Solving the First and Last Mile Problem with Connected and Autonomous Vehicles

Supplementary Materials

Alma Rahat
University of Plymouth, UK
Alma.Rahat@plymouth.ac.uk

Ralph Ledbetter
University of Exeter, UK
R.J.Ledbetter@exeter.ac.uk

Alex Dawn
City Science, UK
Alex.Dawn@CityScience.com

Robert Byrne
City Science, UK
Robert.Byrne@CityScience.com

Richard Everson
University of Exeter, UK
R.M.Everson@exeter.ac.uk

ACM Reference Format:
Alma Rahat, Ralph Ledbetter, Alex Dawn, Robert Byrne, and Richard Everson. 2019. Solving the First and Last Mile Problem with Connected and Autonomous Vehicles: Supplementary Materials. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3319619.3321919

1 INTRODUCTION

In this supplementary paper, we describe the evolutionary optimiser and present some further results and analyses.

2 EVOLUTIONARY OPTIMISATION

Evolutionary optimisation is a search procedure inspired from the theory of natural evolution. A generic algorithm starts with a randomly generated population of solutions. At each step of evolution, we select two parent solutions. These parents are combined to generate two children solutions. The combination operation is known as cross-over. It essentially induces large changes in solutions, and therefore enable exploration of the solution space. Each child solution is then mutated. The mutation operator is usually responsible for making small changes in a solution, and thus this operation promotes local search. The mutated children are then added to the population. The population is then pruned to the desired size by removing worst performing solutions. After a predefined number of evolutionary steps or generations, the best solution in the final population is the best approximation of the globally optimal solution. A more detailed account of the algorithm is in Algorithm 1.

During evolution, for selecting two parent solutions (step 3 in Algorithm), we use tournament selection strategy. In this scheme, n solutions are selected randomly, and then sorted according to their fitness (i.e. function values). Then a solution is selected with the probability $\pi_s(1-\pi_s)^m$, where π_s is a predefined probability and $m \in [0, n-1]$ is the rank of the solution with 0 being the rank of the best solution. In this paper, we set $n = 3$ and $\pi_s = 0.7$.

Algorithm 1 Evolutionary optimisation.

```
Inputs
    $k$ : Number of hubs
    $T$ : Number of evolutionary steps or generations
    $N$ : Population size

Steps
 1: $P \leftarrow$ RandomSamples($\mathcal{L}, N$)   ▷ Generate $N$ random solutions from feasible solution space $\mathcal{L}$
 2: for $i = 1 \to T$ do
 3:     $\{L^1, L^2\} \leftarrow$ TournamentSelection($P$)   ▷ Select two parent solutions
 4:     $\{c^1, c^2\} \leftarrow$ CrossOver($L^1, L^2$)   ▷ Combine two parents and generate two children
 5:     for $j = 1 \to 2$ do
 6:         $c^j \leftarrow$ Mutate($c^j$)   ▷ Mutate solution
 7:         $P \leftarrow P \cup \{c^j\}$   ▷ Add mutated child to population
 8:     end for
 9:     $P \leftarrow$ RetainBest($P, N$)   ▷ Retain the best N solutions in the population.
10: end for
```

The crossover operator takes some parts of one parent and and combines them with the complementary parts of the other parent to construct one child. The second child has the complementary mixture of elements from the parents in comparison to the first child. If the selected parents L^1 and L^2 both have the same number of hubs at the origin and destination, i.e. $|L^1_o| = |L^2_o|$ and $|L^1_d| = |L^2_d|$, for each group of hubs with a probability of π_c we set $c^1_x := L^2_x$ and $c^2_x := L^1_x$ where c^1 and c^2 are the generated children and $x \in \{o, d\}$. And with a probability $1 - \pi_c$ we set $c^1_x := L^1_x$ and $c^2_x := L^2_x$. In this paper, we set $\pi_c = 0.8$. It is possible that L^1 and L^2 may have different number of hubs in each group: $|L^1_o| \neq |L^2_o|$ and $|L^1_d| \neq |L^2_d|$. This makes cross-over challenging. In this case, when we assign the groups, the total number of hubs for a child may be smaller or greater than the desired total number of hubs k. We therefore repair such a child to ensure that it is of size k. If the a child $|c^i| < k$, with probability $\frac{1}{2}$ we add a feasible hub uniformly at random to either the origin group $x = o$ or the destination group $x = d$. The process is repeated until $|c^i| = k$. Similarly, when $|c^i| > k$, we remove a

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6748-6/19/07...$15.00
https://doi.org/10.1145/3319619.3321919

hub from each group instead. This repair scheme ensures that two valid children are generated through cross-over.

On a child solution c^i, the mutation operator may change group sizes $|c_o^i|$ and $|c_d^i|$, and/or any individual hubs. Firstly, with a mutation probability π_m, the group sizes are altered. This is performed by sampling a Dirichlet distribution $D(\alpha_o, \alpha_d)$, where α_o and α_d are concentration parameters. We set $\alpha_x = s \times \frac{|c_x^i|}{k}$ with $x \in \{o, d\}$. This effectively sets the concentration of the distribution at the same proportion of individual group sizes to the total length, and the scaling factor s controls the variation from this proportion. In our experiments, $s = 10$ produces desired variations in samples, that is a small alterations in groups sizes. It should be noted that a higher order of s reduces the amount of variation. A new sample from this distribution may change the original proportion that sums up to one. We therefore scale the new proportion up to k to determine individual group sizes $|c_o^i|$ and $|c_d^i|$. This may result in altering the total group size from k. To fix it, we remove or add to each group with equal probability until the desired size $|c^i| = k$ is reached, and thus we ensure that a feasible solution is generated. Secondly, alteration of any individual hub in c^i is performed with a probability π_m. In this case, we simply replace the current hub with one of the feasible alternatives uniformly at random. As we want to perform mutation infrequently, we set $\pi_m = 0.2$.

It is important to determine a stopping criterion for an evolutionary optimiser. Since, it is not guaranteed to converge to the global optimum, we only perform a number of evolutionary generations beyond which large improvement over the best solution is unlikely and the progression towards the optimum plateaus. The number of generations is usually determined by monitoring the convergence rates over independent runs of the optimiser.

3 CASE STUDY: COMMUTE BETWEEN EXMOUTH AND DIGBY

As a case study, we consider the commute between Exmouth and Digby in Devon, UK. Figure 1, shows the total time required for a journey by car or combined train and CAV system between OAs in Exmouth (vertical axis) and WZs in Digby (horizontal axis). There are many OA-WZ pairs between which there are no car journeys. Nonetheless the train and CAV system allow a potential link between them. It is visually evident that there are several car journeys which are more time consuming than train, and therefore placing a hub near such OA-WZ pairs may convert the car trips into train journeys. Our goal here is to automatically estimate the optimal locations for k hubs using the proposed evolutionary optimiser.

It should be noted that the most popular destination is: the 4th WZ (4th column in Figure 1) with identifier E00101374 that accounts for 81.8% of the 788 car journeys. This is where the Sowton Industrial Estate is located. We therefore expect that a good solution would include a hub at this location.

Following a short experiment, we set the population size to $10k$ and the number of generations for the optimiser to $k \times 10^3$. Note that increasing the number of generations may not yield large improvements (see for example Figure 2). As we generate two children per generation, $k \times 10^3$ generations results in $2k \times 10^3$ function evaluations. In Figure 3, we visualise the optimised solution. The solution clearly prefers the most popular destination, and also places hubs nearer major routes in Exmouth.

We also investigated the distribution of the hubs across OAs and WZs for all the solutions proposed by the optimiser for $k = 5 \longrightarrow 15$ (Figure 4). In many cases, for fixed k, the distribution of the number of allocated hubs at OA and WZ remain the same across solutions. This indicates that the optimiser may have located the optimal distribution of hubs for respective k. Moreover, the number of hubs allocated at WZ seems to be less than the number of hubs at OAs. This is primarily because the 4th WZ accounts for 81.8% of the car journeys: this hub appears in all solutions proposed by the optimiser. Therefore it is sensible to put more hubs at Exmouth side to ensure that more car trips are covered that make a journey to the industrial estate. On the other hand, the hubs at Exmouth side are generally placed near the station or major roads leading to the station.

4 CONCLUSIONS

One of the difficulties in using trains for commuting to work is the problem of travelling to and from a station. This is known as the first and last mile problem. In this report we discuss a potential futuristic solution to this problem with the use of CAVs. A commuter may take a CAV from a designated station called hub and rapidly travel to a station. So long as the total time to commute is less than a car trip to destination, the commuter will choose the train. With this, the challenge becomes where to place these hubs in order to maximise the number of people travelling by train. We model this problem as a combinatorial optimisation problem and propose an evolutionary algorithm to solve it. The proposed optimiser can locate a good estimate of the optimal distribution of hubs across the origin and the destination stations. We demonstrated it on a real world case study: the commute between Exmouth and Digby.

One of the major observations in this work is that there is a natural trade-off between the number of hubs and the number of car trips that may be saved. We will therefore consider this as a multi-objective optimisation problem in future. In addition, we considered the hubs to have unlimited capacity for simplicity. Future work will incorporate this constraint in optimisation.

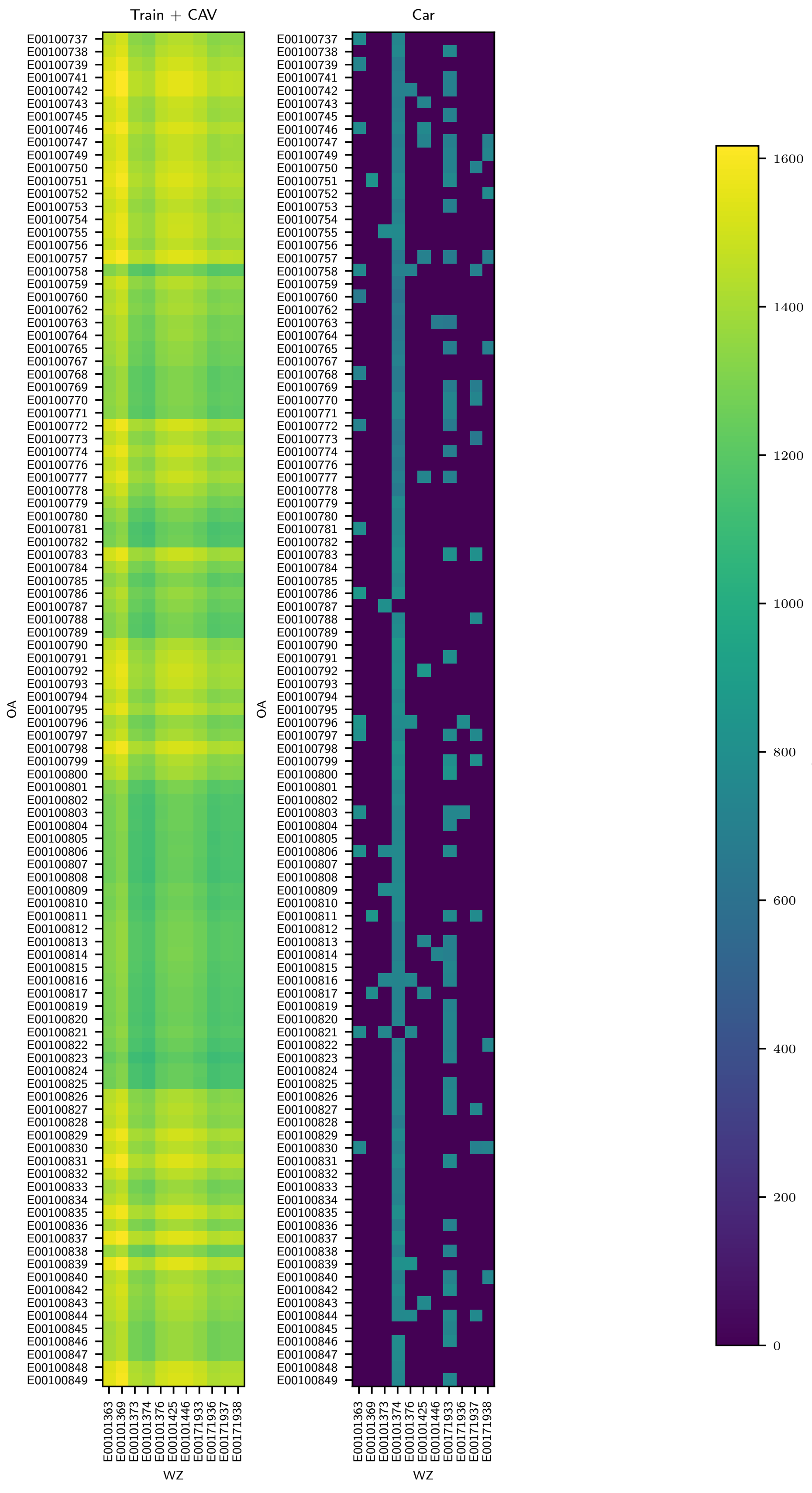

Figure 1: On the *left*, the time required for the combined CAV and train system if there is a hub at every origin OA and a hub at every destination WZ. On the *right*, the time required for a car journey is depicted. While performing optimisation, we only use k hubs and compute appropriate times for each OA-WZ pairs, so the matrix on the *left* varies based on the decision vector.

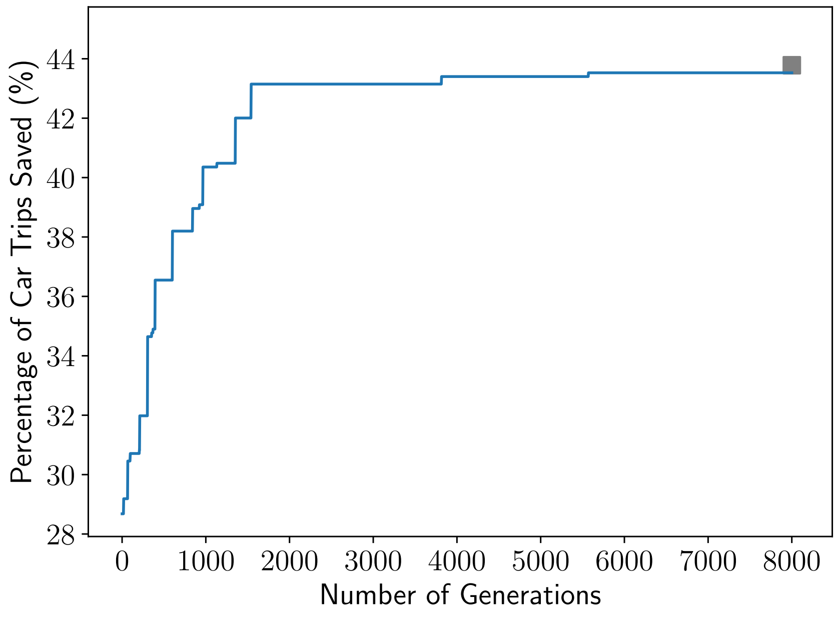

Figure 2: A sample convergence plot for $k = 8$ hubs. The progression plateaus after 5×10^3 generations, but we continue the evolutionary process until $k \times 10^3$ generations to improve the reliability of the optimiser. The grey square depicts the optimal solution that after $5k \times 10^3$ generations. Clearly, $4k \times 10^3$ additional generations resulted in a minute improvement.

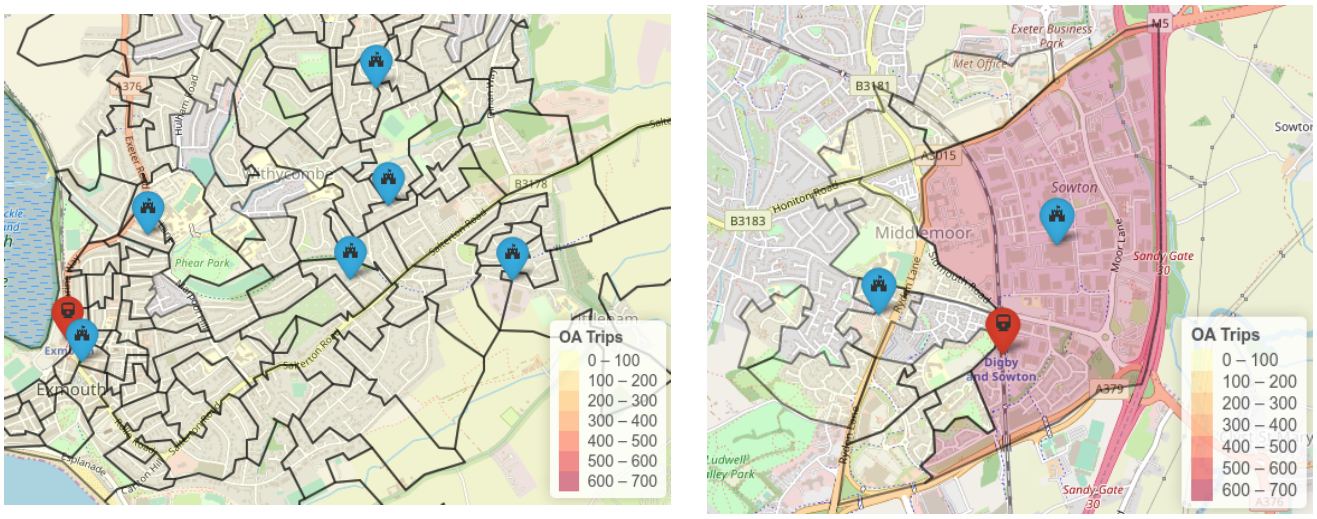

Figure 3: Visualisation of the solution located by the optimiser for $k = 8$. On the *left*, we show the origin station: Exmouth, and on the *right*, we show the destination station: Digby & Sowton. The red pins depict the stations, while the blue pins indicate the hubs. The black borders show the boundaries of the OAs and WZs, and diffrent shades of colours show how many journeys are taking place within the area. Interestingly, the hubs on the origin side are placed either close to the stations or nearer the major roads in Exmouth (e.g. Salterton Road and Bradham Lane) that facilitate faster commute to the station. In addition, there is a hub in the most popular destination: the 4th WZ.

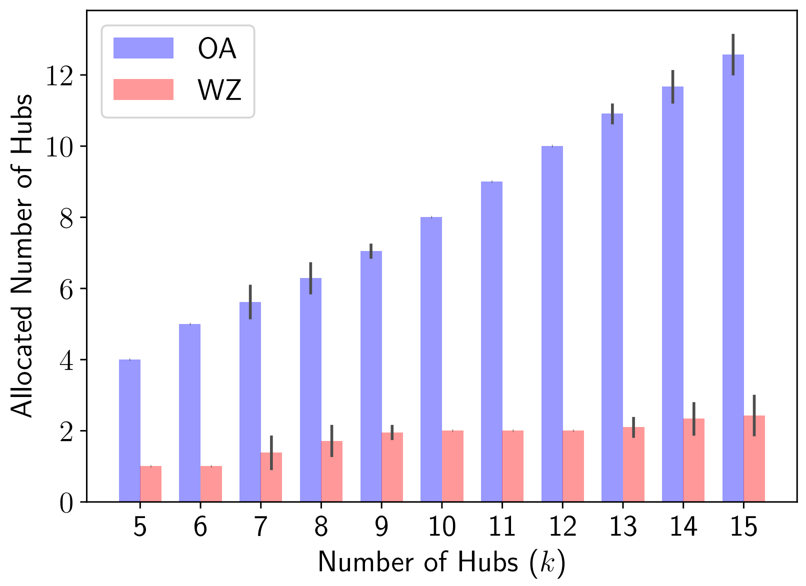

Figure 4: Distribution of hubs between OA (blue bars) and WZ (red bars) for all the optimisation runs for $k = 5 \rightarrow 15$. The standard deviation in the number of allocated hubs is shown with back vertical lines. The solutions clearly prefer small number of hubs (1–3) at the Digby WZs.