Optimizing Urban Road Networks for Automated Driving

Bahman Madadi[1*], Rob van Nes[1], Maaike Snelder[1,2], Bart van Arem[1]

[1]*Department of Transport & Planning, Delft University of Technology, The Netherlands*
[2]*TNO, The Netherlands*
[*]*Corresponding author, b.madadi@tudelft.nl*

Keywords: Automated Driving, Network Design Problem, Multi-class Stochastic User Equilibrium, Optimization

1. INTRODUCTION

Automated Driving (AD) is a trend that will evolve over time, both in the market penetration rate of Automated Vehicles (AVs) and the level of automation. According to SAE International [1], level 5 is the ultimate automation level with unlimited Operating Design Domain (ODD). However, in the transition period to full automation, there will be a mix of vehicles with different automation levels and a limited ODD. We envisage that AD for levels 3-4 AVs will be possible only on certain designated roads. For the remaining roads, manual driving will be mandatory (although supported by various assisting driving automation systems such as collision avoidance systems). On this designated roads, AVs will operate on the same lanes mixed with conventional vehicles (CVs). In order to facilitate safe and efficient AD in the mixed traffic on these selected roads, investments are required to meet certain design requirements. There will be trade-offs between these investments and the benefits they provide. This necessitates a network design approach to decide which roads should be selected to facilitate AD in the transition period. In this paper, we formulate this problem as a bi-level network design problem (NDP), and pay special attention to possible solution methods for this problem and their performance.

2. METHOD

In this section, we outline our proposed network configuration (which will be referred to as AD subnetwork) as well as the problem formulation, and we suggest solution methods for the problem.

2.1. Constructing the AD subnetwork

We envision a subset of an existing network within which AD is allowed everywhere for all automation levels and we refer to this as AD subnetwork. This corresponds to ODD of levels 3-4 AVs. First we specify a set of feasible links in the network that we assume to be safe for AD after some adjustments based on sustainable safety principles [2]. Next, we assume certain adjustment costs based on road types to meet design requirements for AD. Since investing on adjusting all the links in large networks can be infeasible and even unnecessary, the next step is to make an optimal selection (among feasible links) that guarantees highest possible gain from the investment (i.e., total adjustment cost).

2.2. Operational concepts and assumptions

It is assumed in this study that all vehicles are allowed everywhere in the network but AD and forming Cooperative Adaptive Cruise Control (CACC) platoons is only possible for AVs within the AD subnetwork in mixed traffic conditions (i.e. on the same lanes with CVs). Outside the AD subnetwork, all vehicles drive manually.

2.3. AD subnetwork as a bi-level NDP

Optimal investment decisions regarding road networks are often considered within the concept of NDP [3] where the objective is to optimize certain performance indicators for a network (e.g., total travel cost) considering the travelers' reactions to the network performance (route choices). This can be modeled as a bi-level leader-follower Stackelberg game in which the leaders (transport authorities) supply the transport infrastructure aiming at optimizing their objective function (e.g., total travel cost) and the followers (travelers) react with their route choice to optimize their own objective functions (e.g., individual travel costs). We model the optimal AD subnetwork construction problem as a bi-level NDP where the upper level entails the choice of links to include in AD subnetwork given a certain adjustment cost and the lower level represents the travelers' route choice.

We model the lower level problem as a multi-user class (MUC) stochastic user equilibrium (SUE) based on path-size logit (PSL). We distinguish two user classes (AVs and CVs) with different generalized travel cost functions. On links within the AD subnetwork, AVs are assumed to have a lower PCU to account for the shorter gaps between AVs and their leading vehicles, a lower value of time (VoT) because travel time can be used for other activities, and a lower value of distance (VoD) due to fuel efficiency of CACC [4], [5]. The generalized travel cost function is the summation of travel time multiplied by VoT, and distance multiplied by VoD. Link travel time is based on a modified BPR function where total flow is a weighted sum of class-specific flows to capture the correlation between link capacity and the proportion of AVs on the link.

We model the upper level problem as a goal programming (GP) problem with four objectives and four corresponding goal values and minimize sum of the normalized differences between each objective and its goal value weighted by the objective priority. The goal value for total travel cost (summation of individual generalized travel costs) is the solution of cost-based system optimal traffic assignment where all feasible links are included in the AD subnetwork. This corresponds to minimizing the price of anarchy. The same procedure is used for defining the goal value for total travel time based on time-based system optimal routing and the goal value for total travel distance is based on the shortest path assignment. The last goal is total adjustment budget for which different goal values are considered in different scenarios. This GP formulation allows for minimizing four objectives, or any combination of them, simultaneously and solving the multi-objective problem with solution methods developed for the single objective problem. Moreover, different priorities and target values for different objectives can be used with this framework. The decision variables are binary integers representing links to be included in AD subnetwork. The lower level equilibrium conditions are treated as constraints for the upper level problem.

2.4. Solution algorithms

An iterative optimization-assignment algorithm is used which iteratively solves the upper-level optimization problem with fixed values for lower-level decision variables (equilibrium flows), and the lower-level SUE problem with fixed values for upper-level decision variables (choice of links to include in AD subnetwork). To solve the lower level equilibrium, the MUC MSA algorithm introduced in [6] is used with moderate modifications. The upper level problem is a none-differentiable NP-Hard problem with no gradient information available. The value of the objectives measured through the lower level problem evaluations can be used to estimate gradient information; however, this means the lower level problem should be solved once per objective function evaluation. Solving the lower level problem also requires many iterations for convergence. This sets high requirements for the efficiency of the solution algorithm. Moreover, the problem includes multiple demand scenarios and many possible priority combinations for objectives that can further increase the computation time for a comprehensive analysis. Therefore, we compare the performance of three heuristics for solving the upper level problem. Genetic Algorithms (GA) [7] and Simulated Annealing Algorithms (SAA) [8] are two of the most commonly used heuristics to solve DNDP [9]. GA is known to find good (near-optimal) solutions for the DNDP but at the cost of many fitness function evaluations, which requires solving the lower level equilibrium problem once per evaluation. On the other hand, SAA generally requires less function evaluations but it might get locked in a local minimum. Although not common for solving DNDP, another viable heuristic algorithm is Simultaneous Perturbation

Stochastic Approximation (SPSA) introduced in [10] which requires three fitness function evaluations per iteration. This is lower than the required number for GA and higher than the required number for SAA making SPSA a good candidate for investigating the trade-offs between computational efficiency and solution quality. Therefore, we chose the three mentioned heuristics to solve the upper level problem and to compare their performance.

3. CASE STUDY, RESULTS AND DISCUSSION

We use a case study to demonstrate the concept of AD subnetwork and to compare the performance of the three heuristics. A network similar to the road network of Delft, The Netherlands is used and the network data and the demand patterns are based on a tutorial project in OmniTRANS transport modeling software package. It includes 1151 links, 494 nodes and 22 zones. A subset of links (corresponding to motorways, expressways and main roads) are selected as feasible links for AD subnetwork (Figure 1). We considered three demand scenarios with 10%, 50% and 90% penetration rate of AVs. As for objectives, we experimented with different priority combinations. In practice, the priorities for different objectives should be defined by transport authorities. For this extended abstract, we chose the 50% penetration rate scenario with equal priorities for total travel cost and total adjustment cost, and priorities equal to zero for the rest of the goals. The adjustment cost is discounted over ten years (assumed effective life time for the infrastructure adjustments) with 4% discount rate.

Figure 2 demonstrates the convergence of all three algorithms for the chosen scenario. All three heuristics seem to converge by the end of the run, yet each requires a different number of function evaluations. SPSA in this case converges after approximately 500 iterations, each requiring 3 objective function evaluations. The number of iterations for SAA was set to 500, but as evidenced by the figure, it converges in less than 300 iterations. The number of fitness function evaluations per iteration for SAA can vary based on acceptance threshold for deterioration in objective function. In this case, the number of fitness function evaluations was 1500. On the other hand, GA converges in about 300 generations each evaluating a population of size 120. Therefore, the general number of function evaluations for GA is much higher than the other two. Regarding the trends, the GA graph has the smoothest curve since it reports averages and best values of a population while SAA has the sharpest decreases due to high temperatures in first iterations and reannealing processes after the initial iterations. SPSA seems to have a reasonable convergence curve.

FIGURE 1 AD Subnetwork: feasible links for AD subnetwork are shown with (bright) green and the rest with (dark) blue.

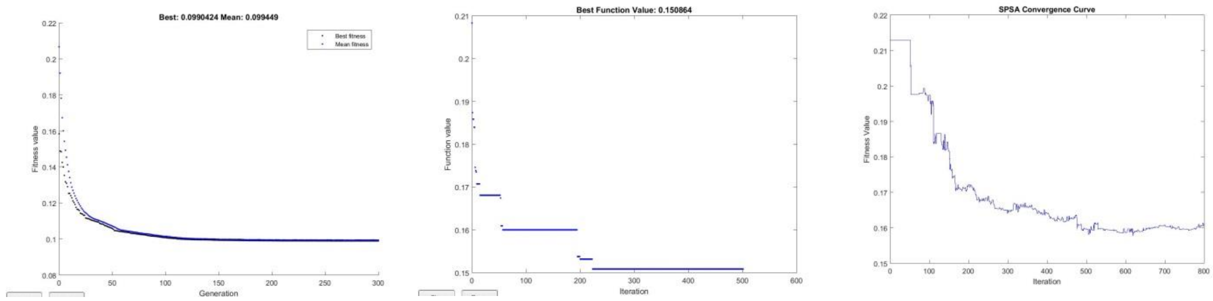

FIGURE 2 Objective function convergence curve of 50% penetration scenario for: (from left, respectively) GA, SAA, SPSA

Table 1 shows the results of 3 stochastically independent runs for each heuristic. GA predominantly outperforms the other two algorithms in the best objective function values found as well as total travel cost and total adjustment cost. However, as discussed earlier, this comes at the cost of significantly higher computation time. Among the other two heuristics, SAA tends to find configurations with slightly lower total travel cost with higher total adjustment costs while SPSA leads to opposite trade-offs between adjustment cost and travel cost. An interesting observation is that GA converges to the same results every time after a certain number of generations while there is some variety in results obtained by SPSA and SAA.

Parameter tuning for these heuristics is also an important issue. GA only requires a minimum number of population size (in this case about 120), elite count (about 20), and a minimum number of generations (about 250) to find plausible results, making the parameter tuning a straight forward procedure. On the other hand, SPSA has several parameters related to the step size and the gain sequence that can have different optimal values based on the scenario. This means the value of the optimal parameters depend on the ratio of changes in the objective function (gain) to the changes in decision variables (step size). This implies that scenarios with lower values of unary objective space require different parameters than scenarios with higher values of unary objective space. Thus for each scenario, SPSA parameters must be tuned and the performance of the algorithm is highly dependent on these parameters. This can be dealt with as a preprocessing step prior to multiple optimization runs with different settings. As for SAA, its performance seems to depend more on the energy landscape rather than its parameters. That is, it performs well in energy landscapes full of hills almost regardless of its parameters' values and performs poor in rather flat energy landscapes despite the amount of parameter tuning effort.

TABLE 1 Comparison of algorithm performances in three independent optimization runs for the 50% penetration rate scenario

Algorithm	Run Number	Objective function value	Total Travel Cost	Total Travel Time	Total Travel Distance	Total Adjustment Cost	Number of Links Selected
GA	1	0.099042	93241	5493	280535	345162	260
GA	2	0.099042	93241	5493	280535	345162	260
GA	3	0.099042	93241	5493	280535	345162	260
SAA	1	0.152708	97388	5501	279457	470160	197
SAA	2	0.150864	96690	5497	280007	536121	203
SAA	3	0.156241	97342	5502	279604	517020	207
SPSA	1	0.157739	98302	5506	279916	286330	161
SPSA	2	0.150211	97507	5506	279527	385187	184
SPSA	3	0.155344	97972	5507	279823	372597	197
Goal Value	-	0.00	84976	4785	268660	100000	-

Figure 3 illustrates the optimal AD subnetworks found by all three heuristics. Several stretches of roads appear in all three results suggesting there are evident efficiency gains in selecting them for AD subnetwork. Although the result from GA seems more plausible, all three suffer from the same flaw, namely, inability to generate connected designs. There are isolated links and stretches within the AD subnetworks produced in all three cases which is undesirable for practical purposes. A possible remedy for this issue can be analyzing the results as a post-processing step and making adjustments on resulting designs to cope with it. Another possibility is penalizing disconnected designs in optimization process. Nonetheless, implementing these solutions can be cumbersome and time-consuming. This signals a need for solution methods that can produce connected designs from the onset. Since DNDP traditionally deals with adding new lanes and links to existing (already connected) networks, connectedness has never been an issue in DNDP. Therefore, solution methods have not been developed to deal with this problem. However, for the case of AD subnetwork, this is a requirement and a priority for the future research.

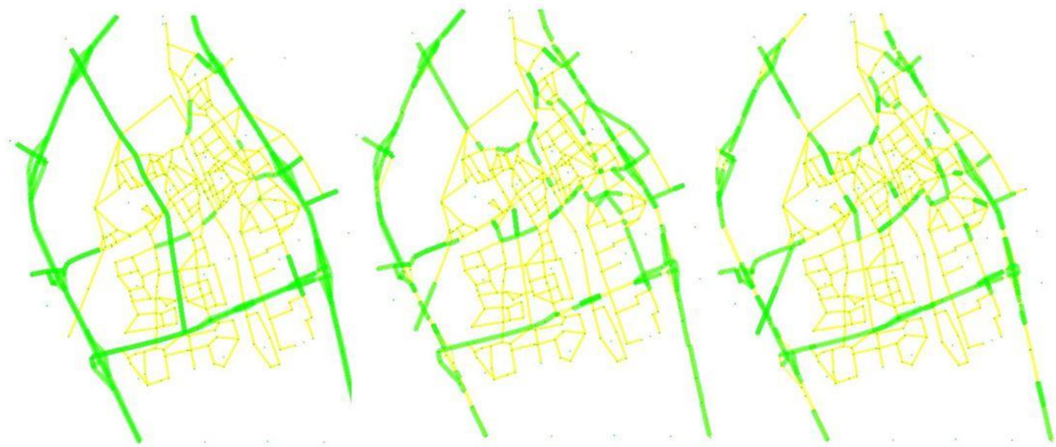

FIGURE 3 Optimal AD Subnetwork for 50% penetration scenario achieved by: (from left, respectively) GA, SAA, SPSA

4. REFERENCES

[1] SAE International, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," 2016.

[2] F. Wegman, L. Aarts, and C. Bax, *Advancing sustainable safety*, vol. 46, no. 2. 2008.

[3] H. Yang and M. Bell, "Models and algorithms for road network design: a review and some new developments," *Transp. Rev.*, vol. 18, no. 3, pp. 257–278, 1998.

[4] B. Madadi, R. van Nes, M. Snelder, and B. van Arem, "Network Design and Impacts of Automated Driving: An Explorative Study," in *Transportation Research Board 97th Annual Meeting*, 2018.

[5] S. Puylaert, M. Snelder, R. van Nes, and B. van Arem, "Mobility impacts of early forms of automated driving - a system dynamic approach," in *Transportation Research Board 96th Annual Meeting 96th Annual Meeting*, 2017.

[6] J. H. Wu, M. Florian, and S. He, "An algorithm for multi-class network equilibrium problem in PCE of trucks: application tothe SCAG travel demand model," *Transportmetrica*, vol. 2, no. 1, pp. 1–9, 2006.

[7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley, 1989.

[8] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science (80-.).*, vol. 220, no. 4598, pp. 671–680, 1983.

[9] R. Z. Farahani, E. Miandoabchi, W. Y. Szeto, and H. Rashidi, "A review of urban transportation network design problems," *Eur. J. Oper. Res.*, vol. 229, no. 2, pp. 281–302, 2013.

[10] J. C. Spall, "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradiant Approximation," *{IEEE} Trans. Autom. Control*, vol. 37, no. 3, pp. 332–341, 1992.