Demo — Dial-a-Ride: A Green Shortest Path Algorithm

Lorenzo Maggi
CREATE-NET, Trento (Italy)
lorenzo.maggi@create-net.org

Mahsa Faizrahnemoon
Hamilton Institute, National University of Ireland Maynooth
mahsa.faizrahnemoon.2013@nuim.ie

ABSTRACT

We consider a Dial-a-ride scenario in which one on-demand bus has to find an efficient route to serve different users willing to reach different destinations. We implemented in MATLAB a shortest path algorithm that minimizes a linear combination of an environmental cost and of the total waiting plus travel time experienced by the travellers. We demonstrate the outcome of the algorithm on the map of Dublin with the software SUMO.

Categories and Subject Descriptors

I.6.3 [**Computing Methodologies**]: Simulation and modeling—*Applications*

Keywords

Bus On Demand, Minimum travel time, Green algorithm

1. INTRODUCTION

Dial-a-ride service, also commonly called demand responsive transport, is the new paradigm for public transport since it allows to tailor the bus' route and schedule to users' needs. Travellers are asked to book the service in advance by specifying their initial position and their intended destination. Each user is then provided with an estimate of the pick-up time, as well as the time at which she will reach her destination. This new form of transportation is clearly more efficient than the more classic fixed-route-fixed-timetable transportation system, both from the network operator's and from the users' perspective. In fact, it permits to reduce the fuel costs, to serve the users more promptly, and eventually to increase the overall efficiency of the transportation system.

Dial-a-ride systems are up and running in USA, Canada, Japan, as well as in several countries in Europe. Such systems has been conceived especially for serving areas where users' demand fluctuates too heavily over time to justify the deployment of fixed bus lines, e.g., rural areas or big cities overnight.

We implement a variant of the dynamic programming algorithm first presented in [2] to design the bus route that minimizes a linear combination of the total service time experienced by all the users and an environmental cost. The shortest path algorithm has been implemented using MATLAB (MATLAB and Statistics Toolbox Release 2012b, The MathWorks, Inc., Natick, Massachusetts, United States). With the support of the software SUMO [1], we show the bus following the optimal route on the map of Dublin, and serving the passengers which are located in - and travel to - one of six different locations.

MobiHoc'14, August 11–14, 2014, Philadelphia, PA, USA.
ACM 978-1-4503-2620-9/14/08.
http://dx.doi.org/10.1145/2632951.2636058.

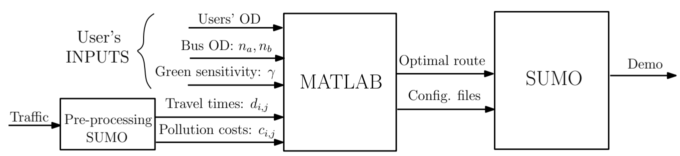

Figure 1: Demo flowchart.

2. GRAPH MODEL

The algorithm that we implement, which is a variant of the one first proposed in [2], provides the optimal route that one on-demand bus should follow in order to minimize a linear combination of an environmental cost and of the total waiting plus travel time for all travellers. It models the road network as a weighted graph $G = (\mathcal{N}, \mathcal{E}, \mathcal{W})$. Each node $n_i \in \mathcal{N}$ represents a bus stop and a link $(n_i, n_j) \in \mathcal{E}$ denotes the presence of a road from n_i to n_j which does not pass through any other bus stop. Each link (n_i, n_j) is characterized by two non-negative weights, $d_{i,j}$ and $c_{i,j}$, describing the travel time between the nodes and the environmental cost that the bus incurs while traversing the link, respectively. For instance, $c_{i,j}$ can be defined as an increasing function of $d_{i,j}$ and of the population density around the road going directly from n_i to n_j.

Let $\mathcal{OD} = \{od^{(k)} = (n_o^{(k)}, n_d^{(k)})\}_k$ be the set of all origin-destination pairs of users waiting for the bus stop at time 0. The pair $od^{(k)}$ is the stream of users waiting at bus stop $n_o^{(k)}$ and intending to reach $n_d^{(k)}$. We call $u^{(k)} \geq 1$ the number of users whose origin-destination pair is $od^{(k)}$. Let n_a and n_b be the initial (at time 0) and final (after all users have reached their destinations) positions of the bus, respectively.

We want to minimize a linear combination of the total waiting plus travel time for all passengers and of the environmental cost of the journey, computed as the sum of costs of the individual links. Let $T_d^{(k)}(p)$ be the time at which user stream k reaches its destination when bus follows path p. Then our goal is to find the optimal p^* minimizing the following objective function:

$$p^* = \underset{p}{\operatorname{argmin}} \sum_{k \in \mathcal{OD}} u^{(k)} T_d^{(k)}(p) + \gamma \sum_{(i,j) \in p} c_{i,j}, \qquad (1)$$

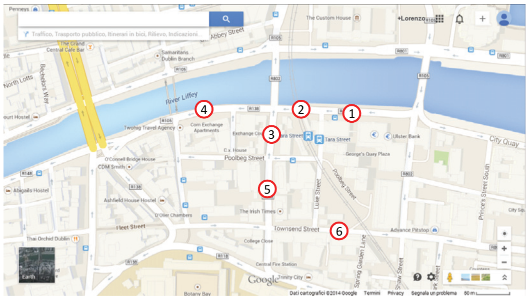

Figure 2: Position of six bus stops on the Dublin map.

where p is a generic feasible path going from node n_a to n_b and $\gamma > 0$ is the environmental sensitivity.

2.1 Shortest path formulation

In order to solve the optimization problem in (1) we utilize a shortest-path formulation. We define an augmented weighted graph $\widetilde{G}$, whose generic augmented node $\widetilde{n}_p \in \widetilde{\mathcal{N}}$ is the quadruple

$$\widetilde{n}_p = (n_i, O, W, S), \tag{2}$$

where $n_i \in \mathcal{N}$ represents the current bus position; O, W, and $S = \mathcal{OD} \setminus (O \cup W)$ are the streams of users that, immediately after the bus has passed by the stop n_i, are on the bus, still waiting at the respective origin nodes, and that have already reached their final destination, respectively.

Now we need to define the directed edges between the nodes in $\widetilde{\mathcal{N}}$. The node $\widetilde{n}_p$ defined as in (2) is connected to the node $\widetilde{n}_q = (n_j, O', W', S')$ whenever $(n_i, n_j) \in \mathcal{E}$ and

$$O' = O \setminus \left\{ od^{(k)} \in O : n_d^{(k)} = n_j \right\} \cup \left\{ od^{(k)} \in W : n_o^{(k)} = n_j \right\},$$

$$W' = W \setminus \left\{ od^{(k)} \in W : \; n_o^{(k)} = n_i \right\},$$

$$S' = S \cup \left\{ od^{(k)} \in O : \; n_d^{(k)} = n_j \right\}.$$

The weight $w'_{p,q}$ associated to the edge $(\widetilde{n}_p, \widetilde{n}_q)$ is

$$w'_{p,q} = d_{i,j} \sum_{k: od^{(k)} \notin S} u^{(k)} + \gamma \, c_{i,j},$$

where the first term is the incremental waiting plus travel time for users that have not reached their intended destination yet.

We intend to find the shortest path $\widetilde{p}^*$ between the initial and final nodes $\widetilde{n}_a$ and $\widetilde{n}_b$, where

$$\widetilde{n}_a = \Big(n_a, \left\{ od^{(k)} \in \mathcal{OD} : \; n_o^{(k)} = n_a \right\}, \mathcal{OD} \setminus \left\{ od^{(k)} \in \mathcal{OD} : \; n_o^{(k)} = n_a \right\}, \emptyset \Big)$$

and $\widetilde{n}_b = (n_b, \emptyset, \emptyset, \mathcal{OD})$.

2.2 Shortest path computation

The shortest path $\widetilde{p}^*$ in the augmented graph $\widetilde{G}$ from node $\widetilde{n}_a$ to node $\widetilde{n}_a$ determines the optimal route p^* for the bus (see Eq. 1). In fact, p^* can be read in the first entries of the succession of nodes compounding $\widetilde{p}^*$. We utilized the classic Djikstra algorithm to compute $\widetilde{p}^*$. In order to reduce the complexity of the algorithm, we first reduced the augmented graph $\widetilde{G}$ by performing a breadth-first search of all the nodes that can be possibly reached from the initial one, $\widetilde{n}_a$.

3. DEMO STRUCTURE

In the demo we considered 6 bus stops located in the centre of Dublin, as shown in Figure 2. By utilizing SUMO [1], we first estimated the travel times $d_{i,j}$, grouped in the matrix D, and then we computed the pollution costs as $c_{i,j} = k_{i,j} d_{i,j}$, where $k_{i,j}$ ranges from 2 to 4 according to the population density around the road link (n_i, n_j):

$$D = \begin{bmatrix} - & 8 & 19 & 19 & 44 & 21 \\ 32 & - & - & 10 & 62 & 65 \\ 32 & - & - & 10 & 62 & 65 \\ 53 & 96 & 97 & - & 82 & 85 \\ - & 43 & 43 & 15 & - & 31 \\ - & 45 & 45 & 52 & 29 & - \end{bmatrix}, \; K = \begin{bmatrix} - & 2 & 3 & 2 & 4 & 4 \\ 2 & - & - & 2 & 4 & 4 \\ 2 & - & - & 2 & 4 & 4 \\ 2 & 2 & 2 & - & 3 & 3 \\ - & 4 & 4 & 4 & - & 4 \\ - & 4 & 4 & 4 & 4 & - \end{bmatrix}$$

By convention, $d_{i,j} = -$ denotes the absence of the direct link, i.e., $(n_i, n_j) \notin \mathcal{E}$. We notice that the resulting graph is densely connected, which makes the computation of the optimal shortest path not trivial.

A demo user can set the following Matlab input (also see Figure 1):

- the Origin/Destination of the passengers, $\mathcal{OD}$, expressed as bus stops between 1 and 6;
- the initial and final position of the bus, n_a and n_b respectively;
- the environmental sensitivity $\gamma > 0$.

Our MATLAB code computes the optimal route as a succession of nodes $p^* = \{n_a, \ldots, n_b\}$ and automatically generates the required SUMO configuration files. Finally, SUMO displays the on-demand bus following the optimal route, picking up and delivering all the users to their intended destination. The bus then terminates its journey in the final node n_b.

We first point out that a trivial solution for the problem is to perform twice an Hamiltonian cycle starting from n_a. Since in each cycle the bus visits each node exactly once, all passengers are ensured to reach their destination. Clearly, such solution is suboptimal. We remark that, if there is a large amount of passengers with the same origin-destination pair, then most likely the bus route will serve them first, in order to minimize the total delay. Finally, increasing the environmental sensitivity γ clearly results in choosing a path p^* which crosses links with lower pollution costs.

4. CONCLUSIONS

We considered a Dial-a-ride system and we implemented in MATLAB a variant of the algorithm in [2], that computes the bus routes that minimizes the linear combination of an environmental cost and of the total waiting plus travel time for the users. We display the resulting optimal bus route on the Dublin map with the aid of the software SUMO [1].

Acknowledgments

This work was supported by the European Commission under TEAM, a large scale integrated project part of the FP7-ICT for Cooperative Systems for energy-efficient and sustainable mobility.

5. REFERENCES

[1] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.

[2] H. N. Psaraftis. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science*, 14(2):130–154, 1980.