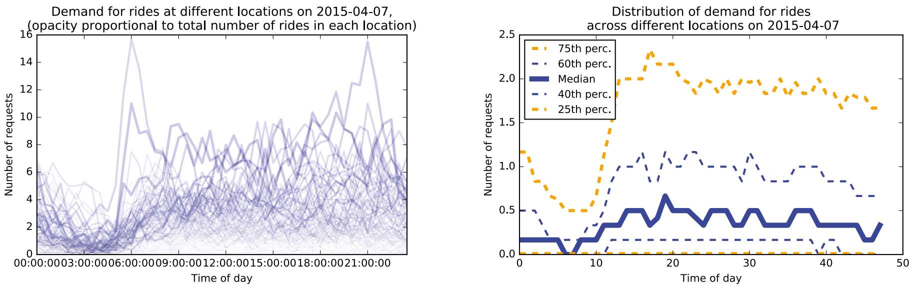

Fig. 2. Left: Demand series for specific locations of NYC on a particular day. Intra-day seasonality appears as a salient feature of the demand of MaaS. For each element of the discretization grid, ride requests are aggregated over 5 minute observation windows and represented with lines whose opacity is proportional to the total number of ride requests observed at the corresponding element of the grid for the entire day. Right: Average and empirical percentiles computed across all the discretization cells used to represent the state of demand for rides across the city.

1.2. Formulation of demand matching with boolean representation of vehicles

The time series $(U_t) \in \mathbb{R}^d$ corresponds to the aggregation of drivers' locations over spatial regions whereas $(D_t) \in \mathbb{R}^d$ is a demand time series summing up collocated requests occurring in the same time interval. The optimization scheme we focus on in this article leverages aggregated demand profiles and enables the optimization of vehicle dispatch at the driver level, taking personal constraints and driver utility functions into account.

The discretized spatial grid can easily be modeled as a network graph Wu et al. (2004) representing the mobility space (see Figure 1). Each edge represents a spatial tesselation region. Neighboring regions are linked one to another by an edge. The position of driver c at time t is modeled by a integer vector over the vertices of $\mathcal{G}$: $(b_t^c) = \left(b_{t,i}^c\right)_{i\in\{1\ldots d\}} \in \mathbb{N}^d$ whose value is 1 for node n if and only if the driver's location is node n at time t. The value of the integer b_t^c may also be used to encode the number of seats available in a given vehicle at a given time and space.

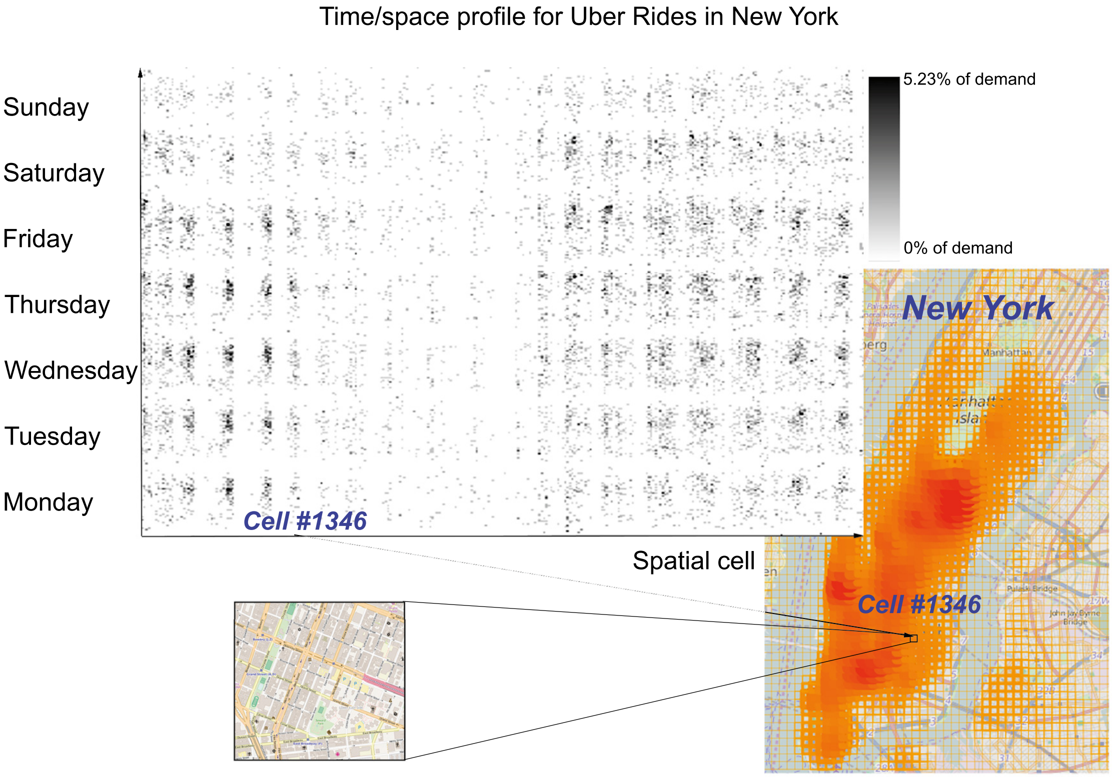

Fig. 3. Spatio-temporal seasonality in Uber demand profile in NYC for April 2014. The strong spatio-temporal seasonality paves the way towards accurate predictions and planning of requests for rides. Forecast can then be dynamically updated by standard linear model iterative fitting to dynamically adapt to randomness in the demand time series.

1.2.1. *Formulation of demand matching as a discrete optimization program*

Up to a scaling factor for the number of shared vehicles located at a given node, alleviating the spread between demand and offer of ride requests can be written as the following objective where C is the total number of drivers taken into account and T the total number of time steps the schedule is planned for.

Fleet management problem:

$$\min_{u\in\{0,1\}^{d\times C\times T}:} \sum_{t=1}^{T}\sum_{n=1}^{d}\left(d_{t,n}-\sum_{c=1}^{C} b_{t,n}^{c}\right)^2+\sum_{c=1}^{C}\sum_{t=1}^{T-1}\sum_{n=1}^{d}\rho_{c,t,n}^{2}\left(b_{t+1,n}^{c}-b_{t,n}^{c}\right)^2+\sum_{c=1}^{C}\sum_{t=1}^{T}\sum_{n=1}^{d}\sigma_{t,n}^{2}\left(b_{t,n}^{c}\right)^2$$
$$\forall c\in\{1\dots C\},\forall n\in\{1\dots D\},\forall t\in\{1\dots T\},b_{t,n}^{c}\geq 0 \text{ if and only if vehicle c in cell } n$$
$$\forall c\in\{1\dots C\},b^{c}\in C^{c} \quad (1)$$

where

- $\forall c\in\{1\dots C\},b^{c}\in C^{c}$ is equivalent to $\forall n\in\{1\dots D\},\forall t\in\{1\dots T\},b_{t,n}^{c}=0$ if vehicle c is not available at time t.
- $d_{t,n}$ is the demand in cell n at time t. This constraint set will be augmented with mobility constraints as we will show later on.
- $\left(\rho_{c,t,n}^{2}\right)$ is a set of travel distance penalization parameters.
- $\left(\sigma_{t,n}^{2}\right)$ is a set of penalization parameters over congested cells.

In the equations above and the following, t will be a temporal index, n a spatial index, and c will identify vehicles. The specific terms in the program above are explained below:

- Demand matching term: $\sum_{t=1}^{T}\sum_{n=1}^{d}\left(d_{t,n}-\sum_{c=1}^{C}b_{t,n}^{c}\right)^2$ is a square loss penalizing the mismatching between the number of vehicles expected in each cell and the number of rides requested (demand). Note that last section of this article proves that any strongly convex loss can be used here instead of this quadratic.
- Travel time and distance penalty: $\sum_{c=1}^{C}\sum_{t=1}^{T-1}\sum_{n=1}^{d}\rho_{c,t,n}^{2}\left(b_{t+1,n}^{c}-b_{t,i}^{c}\right)^2$ is a quadratic penalty term which accounts for the cost in time and energy for the drivers corresponding to moving across the discretized tesselation grid. In other words, penalizing fleet re-balancing is desirable (and encoded in this function). Again, any arbitrary strongly convex loss can be used here, as proved later in the article. This means in particular that instead of a distance an expected travel time can be added, for instance, a quadratic loss on the number of traffic lights present in a cell of the discretized grid.
- Regularization: $\sum_{c=1}^{C}\sum_{t=1}^{T}\sum_{n=1}^{d}\sigma_{t,n}^{2}\left(b_{t,n}^{c}\right)^2$ is a quadratic term that discourages drivers from gathering in large groups in the same cells. Once more, any strongly convex loss can replace the quadratic term we chose. Strong convexity implies that there is an increasing marginal cost to concentrating the distribution of a driver and therefore encourages spreading vehicles across the discretization grid. This term may also represent the aversion of drivers to driving in zones that feature problematic driving conditions such as uncertainty about the level of congestion or the absence of stopping locations.

1.2.2. Constraint sets

In this paragraph, we show how the constraints on the possible movements of the agents can be extended and be more detailed. Any fleet management algorithm has to take the drivers' constraints into account as well as their mobility constraints. Letting $avai\,(c,t)$ be the Boolean variable that represents the availability of driver c at time t, we have $\forall t\in\{1\ldots T\}$, $\forall c\in\{1\ldots C\}$, $\sum_{n=1}^{T}b_{t,n}^{c}=avai\,(c,t)$. Let $init\,(c)$ the entry point of driver c at time t_c^s (where s stands for "start time", the entry point represents the location at which the agent has parked before starting the MaaS part of their day) when vehicle c start its trip, we add the constraint $b_{t=t_c^s}^{c}=init\,(c)$, where $init\,(c)$ is the dirac vector concentrated on the start position of vehicle c. Let $end\,(c)$ the exit point of driver c at time t_c^e when vehicle c ends its trip, $b_{t=t_c^e}^{c}=end\,(c)$ the dirac vector concentrated on the stop position of vehicle c. If we do not assume that the graph is fully connected for any time stamp t, we can add the constraints $\forall t\in\{1\ldots T-1\}$, $b_t^c=M_t b_{t-1}^c$ where M_t is the connectivity matrix of the network. In particular M_t is filled with 0, $M_{t,n_1,n_2}=1$ if anf only if n_1 and n_2 are connected through the network at time t. The final constraint set we obtain, once mobility constraints have been taken into account, can be described as follows for a given agent with index c:

- Availability of agent for MaaS: $\forall c\in\{1\ldots C\},b^c\in C^c$ is equivalent to $\forall n\in\{1\ldots D\},\forall t\in\{1\ldots T\},b_{t,n}^c=0$ if vehicle c is not available at time t.
- Mobility constraints: $\forall t\in\{1\ldots T-1\}$, $b_t^c=M_t b_{t-1}^c$.

1.2.3. Hyperparameter tuning

The problem formulation above features several sets of hyperparameters (ρ, σ) which can be calibrated by a cross-validation procedure in order to produce the best outcome for this optimization scheme. This can be done by operating an actual fleet or more realistically in practice using a vehicle fleet simulator such as Cheng and Nguyen (2011); Maciejewski and Nagel (2013). In our numerical experiments, we have used nominal parameters that did not result from the use of such simulators different numerical values do not affect the performance of our algorithms significantly. However, the results we present show that the set of hyperparameters we selected successfully enabled the tracking of the demand with the vehicle probability of presence distribution.

We consider a convex relaxation of the *Constrained Integer Quadratic Programming* (CIQP) problem which differentiates our approach from Mahmassani et al. (2000); Lee et al. (2012); Robust et al. (1990); Seow et al. (2010) for vehicles. Our choice of an approximate relaxation convex program as in Miao et al. is motivated by the practical fact that we build a system that encourages drivers to move towards regions with a high number of potential customers. Drivers are indeed free of their re-routing decisions in on-demand mobility, which leads to behavioral changes as compared to what has been previously observed for vehicles Yang et al. (2010). As actors on a two sided market, they are also free to refuse to pickup a costumer and therefore the vehicle occupancy is not taken into account in the optimization objective. We derive a dual reformulation which yields a distributed gradient ascent algorithm converging to the optimal solution of a strongly convex problem.

Our method is therefore tailored to shaping incentives drivers to answer demand in an optimal way while taking their personal constraints into account as they join a massive fleet of vehicles. We leverage this unique aspect of MaaS in order to design a distributed algorithm which is tailored for speed of adaptation to changing conditions and privacy preservation rather than precisely controlling each vehicle individually. Our numerical experiments show indeed that the parallel implementation of our algorithm can run on the equivalent of a group of two thousand smartphones with a limited communication bandwidth and yet find an optimal price multiplier for the drivers in a matter of minutes. These results are moreover rather pessimistic as we did not use a warm-start or a specialized convex optimizer to speed it up. A limitation however is that our algorithm is not adapted to the exact and precise control of a fleet of vehicles.

2. Formulation and architecture for distributed privacy preserving motion planning

The computational complexity of the problem prevents efficient solutions to the exact (integer) problem. In addition, solving the problem exactly is not necessary with fleets of drivers that are not directly controlled. This enables us to achieve three main goals which will be at the center of the next two sections.

Specification	Brute force approach	Dual splitting	Crowd obfuscation
Tractable and scalable computations	Not achieved	Achieved	Achieved
Preservation of private constraints	Not achieved	Achieved	Achieved
Preservation of private optimal actions	Not achieved	Not achieved	Achieved

Table 1. Specifications we aim to meet for the fleet management problem.

2.1. Convex relaxation

As a starter to address the issue of tractability in Table 1 above, we employ relaxation. In the previous problem, $\forall t \in \{1 \dots T\}$, $\forall c \in \{1 \dots C\}$, $b^c_{t,n} \in \{0, 1\}$. We now write a relaxed approximation of the problem with $u^c_{t,n} \in [0, 1]$. $\forall c \in \{1 \dots C\}$ let C^c the set of constraints of vehicle c, the constraints are: $\forall t \in \{1 \dots T\}$, $\forall n \in \{1 \dots N\}$, $u^c_{t,n} \geq 0$;$\forall t \in \{1 \dots T\}$, $\forall n \in \{1 \dots N\}$, $u^c_{t,n} = 0$ if vehicle c is not available at time t; $u^c_{t=0} = init\,(c)$ (initial location of vehicle c, e.g. parking location at the driver's home); $u^c_{t=1} = end\,(c)$ (final location of vehicle c, e.g. parking location at the driver's home), $\forall t \in \{1 \dots T - 1\}, u^c_t = M_t u^c_{t-1}$ (movement constraints between traffic regions) if the traffic network is not fully connected at all times. Otherwise one considers the mass conservation constraint $\sum_{n=1}^{N} u^c_{t,n} = 1$. We can see that this problem is equivalent to that of considering the position of a given vehicle at any time t as a probability distribution over the network. Note also that the practice of aggregation of supply often used by MaaS companies such as Uber or Lyft is somewhat aligned with the physical interpretation of relaxation in the present case.

2.1.1. Relaxed problem formulation

As explained earlier, the position of driver c at time t is modeled by a presence vector over the vertices of $\mathcal{G}$: $(u^c_t) = \left(u^c_{t,n}\right)_{n\in\{1\dots N\}} \in \mathbb{R}^N$ whose value is one for node n if and only if the driver's location is node n at time t.

The relaxed problem we consider is now convex:
Convex relaxation:

$$\min_{u\in\mathbb{R}^{d\times C\times T}:\,\forall c\in\{1\dots C\},\,u^c\in C^c} \sum_{t=1}^{T}\sum_{n=1}^{N}\left(d_{t,n} - \sum_{c=1}^{C} u^c_{t,n}\right)^2 + \sum_{c=1}^{C}\sum_{t=1}^{T-1}\sum_{n=1}^{N}\rho_{t,c,n}\left(u^c_{t+1,n} - u^c_{t,n}\right)^2 + \sum_{c=1}^{C}\sum_{t=1}^{T}\sum_{n=1}^{N}\sigma_{t,n}\left(u^c_{t,n}\right)^2 \quad (2)$$

The solution to this problem will be denoted $u^* = concat\left((u^*_c)_{c=1\dots C}\right)$ where u^*_c is the optimal behavior assigned to the agent number c.

One major difference appears with the initial boolean problem. New constraints need to be taken into account, for each vehicle. Namely, $u^c_{t,n} \geq 0$ and $\sum_{n=1}^{N} u^c_{t,n} = 1$. This does not change the separability of constraints among vehicles which is the key assumption needed to parallelize solving the convex optimization problem we obtain after relaxation.

2.1.2. Answering demand for rides with a probability distribution of vehicles

We are mainly interested in fleet management for MaaS companies such as Lyft or Uber. In that particular setting it makes sense to consider a probability distribution of presence for the driver and guiding this probability distribution. Not only is this framework useful for large numbers of drivers as observed in urban areas, but it also corresponds to a modeling approach desired by MaaS companies. Also, the present optimization scheme only considers demand is characterized by the location and number of ride requests. The location of the destinations are not taken into account. This models the fact that Uber or Lyft drivers only get to know the destination of the costumer after the pick up and the rider is therefore free to change the destination before pickup time. Considering destination locations is not problematic. The corresponding increase in computational complexity can potentially be addressed with the optimization algorithm presented later based on a dual splitting method is tailored for solving the main problem by distributing it over many machines. As illustrated in Figure 4, considering the convex relaxation of the problem is key to having a model that although approximate is appropriate for Mobility as Service (MaaS) applications and can be solved in a scalable manner. A thorough analysis has been conducted in Pilanci et al. (2015) and Pilanci et al. (2012) for the recovery of sparse probability measures by convex programming. In these articles, bounds have been proven that could be applied to our setting. We do not need such a thorough analysis in our study as the convex relaxation over the location of the agent characterizes the flexibility offered to the driver by MaaS. Uber drivers are free to go where they want and can only receive an incentive to drive to zones with a higher price multiplier.

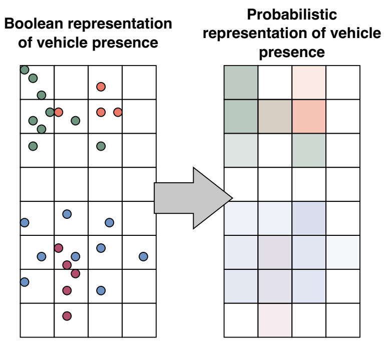

Fig. 4. Core idea behind the convex relaxation. As there is no direct control of the drivers by MaaS companies, we consider we control a distribution of probability of presence as is done by MaaS companies through the internal use of heatmaps. This corresponds to the intuition that the objective to achieve is finding an optimal incentivization scheme for the drivers. It is key to turn the exact problem into an approximation that can scale with respect to the number of drivers.

2.2. Dual splitting method

This section presents a dual perspective of the problem that explicitly separates the constraint sets of the drivers. It leads to the formulation of a Lagrangian quantity encoding both the structure of the demand and the availability constraints of the drivers.

Proposition 1. *The primal problem is equivalent to the* ***Dual problem:***

$$\boxed{\max_{\lambda} - \sum_{t=1}^{T} \sum_{n=1}^{N} \lambda_{t,n} d_{t,n} + \sum_{c=1}^{C} \min_{u^c \in C^c} \left(\sum_{t=1}^{T-1} \sum_{n=1}^{N} \rho_{t,c,n} \left(u^c_{t+1,n} - u^c_{t,n} \right)^2 + \sum_{t=1}^{T} \sum_{n=1}^{N} \sigma_{t,n} \left(u^c_{t,n} \right)^2 \right).} \tag{3}$$

Proof 1. *Let $z_{t,n} = d_{t,n} - \sum_{c=1}^{C} u_{t,n}^c$ be a set of consensus variables Boyd and Vandenberghe (2004). The initial problem can be rewritten as*

$$\min_{\substack{u\,:\ \forall c \in \{1\dots C\},\ u^c \in C^c \\ \forall t,n\ z_{t,n} = d_{t,n} - \sum_{c=1}^{C} u_{t,n}^c}} \sum_{t=1}^{T}\sum_{n=1}^{N} z_{t,n}^2 + \sum_{c=1}^{C}\sum_{t=1}^{T-1}\sum_{n=1}^{N}\rho_{t,c,n}\left(u_{t+1,n}^c - u_{t,n}^c\right)^2 + \sum_{c=1}^{C}\sum_{t=1}^{T}\sum_{n=1}^{N}\sigma_{t,n}\left(u_{t,n}^c\right)^2 \tag{4}$$

We create the set $(\lambda_{t,n})_{t\in\{1\dots T\},\, n\in\{1\dots N\}}$ of Lagrangian variables corresponding to the constraints $\forall t \in \{1\dots T\}, n \in \{1\dots N\}$ $z_{t,n} = d_{t,n} - \sum_{c=1}^{C} u_{t,n}^c$.

$$\min_{\substack{u\,:\ \forall c \in \{1\dots C\},\ u^c \in C^c \\ z \in \mathbb{R}^{N\times T}}} \max_{\lambda\in\mathbb{R}^{N\times T}} \begin{array}{c} \sum_{c=1}^{C}\sum_{t=1}^{T-1}\sum_{n=1}^{N}\rho_{t,c,n}\left(u_{t+1,n}^c - u_{t,n}^c\right)^2 + \sum_{c=1}^{C}\sum_{t=1}^{T}\sum_{n=1}^{N}\sigma_{t,n}\left(u_{t,n}^c\right)^2 \\ + \sum_{t=1}^{T}\sum_{n=1}^{N} z_{t,n}^2 + \lambda_{t,n}\left(z_{t,n} - d_{t,n} + \sum_{c=1}^{C} u_{t,n}^c\right) \end{array} \tag{5}$$

This problem can be rewritten as

$$\max_{\lambda\in\mathbb{R}^{N\times T}} \min_{u\,:\ \forall c \in \{1\dots C\},\ u^c \in C^c} \begin{array}{c} \sum_{c=1}^{C}\sum_{t=1}^{T-1}\sum_{n=1}^{N}\rho_{t,c,n}\left(u_{t+1,n}^c - u_{t,n}^c\right)^2 + \sum_{c=1}^{C}\sum_{t=1}^{T}\sum_{n=1}^{N}\sigma_{t,n}\left(u_{t,n}^c\right)^2 \\ + \sum_{t=1}^{T}\sum_{n=1}^{N} \min_{z_{t,n}\in\mathbb{R}} z_{t,n}^2 + \lambda_{t,n}\left(z_{t,n} - d_{t,n} + \sum_{c=1}^{C} u_{t,n}^c\right) \end{array}$$

where each individual minimization problem with respect to $z_{t,n}$ can be solved analytically with respect to $z_{t,n}$ (minimization of a second order polynomial). The swapping of the max and min operations is a direct consequence of the strong duality of the problem Bertsekas (1999) (strict convexity of the problem, strictly feasible general convex constraints and affine constraints) As,

$$\underset{z_{t,n}\in\mathbb{R}}{\operatorname{argmin}}\ z_{t,n}^2 + \lambda_{t,n}\left(z_{t,n} - d_{t,n} + \sum_{c=1}^{C} u_{t,n}^c\right) = -\frac{\lambda_{t,n}}{2},$$

the proof is concluded by substituting each $z_{t,n}$ by this solution.

Once converged, u^* is equivalent to the primal solution, the solution λ^* to (3) is an optimal price multiplier. We will show that at the optimal point (u^*, λ^*) drivers' constraints are respected and yet need not be communicated, privacy is preserved. Indeed, the optimal solution u^* for the entire fleet is the concatenation of the individual optima $(u_c^*)_{c=1\dots C}$. The calculation of u_c^* for a particular value of c only depends on messages obtained from the rest of the fleet through λ^* as it is the solution to

Proposition 2. ***Agent level problem:*** *Once the optimal value of λ in* (3) *has been found (which will be explained in Algorithm 2), it is sufficient for each agent to solve the sub-problem*

$$\min_{u^c\in C^c} f_c(\lambda^*, u^c) = \sum_{t=1}^{T}\sum_{n=1}^{N}\left[\lambda_{t,n}^* u_{t,n}^c + \sigma_{t,n}\left(u_{t,n}^c\right)^2\right] + \sum_{t=1}^{T-1}\sum_{n=1}^{N}\rho_{t,c,n}\left(u_{t+1,n}^c - u_{t,n}^c\right)^2 \tag{6}$$

to find its optimal schedule.

Proof 2. *This proposition is a consequence of* (3) *and strong convexity Boyd and Vandenberghe (2004) which implies that once the optimal Lagrangian value λ^* has been found for λ, it is sufficient to find the optimal value of u which solves the problem*

$$\sum_{c=1}^{C}\min_{u^c\in C^c} - \sum_{t=1}^{T}\sum_{n=1}^{N}\lambda_{t,n}^* d_{t,n} + \sum_{c=1}^{C}\left(\sum_{t=1}^{T-1}\sum_{n=1}^{N}\rho_{t,c,n}\left(u_{t+1,n}^c - u_{t,n}^c\right)^2 + \sum_{t=1}^{T}\sum_{n=1}^{N}\sigma_{t,n}\left(u_{t,n}^c\right)^2\right).$$

This sum is separable across the different agents whose constraints are independent which means that to minimize the sum it is sufficient to minimize each of its element independently. This concludes the proof.

Algorithm 1: Dual splitting algorithm
Data: Constraints $(C^c)_{c\in\{1...C\}}$, target $d \in \mathbb{R}^{T\times N}$ **Result:** Optimal action u^*_c for driver c 1 Compute optimal Lagrangian multiplier λ^* by solving (3). 2 Compute optimal action $u^*_c = \underset{u^c\in C^c}{\text{argmin}} f_c(\lambda^*, u^c) = \sum_{t=1}^{T}\sum_{n=1}^{N}\left[\lambda^*_{t,n}u^c_{t,n} + \sigma_{t,n}\left(u^c_{t,n}\right)^2\right] + \sum_{t=1}^{T-1}\sum_{n=1}^{N}\rho_{t,c,n}\left(u^c_{t+1,n} - u^c_{t,n}\right)^2$

Dual splitting algorithm The considerations above lead to a straightforward separation of the problem through the dual.

Section 3 describes a gradient ascent based algorithm that achieves the step 1 in Algorithm 1. This distributed privacy preserving optimization scheme is provably able to track a demand distribution. In Figures 7 and 8, we show how the optimal distribution of vehicles is able to track the distribution of demand as it evolves in time. These plots are generated with 10^3 and 10^4 vehicles whose availabilities are generated at random with a uniform law for the beginning and the end of the vehicle shift. In order to check how the scheme performs we compare the demand distribution and the offer distribution with a Sum-Of-Squares metric (Kullback–Leibler divergence cannot be used here as the supports may be disjoints). This result is expected and confirms that the privacy-preserving distributed optimization method we presented is efficient to match demand and offer when constraints on the offer form a cartesian product of independent sets.

2.3. *Optimal Lagrangian value as a coordination price enabling privacy*

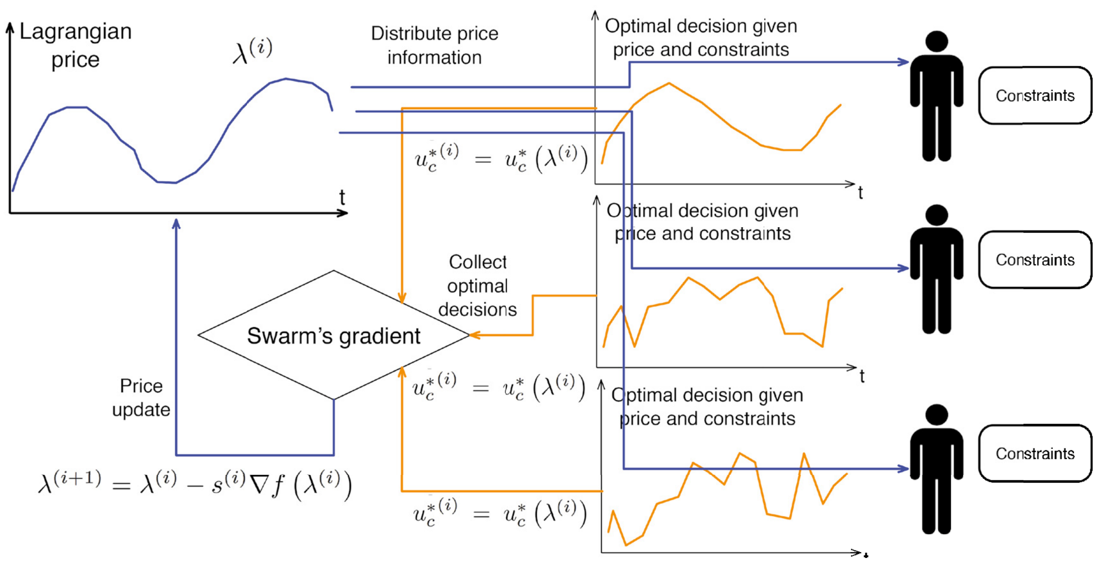

Fig. 5. Architecture for the distributed convex optimization algorithm enforcing privacy for a swarm of smartphones. At each step of the dual gradient ascent, a consensus variable is updated without the need for communicating the availability constraints of drivers on the communication medium. The individual contribution to the group's gradient update can also be perturbed so as to obfuscate them. This privacy-preserving algorithm can be modified and track smooth changes dynamically as it is intrinsically iteratively converging to the optimum. In this architecture, each phone locally computes the solution of a small size Linearly Constrained Quadratic Program which provides each driver with an optimal decision while maintaining his personal availability constraints and preferences confidential.

One of the most interesting outputs of the optimization scheme we propose more λ^* than u^*. Indeed, while u^* gives access to the fleet management, λ^* is a coordination signal in that it is sufficient to broadcast it to all the vehicles involved in the fleet for them to individually change their planning and update the next price with respect to which the fleet is going to coordinate. This result enables two important extensions to the scheme we propose. Splitting by the dual leads to the formulation of a Lagrangian vector λ^*. For each time step, $(\lambda^*_{t,n})_{n=1...N}$ projects a heatmap onto the discretization cells which optimally synthesizes the value that should be attributed to the different regions of the map so that drivers are optimally encouraged to meet the demand. This price takes implicitely into account the availability of the agents through time and the points at which they start and end their service. The optimal value in the dual λ^* is an optimal coordination Lagrangian price that makes taking agents' constraints implicit therefore eliminating the need for communicating them and centralizing them in a single agency.

Theorem 1. *Once λ^* has been found, no communication of constraints is necessary between the agents.*

Proof 3. *In proposition 6, it is sufficient for each agent c to find u_c^* solution to the individual problem*

$$\min_{u^c \in C^c} f_c\left(\lambda^*, u^c\right) = \sum_{t=1}^{T} \sum_{n=1}^{N} \left[\lambda_{t,n}^* u_{t,n}^c + \sigma_{t,n}\left(u_{t,n}^c\right)^2\right] + \sum_{t=1}^{T-1} \sum_{n=1}^{N} \rho_{t,c,n}\left(u_{t+1,n}^c - u_{t,n}^c\right)^2 . \tag{7}$$

This sub-problem can be solved independently across agents and only depends on λ^* and C^c for a given agent c. An agent c therefore does not need any knowledge of other agent constraints, $C^{c'}$ for $c' \in C \setminus \{c\}$ which thereby enforces the privacy of constraints.

Section 3 proves, using an analysis of the convergence of the algorithm we propose, that λ^* can be found without any communication of the agents' constraints.

Dynamic tracking. First, if the optimal solution changes slightly because of an unexpected variation of the demand or the addition or deletion of agents from the system, a new optimal price will be devised once the system has once more converged in the dual space that corresponds to the optimum in the primal space. Using a dual splitting reformulation of a convex relaxation approximation enables a re-computation in real time for the swarm of taxis. This allows adaptation to random variations and changes in vehicle availability.

Electric vehicles. Second, another interesting aspect is that constraints that are independent across vehicles can arbitrarily be added to the system without a modification of the scheme. The corresponding individual constraint set C^c is updated if there is a modification of the properties of the vehicle indexed by c. This naturally enables the optimization program to take into account constraints that entail the charging of electric vehicle batteries. Electric vehicles having a more restricted action radius (sometimes referred to as "range of anxiety"), the operators may want to take the corresponding additional availability constraints into account. These constraints are naturally decoupled across vehicles as they only relate to periods during which a given vehicle is not available for mobility as a service because it needs to be charged. Therefore the procedure we present is naturally able to handle fleets of mobility as a service with electric vehicles.

Enabling coordination while preserving privacy. We therefore showed how separating block independent constraints coupled in the objective by the dual is a technique that entails a natural privacy preservation mechanism. The most significant outcome of the dual splitting method is the formulation of an optimal Lagrangian multiplier λ^* which enables agents to coordinate their response to the demand while respecting their own availability constraints. The following section proves that convergence towards λ^* is robust and can occur without the need for the agents to communicate their constraints at any iteration of the privacy preserving convergence algorithm we devise.

3. Analysis of convergence and algorithm

One of the main objectives of this article is to build a clear analysis of the convergence properties of the method to provide runtime guarantees useful for practical implementations. This analysis can be conducted in a generic manner for any problem that can be interpreted as an optimal dispatch and therefore we will use generic notations in order to emphasize this aspect.

We prove the convergence of this algorithm based on the properties of gradient ascent methods described in Rockafellar (1997); Bertsekas and Tsitsiklis (1989) which needs no broadcasting of the constraints of the agents as explained in Figure 6.

3.1. Primal formulation as a consensus problem

We first reformulate the convex program above that resulted from applying a convex relaxation to the initial optimal dispatch problem.

We flatten the demand matrix $(d_{t,n})_{t=1\ldots T, n=1\ldots N}$ in which t is the temporal indexing of the demand intensity and n its spatial index. The demand intensity is from hereon renumbered as $(d_k)_{k=1\ldots T\times N}$. The primal optimization program obtained after convex relaxation can now be rewritten as follows:

Generic main primal problem:

$$\min_{\substack{u \\ \text{st } \forall c\in\{1\ldots C\},\, u_c\in C^c}} \left[\sum_{k=1}^{T\times N} \ell_k \left(d_k - \frac{1}{C} \sum_{c=1}^{C} u_{k,c} \right) + \frac{1}{C} \sum_{c=1}^{C} \mathrm{r}_c (u_c) \right]$$

Penalizations corresponding to failing to match the demand are represented by the family of strongly convex functions $(\ell_k(\cdot))_{k\in\{1\ldots T\times N\}}$ from $\mathbb{R}$ onto $\mathbb{R}$. By construction, all the elements of the family of regularizing functions $(\mathrm{r}_c(\cdot))_{c\in\{1\ldots N\}}$ from $\mathbb{R}^d$ onto $\mathbb{R}$ are strongly convex.

Practical computational issues for naive approaches. Such a problem, with strongly convex objectives and strictly feasible convex constraints can be solved in a generic but non-scalable manner by standard optimization solvers. A key point in the present article is to develop specific algorithms which efficiently leverage the structure of problem (8). For general $\ell_k(\cdot)$ and $\mathrm{r}_c(\cdot)$ satisfying the equations above, standard solver will require too much memory for a single computer to handle as soon as $C \geq 10^3$ and $T \times N \geq 10$. Therefore we want to find a strategy that distributes this problem across several machines. Note that numerous cities in the US have exceeded this number by a lot given the size of their fleet.

Privacy related issues. We also consider that the constraints $u_c \in C^c$ of each agent c should not be broadcast as they are privacy sensitive. This does not have implications on the formulation (8) of the problem but will be essential in the algorithms to solve it. Let u^* be the solution of the minimization problem (8). The optimal program $u_n^* \in \mathbb{R}^d$ of each agent also contains sensitive information and has to be difficult for an adversary to estimate (the meaning of this statement will be explicitly defined later in the article).

The dual splitting reformulation we propose offers the double advantage of offering a scalable method to solve the problem which as a byproduct does not requires the sharing over a network of individual agents' constraints.

3.2. Distributed optimization

This section introduces additional variables before using a dual reformulation that manages to split the problem with respect to the number of agents. As the number of vehicles, C, is the largest scale factor in the problem, it is the most suitable axis for parallelization. We leverage independence between blocks of constraints to render a low-memory and privacy preserving algorithm.

3.2.1. Introduction of additional variables

For each $k \in \{1\ldots d\}$, let $z_k \in \mathbb{R}^d$. The optimization problem now reads

$$\begin{aligned} &\min_{u} \sum_{k=1}^{T\times N} \ell_k(z_k) + \frac{1}{C}\sum_{c=1}^{C} \mathrm{r}_c(u_c) \\ &\text{st } \forall c \in \{1\ldots C\},\ u_c \in C_c \\ &\qquad \forall k \in \{1\ldots T\times N\},\ z_k = d_k - \frac{1}{C}\sum_{c=1}^{C} u_{k,n} \end{aligned}$$

We form the Lagrangian with $T \times N$ real dual variables $(\lambda_k)_{k\in\{1\ldots T\times N\}}$ corresponding to constraints $\forall k \in \{1\ldots T\times N\}$, $z_k = d_k - \frac{1}{C}\sum_{c=1}^{C} u_{k,c}$. Slater's conditions hold with the assumptions above (see Boyd and Vandenberghe (2004) for details). Minimization and maximization can therefore be swapped in the Lagrangian.

This proves that problem (8) is equivalent to

$$\max_{\lambda}\min_{u,z}\left[\sum_{k=1}^{T\times N}\ell_k(z_k)+\lambda_k z_k+\sum_{k=1}^{T\times N}\lambda_k\left(-d_k+\frac{1}{C}\sum_{c=1}^{C}u_{k,c}\right)+\frac{1}{C}\sum_{c=1}^{C}\mathrm{r}_c(u_c)\right]$$
$$\text{st }\lambda\in\mathbb{R}^{T\times N},\ z\in\mathbb{R}^{T\times N},\ \forall c\in\{1\ldots C\},\ u_c\in C^c$$

3.2.2. Block constraints and distribution of min operators

This formulation solves our need for parallelization across agents as the operator min with respect to each z_k and each u_c. Indeed, z and u are decoupled and the constraints $u_c \in C^c$ are independent by assumption. The independence of constraints among the agents is consistent with the fact that these are considered private. Now, considering the Fenchel-Legendre transform

$$\ell_k^*(\lambda_k)=\sup_{z_k\in\mathbb{R}} l_k(z_k)+\lambda_k z_k$$

of $\ell_k(\cdot)$, one has

$$\min_{z\in\mathbb{R}^d}\sum_{k=1}^{T\times N}\ell_k(z_k)+\lambda_k z_k=-\sum_{k=1}^{T\times N}\ell^*(-\lambda_k).$$

Also, denoting $\Pi_{c=1}^{C}C^c$ the cartesian product of the constraint sets,

$$\min_{u\in\Pi_{c=1}^{C}C^c}\frac{1}{C}\sum_{c=1}^{C}\lambda^T u_c+\mathrm{r}_c(u_c)=\frac{1}{C}\sum_{c=1}^{C}\min_{u_c\in C^c}\lambda^T u_c+\mathrm{r}(u_c).$$

This proves that problem (8) is equivalent to

$$\max_{\lambda}\left[-\sum_{k=1}^{T\times N}\ell_k^*(-\lambda_k)-\lambda^T d+\frac{1}{C}\sum_{c=1}^{C}\min_{u_c\in C_c}\lambda^T u_c+\mathrm{r}_c(u_c)\right].$$

We can now appreciate how the constraints can be taken into account independently one from another. In particular, this will split the memory requirement for a gradient method based numerical resolution in the algorithm presented below (Algorithm 2).

3.2.3. Extended value regularization functions

For each $c \in \{1\ldots C\}$, let $\overline{r_c}(\cdot)$ the extended value function that equals $r_c(u_c)$ whenever $u_c \in C_c$ and $+\infty$ otherwise. With the assumptions above, $\overline{r_c(\cdot)}$ is proper, closed and lower semi continuous. It is not differentiable in general but is strongly convex by assumption. Let σ_c its strong convexity constant. Generic convex analysis (see Rockafellar (1997) for details) allows us to show that the Fenchel-Legendre transform of $\overline{r_c(\cdot)}$, denoted $\overline{r_c(\cdot)}^*$, is differentiable and has a Lipschitz gradient with constant $\frac{1}{\sigma_c}$. It is also trivially convex. For any $c \in \{1\ldots C\}$, the strong convexity assumption on r_c also guarantees uniqueness of $u_c^*(\lambda)=\mathrm{argmin}_{u_c\in C_c}(\lambda^*)^T u_c+\mathrm{r}_c(u_c)$ where λ^* is the unique solution of problem (8) (see Boyd and Vandenberghe (2004) for details).

3.2.4. Formulating an optimal price

The problem (8) is now equivalent to the unconstrained minimization below.

Dual split reformulation:

$$\min_{\lambda\in\mathbb{R}^{T\times N}} f(\lambda)=\min_{\lambda\in\mathbb{R}^{T\times N}}\left[\sum_{k=1}^{T\times N}\ell_k^*(-\lambda_k)+\lambda^T d+\frac{1}{C}\sum_{c=1}^{C}\overline{\mathrm{r}_c}^*(-\lambda)\right] \tag{8}$$

For each $k \in \{1\ldots T\times N\}$ we denote L_k the Lipschitz constant of the gradient of $\ell_k(\cdot)$ and m_k the strong concavity constant of the function. As in Rockafellar (1997), $\ell_k^*(\cdot)$ has a Lipschitz gradient with constant $\frac{1}{m_k}$ and is strongly convex with constant $\frac{1}{L_k}$. Therefore, f is strongly convex with constant $m=\sum_{k=1}^{T\times N}\frac{1}{L_k}$ and has a Lipschitz continuous gradient with constant $L = \sum_{k=1}^{T\times N} \frac{1}{m_k} + \frac{1}{C}\sum_{c=1}^{C} \frac{1}{\sigma_c}$. The strong convexity property shows in particular that there is a unique price vector λ^* that synthesizes the information contained in the common objective and the constraints. Indeed, if agent c is given λ^* it is sufficient for it to individually solve $\min_{u_c \in C_c} (\lambda^*)^T u_c + \mathrm{r}_c(u_c)$ in order to retrieve the optimal action u_c^* that contributes to the overall objective best. In particular, this shows privacy sensitive constraints do not have to be shared with other participants in the system. Also, this problem is much less memory consuming to solve as only one constraint set corresponding to a single agent needs to be considered. Moreover, for a given value of λ, the sub-problems can be solved independently in parallel. This shows that this reformulation is privacy preserving, is computationally tractable and scales with respect to the number of agents being dispatched.

3.2.5. Holistic deterministic gradient descent

Gradient descent and momentum methods are both straightforward ways to minimize f in practice. We have $\nabla f(\lambda) = -\sum_{k=1}^{T\times N} \nabla \ell_k^*(-\lambda_k) + d - \frac{1}{C}\sum_{c=1}^{C} \nabla \overline{\mathrm{r}_c}^*(-\lambda)$.

Usual theorems for differentiating maxima of functions (see Bertsekas (1999) for details) give, $\forall c \in \{1 \dots C\}$, $\nabla \overline{\mathrm{r}_c}^*(\lambda) = u_c^*(-\lambda)$, therefore

$$\nabla f(\lambda) = -\sum_{k=1}^{T\times N} \nabla \ell_k^*(-\lambda_k) + d - \frac{1}{C}\sum_{c=1}^{C} u_c^*(\lambda) \tag{9}$$

.

From Rockafellar (1997), we know $O\left(\log\left(\frac{L}{m}\frac{1}{\varepsilon}\right)\right)$ iterations are sufficient for the distributed gradient based algorithm below (Algorithm 2) to achieve an ε precision in the value of the function we are trying to minimize.

Algorithm 2: Holistic distributed gradient descent

Data: Constraints $(C_c)_{c\in\{1\dots C\}}$, target $d \in \mathbb{R}^{T\times N}$
Result: Optimal dual price $(\mathrm{argmin}_{\lambda\in\mathbb{R}^{T\times N}} f(\lambda))$
decide on initial value $\lambda^{(0)} \in \mathbb{R}^{T\times N}$.
for $i \leftarrow 1$ ***to*** *maximum number of steps* **do**
    broadcast $\lambda^{(i)}$
    compute optimal response $u_c^*\left(\lambda^{(i)}\right)$
    broadcast $u_c^*\left(\lambda^{(i)}\right)$
    gather and compute concatenated $u^*\left(\lambda^{(i)}\right)$
    compute $\lambda^{(i+1)} = \lambda^{(i)} - s^{(i)}\nabla f\left(\lambda^{(i)}\right)$
end

Theorem 2. *Algorithm 2 solves the dual split problem without requiring the drivers to communicate their constraint sets. At any step i, the only communication needed from agent c is* $u^*\left(\lambda^{(i)}\right)$*, which does not involve the constraint set* C_c*. Only the local computation of that optimal response by the agent required knowledge of the agent's constraints.*

Proof 4. *With regularity and strong convexity assumptions above, gradient descent in algorithm 2 converges at an linear rate towards* λ^* *and therefore Theorem 1 completes the proof.*

3.2.6. Parallelism with respect to the agents

Let us precisely describe here how computation of the gradients are based on independent calculations by the agents. Indeed, calculating $\nabla \overline{\mathrm{r}_c}^*(-\lambda) = u_c^*(\lambda)$ is the only point where agents' constraints are to be taken into account and they are completely decoupled here. In particular, if each agent computes this step locally, it does not have to give any information about its individual constraints to others. This means that the reformulation above yields an intrinsically privacy preserving gradient method which does not require individual constraint to be communicated or centralized at any point.

3.3. Privacy preservation through obfuscation: approach and architecture

In the procedures above, the aim of enforcing the confidentiality of individual's constraints has been successfully achieved. This section will focus on obfuscating information that would help infer those indirectly. It is now a supplementary privacy-preserving measure that complements the privacy preservation of drivers' constraints.

Objective for privacy preservation through obfuscation: Obfuscation has to protect each agent's optimal program, $u_c^*(\lambda^*)$, from an opponent that is eavesdropping on the broadcast channel.

A first obvious solution is to de-identify the vectors $\mathring{u}_n$ that are communicated to other agents so as to compute the new value of λ. Indeed, computation of λ is aggregated with respect to all individuals so as long as each $\mathring{u}_n$ is included once and only once in the process, the result is not changed. In the following we will consider a safer procedure that obfuscates the individuals' programs by adding white noise to communications.

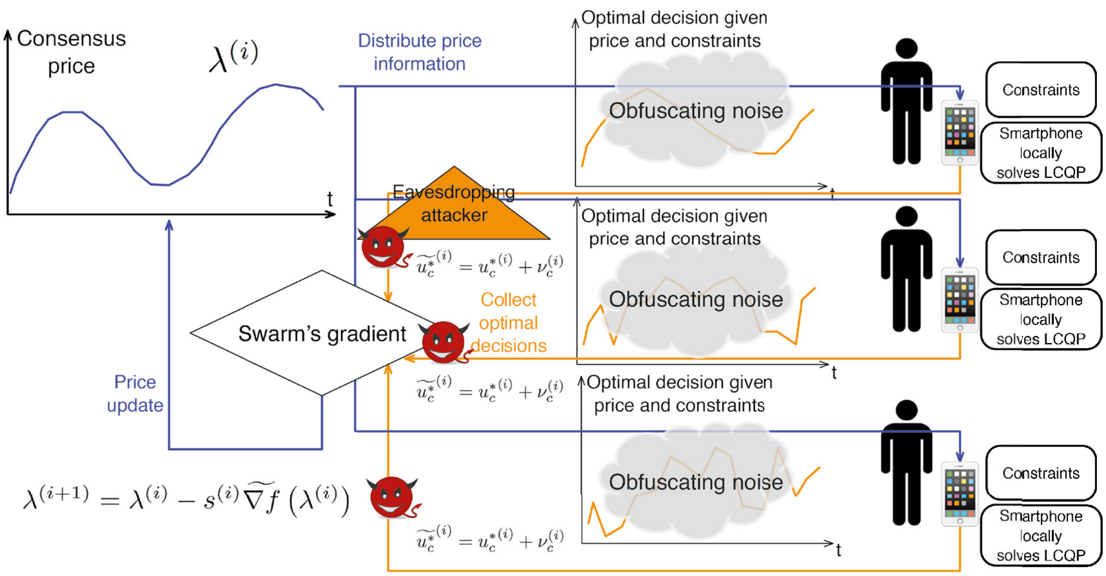

Fig. 6. Obfuscated distributed convex optimization algorithm enforcing privacy for a swarm of smartphones. Noise is added to outbound communication from the smartphones in order to obfuscate individuals' information related to the trajectories while enabling an accurate gradient update as the gradient is computed as an average across agents.

3.3.1. Stochastic privacy model

The specific approach chosen is to protect against a man-in-the-middle attack Bishop (2004) which occurs in the system that allows the opponent listens to any broadcast message Agrawal and Srikant (2000); Sweeney (2002). We further assume that this adversary has a perfect ability to re-identify message senders. Algorithm 2 is interpreted as a survey in which, at each iteration i, all agents are queried for their optimal action $u_c^*\left(\lambda^{(i)}\right)$. Agents do not have to send out their personal sets of constraints, C^c, for the dual optimum λ^* to be estimated. However, they send out vectors $u_c^*(\lambda)$ which correspond to the optimal series of actions to undertake with respect to a given signal vector λ. This information is considered privacy sensitive as it can help infer the agents' trajectories. Therefore, one considers a framework close to that of Warner (1965), in which participants in a survey are reluctant to give out personal data. It is possible here to leverage the averaging behavior of the dual gradient in order to compute the common optimum of the whole community without jeopardizing individual's privacy.

Consider iteration i of Algorithm 2, the set of optimal programs being broadcast is $u_c^*\left(\lambda^{(i)}\right)$. The attacker can keep these messages in memory and then compare them with $u_c^*\left(\lambda^{(i)}\right)$. Although messages have been de-identified, we consider the attacker can break that protection. In the example below, smartphones are the fundamental computational nodes of the procedure. An adversary can target a given device and listen to its outgoing messages with perfect knowledge of the home address they are being sent from. We assume that the only ability the attacker does not have is accessing the computations inside the nodes that enclose individual constraints. This assumption is standard in a man in the middle attack scheme Bishop (2004).

3.3.2. Noise obfuscation of broadcast

Adding artificial noise to communicated data is a powerful tool used in differential privacy for databases in Dwork et al. (2006b,a); Domingo-Ferrer et al. (2004); Sarathy and Muralidhar (2011) and in filtering Le Ny and Pappas (2014) for example. Inspired by this work, we design an algorithm where, instead of sending $u_c^{*(i)} = u_c^*\left(\lambda^{(i)}\right)$, agent c broadcasts

$$\widetilde{u_c^*}^{(i)} = u_c^{*(i)} + \nu_c^{(i)} \tag{10}$$

in which the d-dimensional white noise sequences $(\nu_c^{(i)})_{i\in\mathbb{C}}$ are all mutually independent and have variance η^2 for each of their d components. This framework where only blurry observations of the gradient are available has also been studied in Duchi et al. (2013). The approach presented here diverges in that it intrinsically leverages the effect of having distributed processors taking part in the computation of the gradient. In particular a high value of C is core to obtaining good precision and at the same time privacy enforcement, as in many stochastic approaches to privacy.

3.3.3. Learning rate on personal information

In this setting, the system itself cannot be trusted and there is competition between the speed at which the community discovers λ^* and the rate at which a spying statistician can learn individual information. This attacker is trying to estimate $\widehat{u^*}_c^{(i)}$ based on a series of i observations $\left(\widehat{u^*}_c^{(j)}\right)_{j\in\{1\ldots i\}}$ for a given agent c that is targeted as an individual. Classically, when trying to estimate a vector from a series of linearly perturbed measurements, empirical mean estimators or Kalman filters yield a *Mean Squared Error* (MSE) that will scale proportionally to the variance $T \times N\eta^2$. Therefore we assume the attacker's estimator for $u_c^{*(i)}$ is unbiased and has variance $\mathbb{E}\left[\left\|\widehat{u_c^*}^{(i)} - u_c^{*(i)}\right\|_2^2\right] = \frac{T\times N\eta^2\kappa}{i^\gamma}$ where κ is a constant that depends on the estimation technique adopted by the adversary and γ it's learning rate.

The most favorable case for the attacker occurs when the sequence $\left(u_c^{*(j)}\right)_{j=1\ldots i}$ remains constant. The law of large numbers guarantees a convergence rate $\gamma = 1$ for the empirical mean estimator. Thus, from hereon, we will assume $\gamma \leq 1$.

The privacy enforcement criterion here is that the MSE of the estimator of the attacker remains above a certain lower bound $\kappa_{\min}$. This implies the optimization program has an iteration budget

$$i^{\max} = \left(\frac{d\kappa\eta^2}{\kappa_{\min}}\right)^{\frac{1}{\gamma}}. \tag{11}$$

3.3.4. Noisy descent

The privacy enforcing strategies below aim at converging towards an optimal scheduling price λ^* faster than the attacker increases its precision in the estimation of u_c^*.

A strategy to preserve privacy in the distributed gradient computation is to run the deterministic holistic descent above with noisy broadcasts from the agents. The update of λ in the descent becomes $\lambda^{(i+1)} = \lambda^{(i)} - s^{(i)}\widetilde{\nabla f}\left(\lambda^{(i)}\right)$ where

$$\widetilde{\nabla f}\left(\lambda^{(i)}\right) = -\sum_{k=1}^{T\times N} \nabla\ell_k^*\left(-\lambda_k^{(i)}\right) + d - \frac{1}{C}\sum_{c=1}^{C} \widetilde{u^*}_c^{(i)}\left(-\lambda^{(i)}\right).$$

Recalling that $\widetilde{u_c^*}^{(i)} = u_c^{*(i)} + v_c^{(i)}$, as $v_c^{(i)}$ is a white noise whose variance trace is $T \times N\eta^2$, $\mathbb{E}\left[\widetilde{\nabla f}\left(\lambda^{(i)}\right)\right] = \nabla f\left(\lambda^{(i)}\right)$ and

$$\mathbb{E}\left[\left\|\widetilde{\nabla f}\left(\lambda^{(i)}\right)\right\|_2^2\right] \leq A_{\text{hol}}^2 \left\|\lambda^{(i)} - \lambda^*\right\|_2^2 + B_{\text{hol}}^2. \tag{12}$$

where $A_{\text{hol}} = L = \sum_{k=1}^{T\times N} \frac{1}{m_k} + \frac{1}{C}\sum_{c=1}^{C} \frac{1}{\sigma_c}$ and $B_{\text{hol}}^2 = \frac{T\times N}{C}\eta^2$.

These considerations lead to the following "obfuscation by the crowd" theorem.

Theorem 3. *The precision that can be reached in terms of MSE for the numerical Lagrangian multiplier while obfuscating the agents' actions is inversely proportional to the size of the fleet.*

Proof 5. *Using the notation $m = \sum_{k=1}^{\times N} \frac{1}{m_k}$, if one uses step size $s^{(i)} = \left(m\left(2\frac{A_{hol}^2}{m^2} + i\right)\right)^{-1}$, after i iterations of the noisy holistic descent,*

$$\mathbb{E}\left[f(\lambda^{(i)}) - f(\lambda^*)\right] \leq \frac{2B_{hol}^2}{L\left(2 + \frac{im^2}{A_{hol}^2}\right)} \leq 2\frac{\frac{T\times N}{C}\eta^2 L}{im^2}. \tag{13}$$

As we defined L as a function of the average of $\frac{1}{\sigma_c}$ among the agents $c \in \{1 \ldots C\}$, L is indeed independent from the size of the fleet we consider.

It is noteworthy that the fact that the constants in this bound are sensitive to T and N does not change the conclusion regarding the role of the number of agents, C, which, as it increases, offers better opportunities for privacy enforcement by obfuscation.

4. Numerical results

The efficiency of these algorithmic and theoretical contributions to fleet allocation optimization is confirmed by numerical work conducted with on actual data collected in New York for Uber ride requests.

4.1. Data

Within the data set used consisting of Uber ride requests in 2014, we chose a random day, April 7th, to illustrate the properties of our algorithm with actual demand measured in Manhattan for Uber rides extracted by Flowers (2016). Drivers constraints where simulated at random with a start time picked uniformly at random through the day and an end time picked uniformly at random within the reminder of day after the start time. We divided the day into 24 one hour periods and aggregated the demand within these time bins. We also discretized the spatial space of the city with a tessellation grid of size 16×16.

4.2. Convergence towards a distribution of vehicles tracking the demand

We now demonstrate that our approach is effective with field data and is able to match the spatial and temporal distribution of demand for mobility as a service with the offer. We illustrate this with two different fleet sizes of 10^3 and 10^4 vehicles in Figures 7 and 8 respectively. The first set of plots in the former show that the offer tracks the demand in the planning that the algorithm has converged to. In the later we show how a simple map of λ^* can be interpreted as a price multiplier and formulates a coordinating scalar map that protects the privacy of drivers. In the figures below, we focus on normalized distributions of offer and demand as our algorithm is used with different fleet sizes which naturally leads to different absolute levels of offer. In particular we check that the distribution of vehicles the algorithm converges to does match that of the demand for rides. This is indeed observed in Figure 7 which confirms the theoretical guarantees of the previous section.

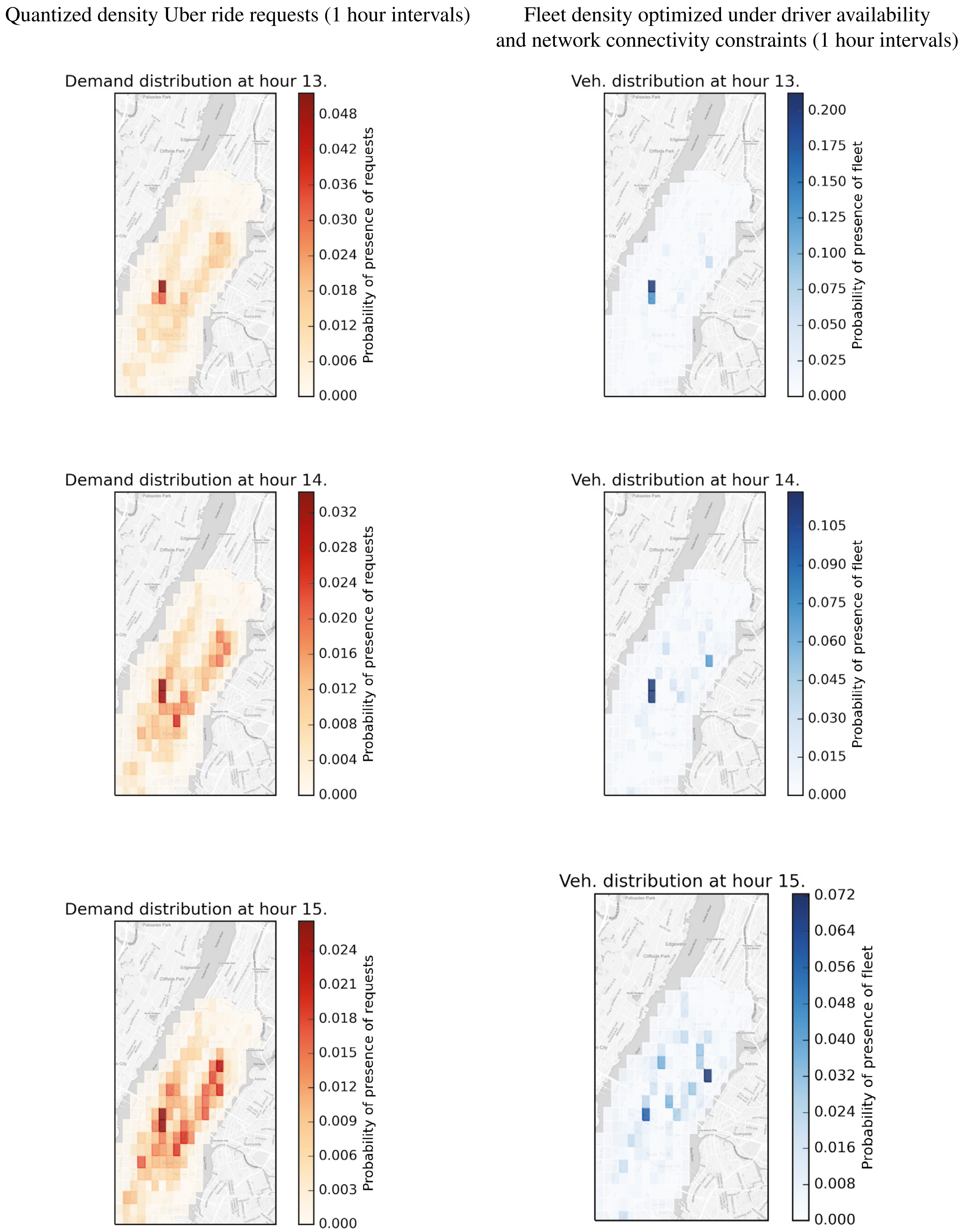

Fig. 7. Best viewed in color. Numerical results of our privacy preserving demand tracking algorithm for April 7^{th}, 2014, optimizing for 1000 vehicles. The vehicles belonging to the mobility on-demand pool obey to availability constraints. We also take into account the desire for drivers to minimize the distance they travel. The resulting optimized distribution of vehicles (number of vehicles per cell normalized by the total number of vehicles available) accurately tracks the distribution of demand for ride requests (number of ride requests per cell normalized by the total number of ride requests). Comparing the left and right columns shows the ability of our solution to track the demand for rides with the distribution of vehicles despite hard constraints on the offer in terms of available drivers. The constraints make a perfect match impossible but we find the closest solution given the variable and limited supply of drivers.

Lagrangian price signals (1 hour intervals) | Fleet density optimized under driver availability and network connectivity constraints (1 hour intervals)

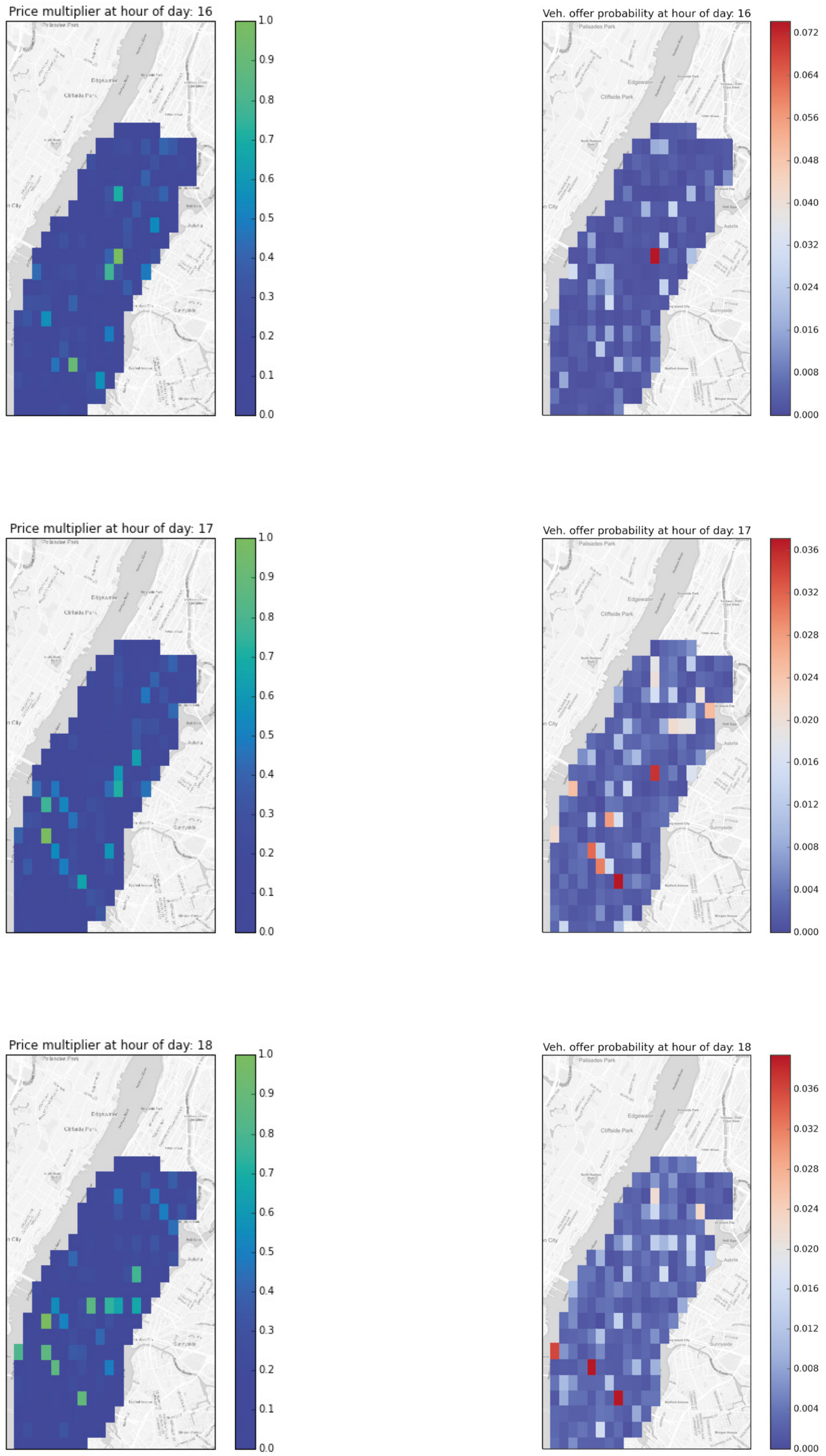

Fig. 8. Best viewed in color. Numerical results of our privacy preserving demand tracking algorithm for April 7^{th}, 2014, optimizing for 10000 vehicles. We show here that one of the most compelling outcomes of the dual-splitting approach is the convergence in the dual space towards a Lagrangian variable that can be interpreted, after renormalization, as a price multiplier. On these two series of plots one can appreciate how a single signal broadcast to all the devices leads to a scalable convergence of the fleet towards a distribution that tracks the multiplier. Note that the price multiplier drives the fleet towards locations where it is higher.

4.3. Influence of noise

This section focuses on showing the robustness of the scheme with respect to noise added in the broadcasts as in equation (10). Adding different Laplace noises in multiple gradient descents as in Figure 9 shows that the scheme we present is robust to perturbations. Indeed, the randomness in distributions of L_2 norm between offer for rides and demand does not explode as the noise becomes higher in variance. On average over 20 experiments there is indeed convergence towards an equilibrium.

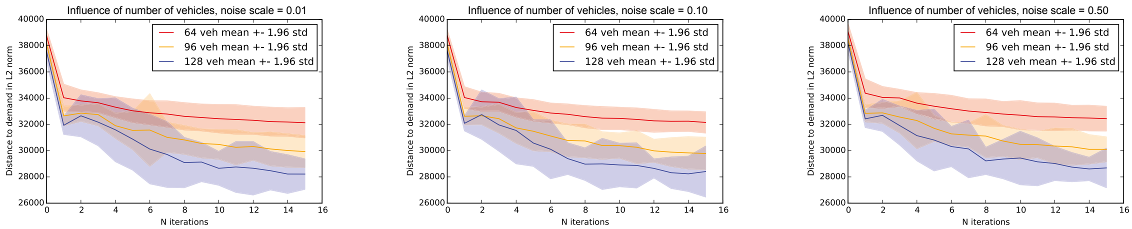

Fig. 9. Best viewed in color. On these plots, we show how adding noise to the gradient descent in the dual, therefore making it stochastic, does not hamper the convergence towards an optimal Lagrangian price and therefore a optimal matching between offer and demand. These curves show how, with different Laplace noise magnitudes and different fleet sizes, convergence to an optimal matching of offer and demand still occurs.

4.4. Increasing the number of vehicles and obfuscation by the crowd

As demonstrated in the theoretical analysis of privacy through obfuscation, the higher the number of vehicles, the higher the number of iterations the stochastic gradient method can go through without revealing too much of the actions the vehicles will undertake. We test the implications of the analysis in a numerical experiment in which we increase the number of iterations linearly with the number of vehicles in the fleet as prescribed in (11) for the worst case scenario of $\gamma = 1$. Figure 10 shows how with a higher number of vehicles the supplementary number of iterations does enables quite a substantial improvement. This improvement is not only due to the higher availability of cars as the L_2 distance between offer and demand steadily decreases until the end of each gradient method we present.

4.5. Influence of the model parameters

The previous paragraph demonstrated that the method we present is able to reach an optimal schedule in spite of only being allowed to compute a limited number of gradient method steps to preserve privacy. It can be observed on Figure 7 that if we only use 1000 vehicles the tracking of the demand by the agent presence

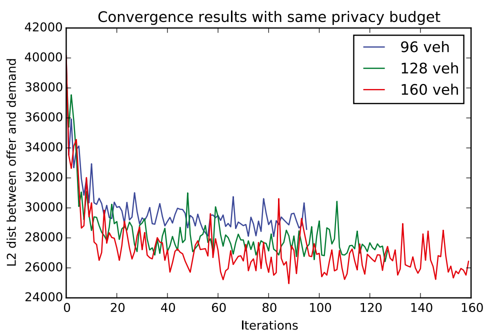

Fig. 10. Best viewed in color. On these curves we show the convergence results on single runs for a variable fleet size. With a larger fleet size, the obfuscation by the crowd of the information entailed in the messages shared to conduct the stochastic gradient method in the dual space allows for a better matching between offer for rides and demand.

probability is only approximate. In this section, Figure 11 shows that this matching of offer with demand can be improved as soon as we consider more vehicles (2000 in this paragraph). Indeed, as we increase the number of vehicles, more degrees of freedom are available and demand for rides can better be answered. The remainder of this section therefore focuses on three questions that still need to be answered. First, we need to characterize the influence of the number of spatial cells. After that, we will analyze how the number of discretization time steps influence impacts the convergence towards a solution. A third concern we address is that of being able to compute a solution at a speed adapted to the time discretization resolution. We show that coarser models allow for faster computation of a less accurate solution. We also demonstrate that over-simplification is not necessary as we are able to compute a solution in less seconds than there is time in a time step with a fined grained model. In this model, spatial cells are only the size of a few city blocks and time steps only have a duration of 15 minutes.

4.5.1. Influence of the spatial discretization granularity

In Figures 8, 7, 11, a discretization grid with small cells was used. More precisely, each cell is 0.41 km wide and 0.93 km tall.

A finer spatial discretization resolution is problematic in two aspects. As we target the average predicted demand, if fewer observations are available for an element of the state-space of the demand, more noise impacts our estimates. Also, the scale needs to be adapted to the dimensions of the city as we need each discretization cell to feature at least one street for it to be practically used by the drivers. As in Figure 1, some cells in central park only feature a single street and most cells only comprise of 10 city blocks.

Considering a coarser spatial discretization will have two positive effects on the solution that will be found. The estimates for demand will be more robust because we will leverage more information for each cell. The computational burden will also be lighter for each sub-problem that needs to be solved in the distributed algorithm we design. An obvious negative effect will obviously be the illusion that the problem is easier to solve when it fact this is only a direct consequence of using a less detailed model. The convex problem we solve with a coarser spatial grid at the level of each individual agent is by definition less complex but the schedule we obtain fits the demand profile less accurately.

In Figure 12, we indeed show that we converge faster with a discretization grid that is ten times as coarse than the initial grid we used.

4.5.2. Influence of the temporal resolution

The influence of the temporal resolution is similar, from an accuracy and computational complexity standpoint, to that of the spatial scale. Figure 12 clear demonstrates that it is much faster to converge with a solution with coarser time steps. This solution will only be adapted to a coarser approximation of the demand unfortunately.

Based on the empirical observations presented in Donovan and Work (2015), the mode of the pace in New York is 8 minutes for a single mile. A temporal resolution of 15 minutes is therefore reasonable choice given the characteristic time needed by an agent to traverse a fine spatial cell blocks in the presence of congestion. This means practically that from the point of the model, a given agent will be unlikely to "jump" over a spatial cell in less than a time step.

4.5.3. Computing a solution fast enough to be consistent with the temporal resolution of the model

Another critical aspect of the influence of the temporal resolution is related to an implicit computation speed requirement which is necessary to compute solutions for them to be relevant to the time steps of the optimization we are attempting to solve.

Having chosen a time step of 15 minutes, we need to be able to find a solution to our optimization problem in less than 900 seconds for it to be even relevant if unexpected perturbations to the demand require a quick re-computation of the solution. Figure 12 shows that the finest model we consider is solved in less than 900 seconds.

This result demonstrates that implementing the scheme on smartphones while providing on-line adaptation capabilities is feasible. Indeed, as only 400 Amazon EC2 CPUs were used in the experiment featured on Figure 12 and a general purpose convex optimization solver was employed to solve the agent-level optimization problems in the multiple steps of the algorithms. The large number agents plays a favorable role here. Our algorithm being scalable with respect to the number of vehicles, if more agents are present, using more CPU cores (practically the drivers' smartphones) will be sufficient. Moreover, more averaging of artificial privacy

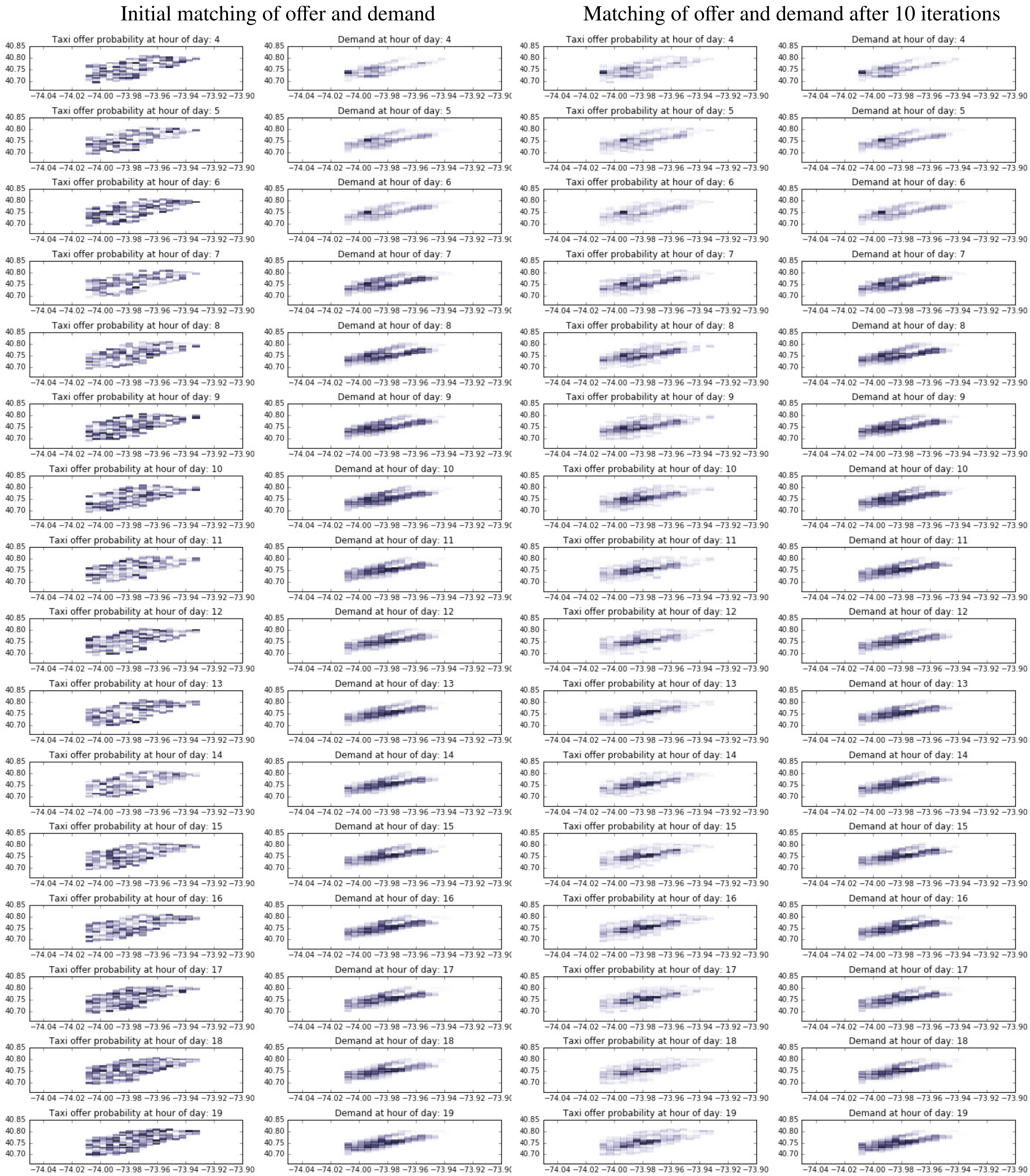

Fig. 11. Left: comparison of the offer initially computed with the demand for rides. Right: comparison of the offer obtained after 10 steps of gradient method with the demand for rides. After 10 iterations of the gradient method we introduced, the availability of 2000 vehicles considerably improves the quality of the solution we find to match the demand for rides. The discretization is here hourly, and the spatial scale as fine as an area representative of a few streets. The results were computed for April 9th 2013.

enforcing noise will occur in the algorithm and the convergence will be more stable. One EC2 Compute Unit offered for rent by this service provider runs on commodity hardware and is the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor. Showing that we can allow for each of these to support 5 agents in Figure 12 and yet be able to compute a solution in less than 15 minutes therefore indicates that this scheme can be use to have a reactive distributed control scheme supported by the smartphones of the drivers.

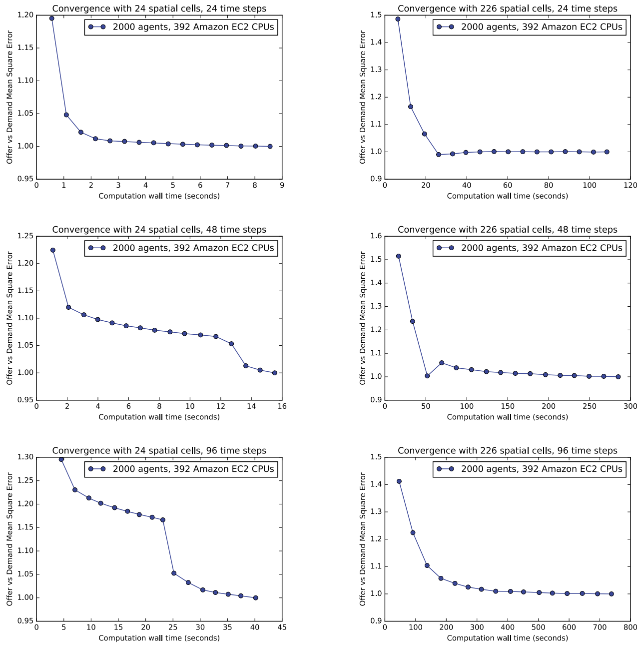

Fig. 12. Influence of the discretization resolution, in space and time, on the convergence speed of the distributed privacy preserving algorithm. In this experiments, 2000 vehicles are considered and only 400 Amazon EC2 CPUs are used. Here, even with additive noise on communications (privacy preserving Laplace noise with scale 0.1) the schedule can be optimized with a fine spatial resolution and temporal resolution of 15 minutes in less than 5 minutes. This implies that, even with a very detailed model, we can recompute solutions fast enough to adapt to unexpected conditions.

Therefore, with the method we propose, drivers can use a simple smartphone and do not need to compromise with information they want to remain private to optimize their actions as a group. The group will however be able to optimize its actions in less than 15 minutes and react to changing conditions. This advantageous property is ultimately a consequence of the approximate model we used whose assumptions are closely tailored to the operational setting of MaaS.

Conclusion

After a statistical characterization of the spatio-temporal structure of demand for MaaS in New York, we devised a CIQP to optimize the response to that demand while taken drivers' availability constraints into account. In order to make solving this problem computationally tractable and more tailored to the fact that MaaS companies such as Uber or Lyft do not directly control their drivers, a convex relaxation version of the problem has been used. Using the dual splitting approach paved the way towards a procedure to optimize the demand response that scales with the number of drivers and does not require them to communicate their availability constraints at any point, thereby enforcing strong privacy standards. The method formulates an optimal price multiplier akin to that of Uber or Lyft (so called "surge" or "primetime") that has the distribution of the fleet converge to a global optimum. A thorough theoretical analysis of the convergence of the dual gradient ascent algorithm we presented proved that the scheme was reliable, robust to perturbations and thereby enables an obfuscation of the optimal driving program that becomes more efficient as more drivers join the fleet. Indeed, the stochastic version of the gradient ascent Nemirovski et al. (2009) obfuscates the advised probability of presence of vehicles in order to protect the privacy of the users of the system in the occurence of an man-in-the-middle attack. Our theoretical contribution shows that this strategy converges to the initial optimum with an arbitrary precision provided enough drivers are involved in the optimization. This analysis of learning rates is different from pre-existing work in Duchi et al. (2013); Chathuranga Weeraddana et al.; Wainwright et al. (2012) and highlights the power of the crowd as a privacy preserving mechanism thanks to a thorough comparison of the convergence rates of the crowd and the attacker. Such strong theoretical guarantees are crucial to provide practitioners with an accurate estimate of their computational time. The theoretical analysis is generic enough to give guarantees for an entire class of constraint separable problems.

References

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

Agrawal, R., Srikant, R., 2000. Privacy-preserving data mining, in: ACM Sigmod Record, ACM. pp. 439–450.

Ak, A., Erera, A.L., 2007. A paired-vehicle recourse strategy for the vehicle-routing problem with stochastic demands. Transportation Science 41, 222–237.

Balan, R.K., Nguyen, K.X., Jiang, L., 2011. Real-time trip information service for a large taxi fleet, in: Proceedings of the 9th international conference on Mobile systems, applications, and services, ACM. pp. 99–112.

Ben-Akiva, M., Bierlaire, M., Burton, D., Koutsopoulos, H.N., Mishalani, R., 2001. Network state estimation and prediction for real-time traffic management. Networks and Spatial Economics 1, 293–318.

Bertsekas, D.P., 1999. Nonlinear programming .

Bertsekas, D.P., Tsitsiklis, J.N., 1989. Parallel and Distributed Computation: Numerical Methods. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Bishop, M., 2004. Introduction to Computer Security. Addison-Wesley Professional.

Bogers, E., Bierlaire, M., Hoogendoorn, S., 2015. Modeling learning in route choice. Transportation Research Record: Journal of the Transportation Research Board .

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine Learning 3, 1–122.

Boyd, S., Vandenberghe, L., 2004. Convex optimization. Cambridge university press.

Brillinger, D.R., 1981. Time series: data analysis and theory. volume 36. Siam.

Brockwell, P.J., Davis, R.A., 2013. Time series: theory and methods. Springer Science & Business Media.

Cai, H., Jia, X., Chiu, A.S., Hu, X., Xu, M., 2014. Siting public electric vehicle charging stations in beijing using big-data informed travel patterns of the taxi fleet. Transportation Research Part D: Transport and Environment 33, 39–46.

Chang, T., Nedic, A., Scaglione, A., 2014. Distributed constrained optimization by consensus-based primal-dual perturbation method .

Chathuranga Weeraddana, P., Athanasiou, G., Jakobsson, M., Fischione, C., Baras, J.S., . On the Privacy of Optimization Approaches. ArXiv e-prints `arXiv:1210.3283`.

Cheng, S.F., Nguyen, T.D., 2011. Taxisim: A multiagent simulation platform for evaluating taxi fleet operations, in: Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 02, IEEE Computer Society. pp. 14–21.

Domingo-Ferrer, J., Sebé, F., Castella-Roca, J., 2004. On the security of noise addition for privacy in statistical databases, in: International Workshop on Privacy in Statistical Databases, Springer. pp. 149–161.

Donovan, B., Work, D.B., 2015. Using coarse gps data to quantify city-scale transportation system resilience to extreme events. arXiv preprint arXiv:1507.06011 .

Duchi, J.C., Jordan, M.I., Wainwright, M.J., 2013. Local privacy, data processing inequalities, and minimax rates. arXiv preprint arXiv:1302.3203 .

Durango-Cohen, P.L., 2007. A time series analysis framework for transportation infrastructure management. Transportation Research Part B: Methodological 41, 493–505.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M., 2006a. Our data, ourselves: Privacy via distributed noise generation, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer. pp. 486–503.

Dwork, C., McSherry, F., Nissim, K., Smith, A., 2006b. Calibrating noise to sensitivity in private data analysis, in: Proceedings of the Third Conference on Theory of Cryptography, pp. 265–284.

Flowers, A., 2016. Uber foil request. URL: `https://github.com/fivethirtyeight/uber-tlc-foil-response`.

Gacias, B., Meunier, F., 2015. Design and operation for an electric taxi fleet. Or Spectrum 37, 171–194.

Goldfarb, D., Ma, S., 2012. Fast multiple-splitting algorithms for convex optimization. SIAM Journal on Optimization 22, 533–556.

Horni, A., Nagel, K., Axhausen, K.W., 2016. The multi-agent transport simulation matsim .

Jayakrishnan, R., 2015. A real-time algorithm to solve the peer-to-peer ride-matching problem in a flexible ridesharing system .

Le Ny, J., Pappas, G.J., 2014. Differentially private filtering. Automatic Control, IEEE Transactions on 59, 341–354.

Lee, S., Turner, J., Daskin, M.S., Homem-de Mello, T., Smilowitz, K., 2012. Improving fleet utilization for carriers by interval scheduling. European Journal of Operational Research 218, 261–269.

Liao, Z., 2001. Taxi dispatching via global positioning systems. IEEE Transactions on Engineering Management 48, 342–347.

Liao, Z., 2003. Real-time taxi dispatching using global positioning systems. Communications of the ACM 46, 81–83.

Lütkepohl, H., 2005. New introduction to multiple time series analysis. Springer Science & Business Media.

Ma, S., Zheng, Y., Wolfson, O., 2013. T-share: A large-scale dynamic taxi ridesharing service, in: Data Engineering (ICDE), 2013 IEEE 29th International Conference on, IEEE. pp. 410–421.

Maciejewski, M., Nagel, K., 2013. Simulation and dynamic optimization of taxi services in MATSim. Technical Report. VSP Working Paper 13-05, TU Berlin, Transport Systems Planning and Transport Telematics. www. vsp. tu-berlin. de/publications.

Mahmassani, H., Kim, Y., Jaillet, P., 2000. Local optimization approaches to solve dynamic commercial fleet management problems. Transportation Research Record: Journal of the Transportation Research Board , 71–79.

May, A.D., Cayford, R., Coifman, B., Merritt, G., 2003. Loop detector data collection and travel time measurement in the berkeley highway laboratory. California Partners for Advanced Transit and Highways (PATH) .

Miao, F., Han, S., Lin, S., Pappas, G.J., . Robust taxi dispatch under model uncertainties .

Miao, F., Lin, S., Munir, S., Stankovic, J.A., Huang, H., Zhang, D., He, T., Pappas, G.J., 2015. Taxi dispatch with real-time sensing data in metropolitan areas: a receding horizon control approach, in: Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems, ACM. pp. 100–109.

Nedic, A., Ozdaglar, A., 2009. Distributed subgradient methods for multi-agent optimization. Automatic Control, IEEE Transactions on 54, 48–61.
Nemirovski, A., Juditsky, A., Lan, G., Shapiro, A., 2009. Robust stochastic approximation approach to stochastic programming. SIAM Journal on Optimization 19, 1574–1609.
Pilanci, M., Ghaoui, L.E., Chandrasekaran, V., 2012. Recovery of sparse probability measures via convex programming, in: Advances in Neural Information Processing Systems, pp. 2420–2428.
Pilanci, M., Wainwright, M.J., El Ghaoui, L., 2015. Sparse learning via boolean relaxations. Mathematical Programming 151, 63–87.
Qian, X., Zhan, X., Ukkusuri, S.V., 2013. Characterizing urban dynamics using large scale taxicab data. Transportation Research Record 35, 37.
Robust, F., Daganzo, C.F., Souleyrette, R.R., 1990. Implementing vehicle routing models. Transportation Research Part B: Methodological 24, 263–286.
Rockafellar, R., 1997. Convex Analysis. Convex Analysis, Princeton University Press.
Sarathy, R., Muralidhar, K., 2011. Evaluating laplace noise addition to satisfy differential privacy for numeric data. Transactions on Data Privacy 4, 1–17.
Seow, K.T., Dang, N.H., Lee, D.H., 2010. A collaborative multiagent taxi-dispatch system. Automation Science and Engineering, IEEE Transactions on 7, 607–616.
Sun, H., Liu, H., Xiao, H., He, R., Ran, B., 2003. Use of local linear regression model for short-term traffic forecasting. Transportation Research Record: Journal of the Transportation Research Board , 143–150.
Sweeney, L., 2002. k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10, 557–570.
Wainwright, M.J., Jordan, M.I., Duchi, J.C., 2012. Privacy aware learning, in: Advances in Neural Information Processing Systems, pp. 1430–1438.
Warner, S.L., 1965. Randomized response: A survey technique for eliminating evasive answer bias. Journal of the American Statistical Association 60, 63–69.
Wong, K., Wong, S., Yang, H., 2001. Modeling urban taxi services in congested road networks with elastic demand. Transportation Research Part B: Methodological 35, 819–842.
Wu, J., Gao, Z., Sun, H., Huang, H., 2004. Urban transit system as a scale-free network. Modern Physics Letters B 18, 1043–1049.
Yang, H., Leung, C.W., Wong, S., Bell, M.G., 2010. Equilibria of bilateral taxi–customer searching and meeting on networks. Transportation Research Part B: Methodological 44, 1067–1083.