A Simulation Tool for Large-Scale Online Ridesharing

Filippo Bistaffa
IIIA-CSIC
Cerdanyola, Spain
filippo.bistaffa@iiia.csic.es

Juan Rodríguez-Aguilar
IIIA-CSIC
Cerdanyola, Spain
jar@iiia.csic.es

Jesús Cerquides
IIIA-CSIC
Cerdanyola, Spain
cerquide@iiia.csic.es

Christian Blum
IIIA-CSIC
Cerdanyola, Spain
christian.blum@iiia.csic.es

ABSTRACT
Ridesharing is a prominent collective intelligence application producing significant benefits both for individuals (reduced costs) and for the entire community (reduced pollution and traffic). We tackle the online ridesharing (ORS) problem with the objective of forming cost-effective shared rides among commuters that submit requests to be served in a short time period (i.e., in a few minutes). We demonstrate a web-based simulation tool that computes and shows cost-effective shared cars along with the optimal path for each car. Our tool internally employs an online optimisation approach that can tackle large-scale ORS problems originating from real-world data (i.e., with ~400 requests per minute). Specifically, our simulation tool uses data from a real-world dataset, i.e., the New York City taxi dataset.

KEYWORDS
Online ridesharing; online stochastic combinatorial optimisation

2 SOLUTION APPROACH
In real-world ridesharing scenarios, even the most simple approach previously proposed to solve online stochastic optimisation problems (i.e., a standard greedy approach) [9] is not a viable solution method, as the computation takes too long to keep pace with the incoming rate of requests. We also remark that, as argued by Nourinejad and Roorda [10], the centralised approaches so far proposed in the literature are not able to cope with the computational complexity associated to online large-scale ridesharing problems in metropolitan scenarios. Against this background, there is a clear need for novel techniques that can make good decision under extremely severe time constraints.

We solve the ORS problem by proposing an online approach with two main ingredients, i.e., a hybrid optimisation approach and a lookahead reasoning, as seen in Figure 1. Specifically, we tackle the ORS problem as a sequence of optimisation problems. Each problem depends on the coming rate of requests. We also remark that, as argued by Nourinejad and Roorda [10], the centralised approaches so far proposed in the literature are not able to cope with the computational complexity associated to online large-scale ridesharing problems in metropolitan scenarios. Against this background, there is a clear need for novel techniques that can make good decision under extremely severe time constraints.
Input stream of sets of requests
\( (R_1, \ldots, R_t, \ldots, R_h) \)

Output stream of sets of cars
\( (S_1, \ldots, S_t, \ldots, S_h) \)

Pool\(_t\) Offline approach

Time budget

Look-ahead reasoning

Candidate car generation

Generation time budget

Candidate cars

Selection time budget

Car selection

Output set of cars

Figure 1: Overview of our online approach. Double line indicates the input and the output of the algorithm.

Candidate car generation

Generation time budget

Candidate cars

Selection time budget

Car selection

Output set of cars

Figure 2: Overview of our hybrid optimisation algorithm. Double line indicates the input and the output.

We achieve this objective by employing a hybrid approach, motivated by the successful application of similar techniques to other large-scale optimisation problems [4]. In more details, we employ a probabilistic greedy algorithm to generate a reduced sub-problem of the original large-scale problem, which only considers presumably good candidate cars. This reduced problem is then solved by using an integer linear programming (ILP) solver. Figure 2 shows the general scheme of our hybrid optimisation approach.

Then, we put each one-shot solution in the context of the overall online problem with a look-ahead reasoning, which avoids the formation of cars that, although profitable at current time, could result in a loss of better opportunities in the future.

3 WEB-BASED SIMULATION TOOL

We provide a simulation tool that computes and shows cost-effective shared cars by means of the online solution approach discussed in Section 2. Our tool is implemented by a web-based\(^1\) interface, i.e., it does not require the installation of any additional client application. Figure 3 shows an example screenshot of our simulation tool.

The user can select several simulation parameters, e.g., the number of simulation ticks, the number of requests per tick, and the maximum waiting time for each request. Our interface then shows the result of the simulation over the map of the considered urban area (i.e., Manhattan in our case). For each simulation tick we show the cars formed by our optimisation approach, along with the optimal path for each car. Furthermore, we show the requests that are still unserved on the map. The user can pause the simulation at any time, and visualise the results for every simulation tick.

Specifically, the user has access to two different levels of information on the right side of the graphical interface:

- On the one hand, in the “Global Statistics” tab we provide high level information about the simulation by means of two plots. The first plot shows the number of requests in the pool (i.e., the number of still-unserved requests) for each simulation tick. The second plot shows the cumulative number of formed cars with respect to the number of passengers.
- On the other hand, the “Requests & Cars” tab provides detailed information about the current simulation tick, i.e., the list of the request in the pool and the list of the cars that have been formed. For better clarity, the user can visualise the optimal path of each car on the map, as shown in Figure 4.

\(^1\)Our simulation tool is available at http://necro.iiia.csic.es:5000.

ACKNOWLEDGMENTS

Bistaffa was supported by the H2020-MSCA-IF-2016 HPA4CF project.
REFERENCES


