The Braess's Paradox in Dynamic Traffic and Mixed Autonomy

Dingyi Zhuang (dingyi@mit.edu)
Yuzhu Huang (yuzhuh@mit.edu)

Abstract

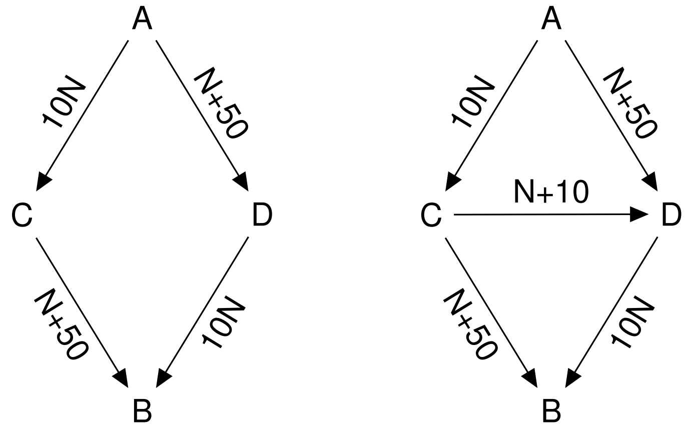
The Braess's Paradox (BP) is the observation that adding one or more roads to the existing road network will increase traffic congestion and slow down the overall traffic flow. In the context of traffic assignment, BP is usually proven by assuming that the travel cost for taking each road is related to the number of vehicles on it and that all vehicles are distributed instantaneously. Such assumption may not be accurately representing the BP phenomenon in real-world traffic scenarios. If BP is examined in a non-instantaneous, dynamic case, can we use mixed autonomy to mitigate the congestion introduced by the shortest path? In this project, we use the Flow project to design and simulate driver behaviors on a rotary road network with and without a shortcut. We find that BP exists within a certain number of vehicles. We try to alleviate the congestion caused by BP by adding autonomous vehicles (AVs) and applying reinforcement learning (RL) algorithm Proximal Policy Optimization (PPO) for control. Results show that the AV learns not to attenuate traffic but blocks the road for better system efficiency. This project is related to the Unit 3 contents about Markov Decision Process and deep reinforcement learning.

I. Introduction & Motivations

Dietrich Braess, a German mathematician, has discovered that adding a new road to an existing road network, although increasing capacity, will actually exacerbate overall congestion. He argued that if each driver is making self-interested routing decisions, the shortest path would be frequently selected, which will lead to impedance on the shortcut and slow down overall traffic speed in the end [1, 2]. A classical approach to formulating this problem usually involves using road cost function related to the number of vehicles on each road to solve for the traffic equilibrium equations and prove the Braess's Paradox (BP) [3].

We articulate how to solve the equilibrium of Braess's Paradox by showing a classic example. As shown in Fig. 1, suppose there are 6 cars that want to travel from point A to point B via routes *ACB* or *ADB*. The travel cost (usually means travel time) associated with each road segment is marked beside the edge, where N is the number of vehicles in the same road segment. Intuitively, that means that if there are more vehicles in the same road, the traffic intends to congest and the travel time is consequently larger. Since the cost function of each route is the same for routes *ACB* and *ADB* if *CD* is not connected, the demand is evenly distributed because using one route is just as good as using the other. The equilibrium reaches when 3 cars go route ACB and the other 3 choose ADB with travel cost $C_{ACB} = 30 + 53 = 83 = C_{ADB}$. Now, suppose a new shortcut *CD* connecting the two routes is constructed. Drivers will shift to the shortcut at which point a new travel equilibrium is reached until $C_{ACDB} = C_{ACB} = C_{ADB}$. The solution is 4 cars are assigned in road *AC* and *DB* and 2 cars in road *AD,CD*, and *CB* with travel cost 92 for each route, larger than the previous case without the shortcut. From this example, we could see that adding a new road to the network could in fact increase the travel time for everybody, which is the Braess's Paradox.

However, one hidden crucial assumption is that vehicles are assigned instantaneously to reach the traffic equilibrium. This static assignment is quite impractical as in real world vehicle



(a) Basic configuration

(b) Basic configuration with added shortcut

Fig. 1: Braess's Paradox in classic traffic assignment.

behaviors are dynamically changing with the surrounding environment rather than instantaneously determined. Moreover, the junction effects and the interactions of human drivers are not fully studied. Therefore, we want to use simulation tool to examine whether BP exists in the dynamic scenario. Apart from that, the deterioration of traffic in BP is caused by congestion and impedance of vehicles attracted to the shortcut. As noticed by Wu et al. [4], Vinitsky et al. [5], mixed autonomy can reduce traffic congestion and smooth traffic flow. Therefore, it would be interesting to explore whether introducing reinforcement learning controller into the mix of vehicles (these can be regarded as the AVs) would help alleviate the congestion in BP. In this project, we use the Flow project [4, 5] as simulation environment to design toy scenarios and prove the existence of BP in the dynamic environment. After that, we train AV with RL controller to resolve the congestion caused by BP and maximize the efficiency of human drivers.

There are some recent papers discussing the impact of AVs in BPp, Mehr and Horowitz [6] discussed the uniqueness of Wardrop Equilibrium with mixed autonomy and Belkina et al. [7] resolved the BP by using the mixed-autonomy and artificial restriction in traffic. These methods are only the continuation of the static traffic assignment and just providing the mathematical proof without fully discussing how AVs can learn strategies to mitigate BP. The contributions of this project are:

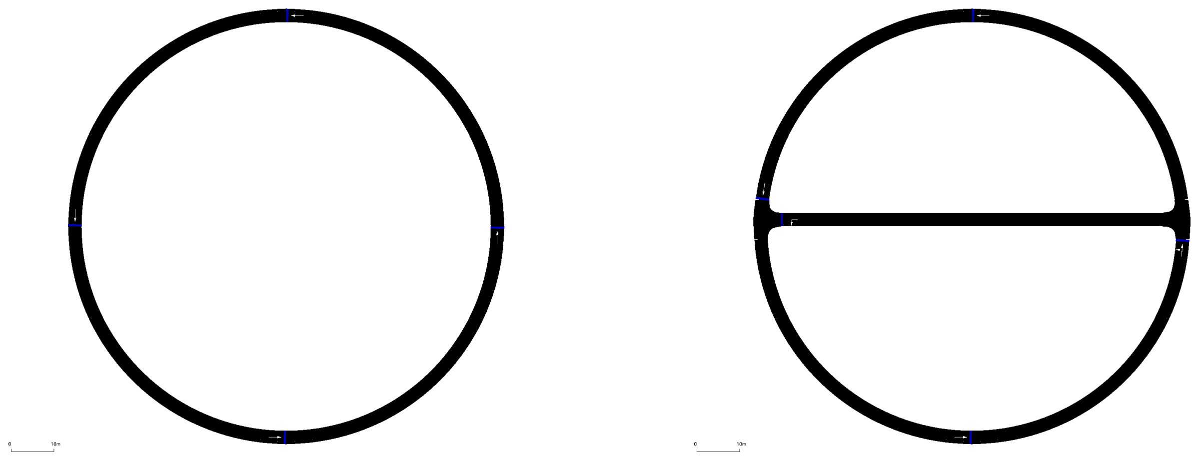
1) We prove that BP also exists in the dynamic scenario but within certain threshold
2) We alleviate congestion caused by BP using mixed autonomy and maximize the travel efficiency of human drivers

As far as we know, we are the first to analyze the Brass's Paradox in the dynamic environment (i.e. simulation tool) and explore the what the RL algorithms can bring about.

II. Experiment Design

A. Scenario Design

Instead of the diamond-shape road network in Fig. 1, a unidirectional rotary is designed as the baseline scenario. We choose the simple ring roundabout so that we can only focus on how the influence of a newly-added shortcut could be. Drivers travel counter-clockwise around the



(a) The basic unidirectional roundabout scenario, the bottom junction is the origin and destination of vehicles.

(b) Adding one road to the roundabout scenario.

Fig. 2: Scenario designs for proving BP. A closed system is used to better capture the change of vehicles.

rotary, passing through 4 stop-signs along the way. The 4 stop-signs are evenly distributed in the bottom, right, top, and left part of Fig. 2a. Then, we add a one horizontal shortcut to the original configuration, thus introducing a shortened path, which is shown in Fig.2b. A complete trip is defined as starting from the bottom node and finishing at the bottom node again. The travel time for a complete trip is determined likewise. The free-flow travel time for traveling outer ring and the shortcut are 64 seconds and 47 seconds respectively.

We try to pursue two research contributions demonstrated in Section I using the closed system of Fig. 2. On one hand, to prove that the BP exists in a dynamic environment, we vary the number of vehicles and look into the difference of average travel time. The reason to focus on the number of vehicles is inspired by Nagurney [8]. They found that BP only appears when vehicle demand of the system is within a certain range and that adding a road does not always deteriorate the system efficiency. On the other hand, we want to discuss how AV and RL controller can help mitigate the congestions created by BP. We formulate it as a Markov Decision Process and train it through RL algorithm. Details can be found in Section II-B and II-C.

B. Markov Decision Process Design

We specify the key elements, including agent, state, action and reward as:

- **State**: The state space consists of position and speed of AV, as well as its leading and following vehicles. Apart from that, headway and tailway of the vehicle and the queue length and speed of the shortcut are also included.
- **Action**: The action space a list of possible acceleration values for each RL vehicle, bounded by the maximum acceleration and deceleration specified by the road network.
- **Reward**: The reward function rewards high average speeds of all human vehicles in the network, and penalizes accelerations by the RL vehicle.
- **Discount**: Discount factor is selected as 0.999.
- **Termination**: An epoch is terminated if the expected simulation timesteps are reached.

There are two other important elements of the environment design that are not in the criteria of MDP, namely the routing policies for both RL vehicles and human drivers and the number of RL vehicles. For routing policies of human drivers, they choose which route to take based on the largest average speed of all available routes. This is imitating drivers using the navigation map in practice. As for the routing control of RL vehicles, we enforce them to choose either the outer rotary road or the new roads to help smooth out the traffic of that route.

C. Methodology

Policy learning algorithms are preferred in our experiment as we directly control the acceleration of the AV. In practice, vanilla policy gradient is not scalable for large scale problems. To solve the problem of hard convergence in vanilla policy gradient, Schulman et. al propose Proximal Policy Optimization (PPO) algorithm [9]. Details can be found in Algorithm 1.

Algorithm 1 PPO

Initialize: policy parameter θ_0, KL penalty β_0, target KL-divergence δ

```
for k=0,1,2,... do
    Sample trajectory s_{i,t}, a_{i,t}, r(s_{i,t}, a_{i,t}), s_{i,t+1} from π_{θ_k}(s|a) in the simulator
    Evaluate the advantage as Â^{π_{θ_k}}(s_{i,t}, a_{i,t}) = r(s_{i,t}, a_{i,t}) + V̂_φ^π(s_{i,t+1}) − V̂_φ^π(s_{i,t})
    Compute policy update θ_{k+1} = arg min_θ J(θ_k) − β_k D̄_{KL}(θ||θ_k) by taking K steps of
    minibatch stochastic gradient descent.
    if D̄_{KL}(θ_{k+1}||θ_k) ≥ 1.5δ then
        β_{k+1} = 2β_k
    else if D̄_{KL}(θ_{k+1}||θ_k) ≤ δ/1.5 then
        β_{k+1} = β_k/2
    end if
end for
```

To implement the PPO, we use RLlib [10] integrated in the Flow project [4]. We build 3-layer fully connected neural network with 32 neurons in each hidden layer. In each iteration, we assign 200 roll-outs and each roll-out contains the continuous simulation with 3000-step horizon. With 8*i9-9820X CPU and 130GB RAM machine, it takes 2 hours to train 5 iterations.

III. Results

A. Braess Paradox in a Dynamic Environment

As introduced in the designing above, we explored the traffic dynamics of both road configurations by varying the number of vehicles in the system.

Firstly, we compared the flow rate and average travel time of each scenario with a small number of vehicles in the system. As shown in Table I, when there are 3 or 5 vehicles in the system, the average travel time using the shortcut configuration is smaller than using the original rotary. This is expected because when the number of vehicles in the system is small, vehicles can travel without much impedance from others. Therefore, vehicles can take advantage of the shortcut and complete their trips faster.

Then, more vehicles are added to the network. When the number of vehicles is increased to 10, we observed that the average travel time of all vehicles is larger in the shortcut configuration than the original rotary. As shown in Fig. 3, because of the fact that vehicles using the shortcut are able to complete their trips faster, vehicles congregate at the bottom node and start forming a queue. In addition, vehicles will encounter others at the merging junction, so the time spent yielding to others at the stop sign also contributed to the increased travel time. As a result, the average travel time using the added shortcut ends up becoming larger than using the original ring rotary, evidently showing the phenomenon that is stated in BP.

The appearance of BP persists and becomes more obvious visually when there are 25 vehicles in the system. As shown in Figure 5, the queue formed near the bottom node becomes longer and extends into the shortcut and the top arc of the rotary. The delay from this queue offsets any travel time savings that vehicles gained from traveling on the shortcut, so the resultant average travel time shows no improvement from using the original rotary.

The Paradox continues to show up in the comparison, until the number of vehicles is above a certain number, e.g. 30. With 35 vehicles in the system, the original rotary is almost reaching its capacity, so vehicles in the network experience stop-and-go traffic condition, and the network is close to a total breakdown. In comparison, even though the shortcut configuration is congested, it is still far from reaching total breakdown credit to the additional capacity offered by the added shortcut. Therefore vehicles can be processed faster in the shortcut configuration, and it exhibits a better performance than the original rotary. The additional capacity and more advantageous performance of the shortcut configuration are also shown on the flow-density curves in Fig. 6, where we can clearly see that the congestion portion of the flow-density curve appears much earlier than in the shortcut configuration.

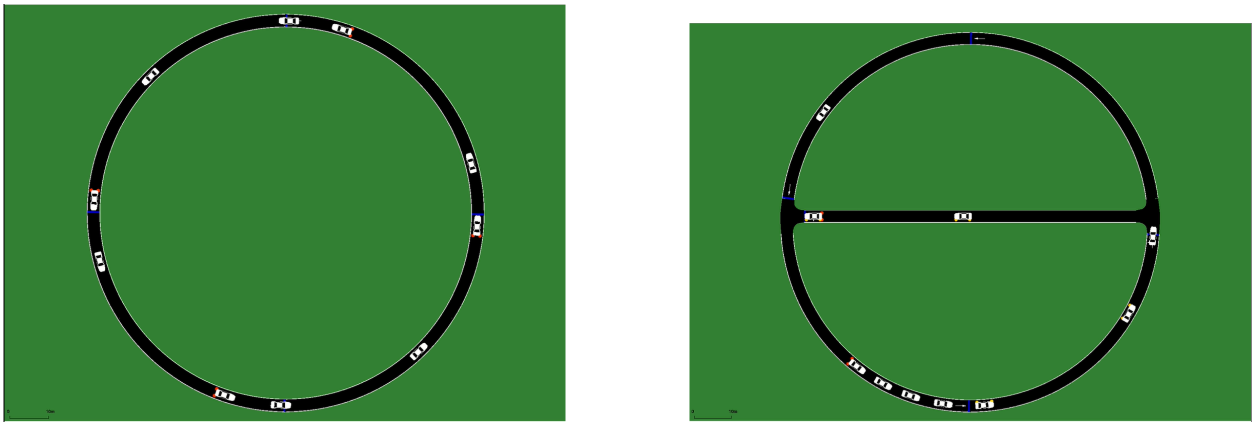
	Rotary		Rotary + Shortcut		Difference	
# of Vehicles	Flow (veh/min)	Travel Time (sec)	Flow (veh/min)	Travel Time (sec)	Flow (veh/min)	Travel Time (sec)
3	2.8	64	3.2	53.8	0.5	-10.2
5	4.6	64.4	5.3	55.2	0.7	-9.3
10	8	72.4	7.8	76	-0.2	3.6
11	7.8	82.6	7.8	83.6	0	1.1
15	7.8	114	7.8	114	0	0
20	7.8	152	7.8	150.3	0	-1.7
25	7.8	190	7.8	190	0	0
30	7.7	231	7.8	227.8	0.1	-3.2
35	6.8	294.9	7.8	265.2	1	-29.6
37	-	-	7.8	279.6	-	-
40	-	-	7.7	307.4	-	-
41	-	-	7.6	312.3	-	-
42	-	-	7	354.9	-	-

TABLE I: Travel Time Comparison

B. Impact of Penetration level of Traveler Information System

The traffic operations of the shortcut configuration was simulated using an assumed driver routing controller, where drivers choose which route to take–top arc or the shortcut–based on the average speed. Specifically, when drivers travel to just upstream of the diverging point (the right node), they would compare the average speed of all vehicles on the top arc with that of the shortcut (if there is no vehicle, on the path, then a maximum speed is used).

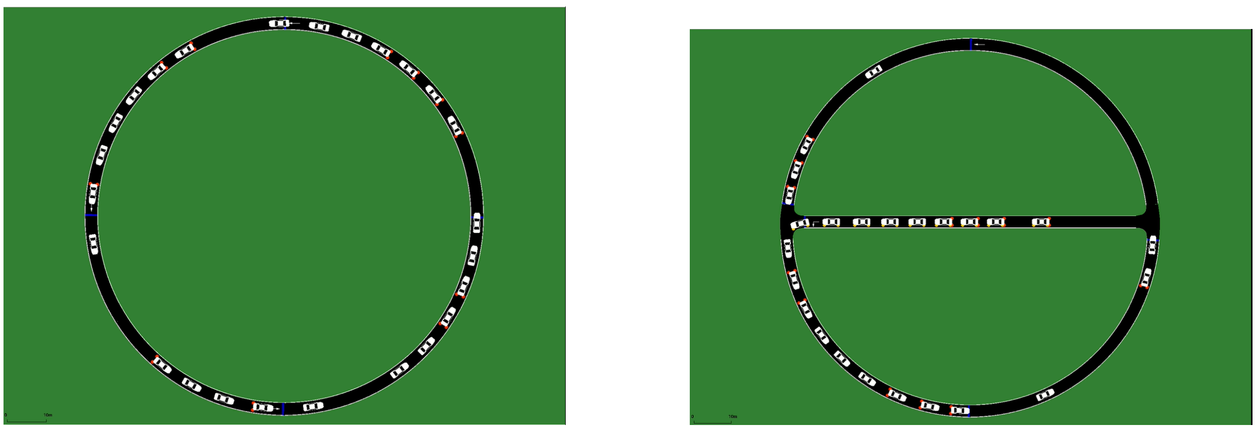
The simulation of the routing controller described above is equivalent to the real-world situation where all drivers are equipped with a traveler information system, for examples, navigation apps that offer real-time traffic information of all routes. However, it could be unrealistic to assume



(a) 10 vehicles using original rotary.

(b) 10 vehicles using rotary with shortcut.

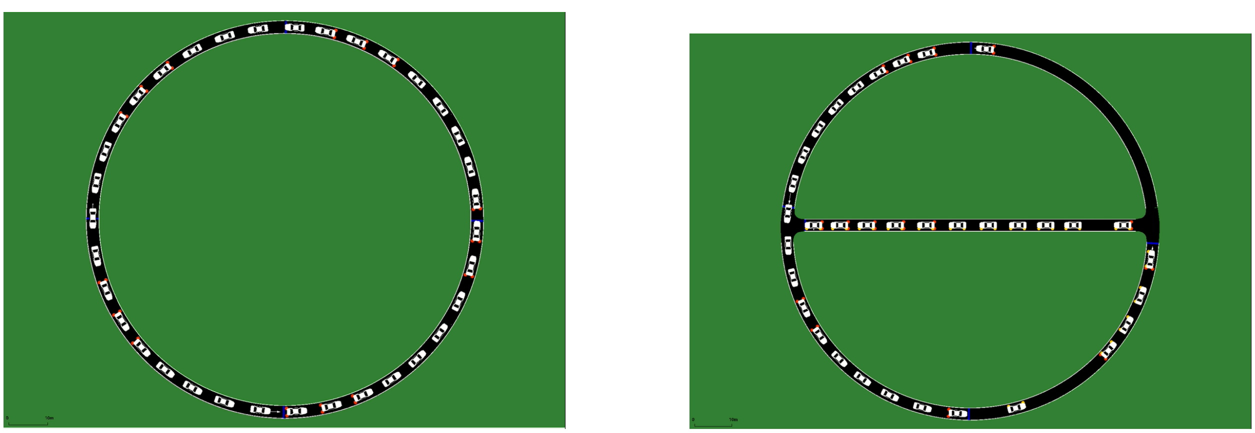
Fig. 3: Scenario: 10 vehicles in rotary vs. rotary + shortcut



(a) 25 vehicles using original rotary.

(b) 25 vehicles using rotary with shortcut.

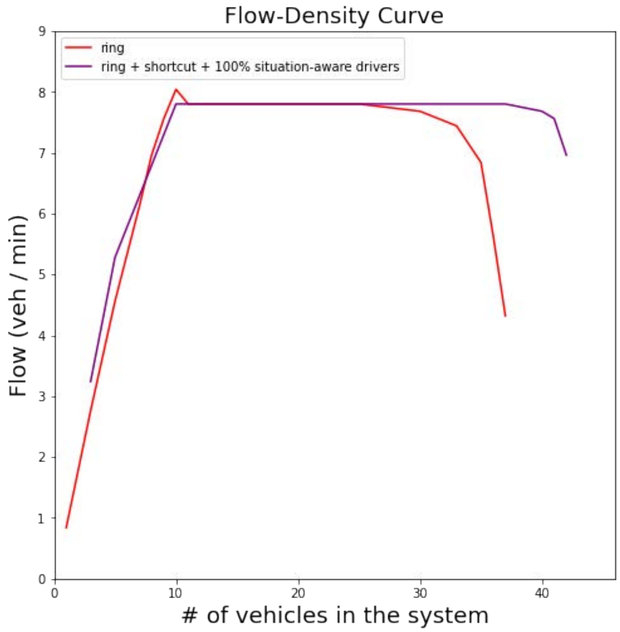
Fig. 4: Scenario: 25 vehicles in rotary vs. rotary + shortcut



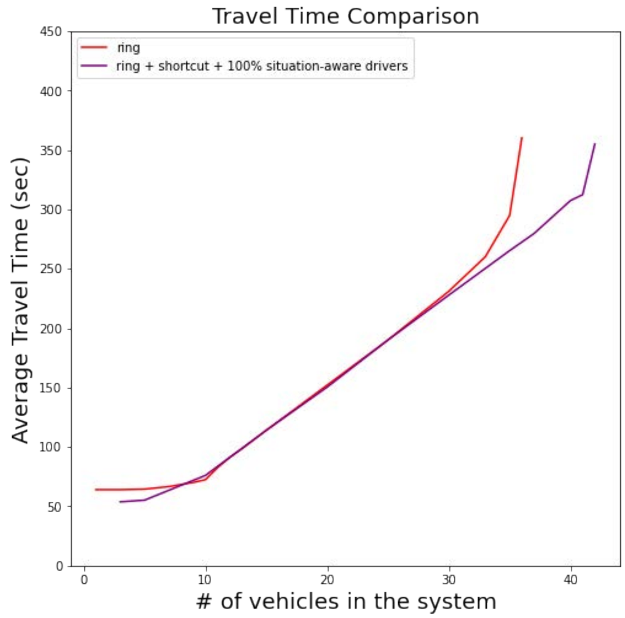
(a) 35 vehicles using original rotary.

(b) 35 vehicles using rotary with shortcut.

Fig. 5: Scenario: 35 vehicles in rotary vs. rotary + shortcut



(a) Flow-density curves.



(b) Average vehicle travel time.

Fig. 6: Comparison of traffic performance using the original rotary and added shortcut

that the penetration level of navigation apps is 100%. Therefore, we evaluated the performance of both scenarios assuming varying levels of penetration of navigation apps.

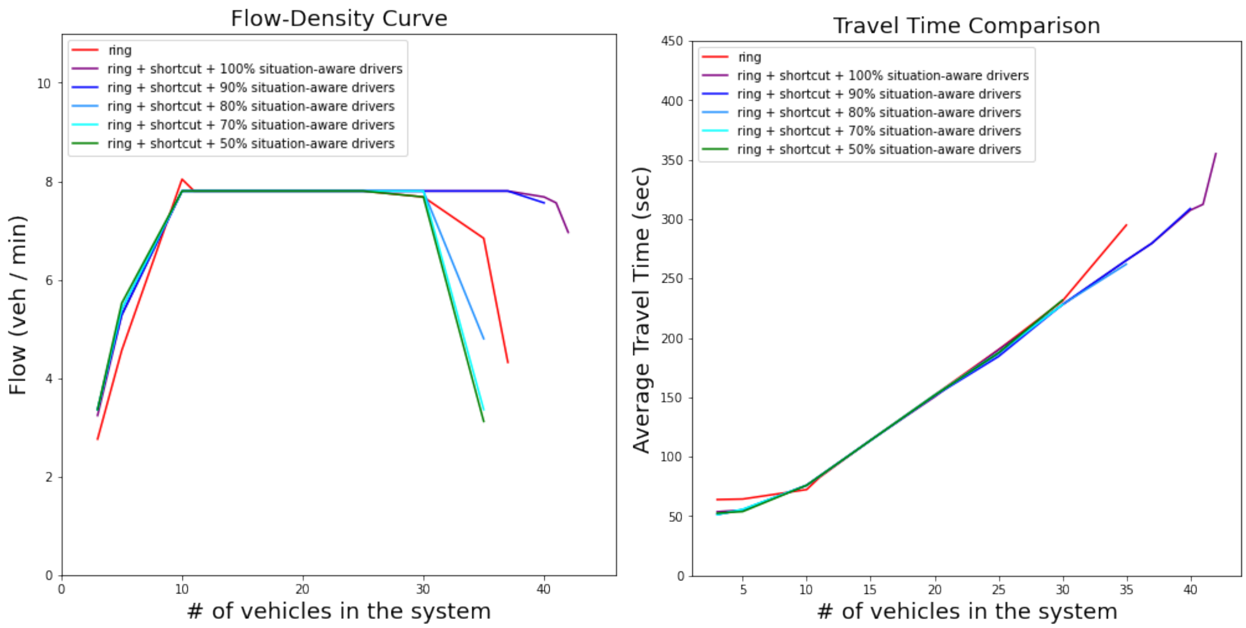
As shown in Figure 7b, we can see that the Paradox exists however much the penetration level is. However, in Figure 7b, the experiments show that the capacity of the network drops significantly when the penetration level drops from 90% to 80%. This result is aligned with the intuition that when penetration level lowers, more vehicles default to using the shortcut without knowing that the other route may be a better option, therefore decreasing the capacity of the network.

C. *Effects of Autonomous Vehicles in the Braess Paradox*

We conducted multiple experiments by introducing autonomous vehicle (AV) - the reinforcement learning agent to the road network. In each experiment, we specified that the AV would only follow one path - either using the original rotary or following the shortcut.

In each experiment, the vehicle composition consists of 10 human-driven vehicles and 1 AV, making for a total of 11 vehicles in the system. The benchmark number 11 was used because, as we have shown in Table I, the deterioration of traffic performance attributed to the Braess Paradox is most obvious when the number of vehicles in the system is around 10 or 11.

In the first experiment, the AV is configured to use the original rotary only. As a reminder, the immediate reward is defined as the average speed of all human-driven vehicles with a small penalty given for acceleration to discourage shock wave formation. In the initial iterations, the training result showed that the AV explored using wave attenuation to encourage fewer stop-and-go events as the AV exhibit the behavior of traveling very slowly on the bottom arc before reaching the diverging point. At Iteration 10, the AV slightly rolled back the aggressive attenuation strategy, and instead traveled faster on the bottom arc, though still not accelerating fully to avoid excessive



(a) Flow-density curves.

(b) Average vehicle travel time.

Fig. 7: Comparison of different penetration levels of navigation apps

acceleration. At Iteration 20, the AV started exhibiting the behavior of traveling very slowly on the top arc after the diverging point. This behavior forced all human-driven vehicles to use the shortcut instead, because every driver would observe that the average speed of the top arc is much slower than that of the shortcut. As a result, drivers experience a lower control delay at the stop sign at the merging junction while still being able to take advantage of the savings from the shortcut, thus experiences a lower travel time.

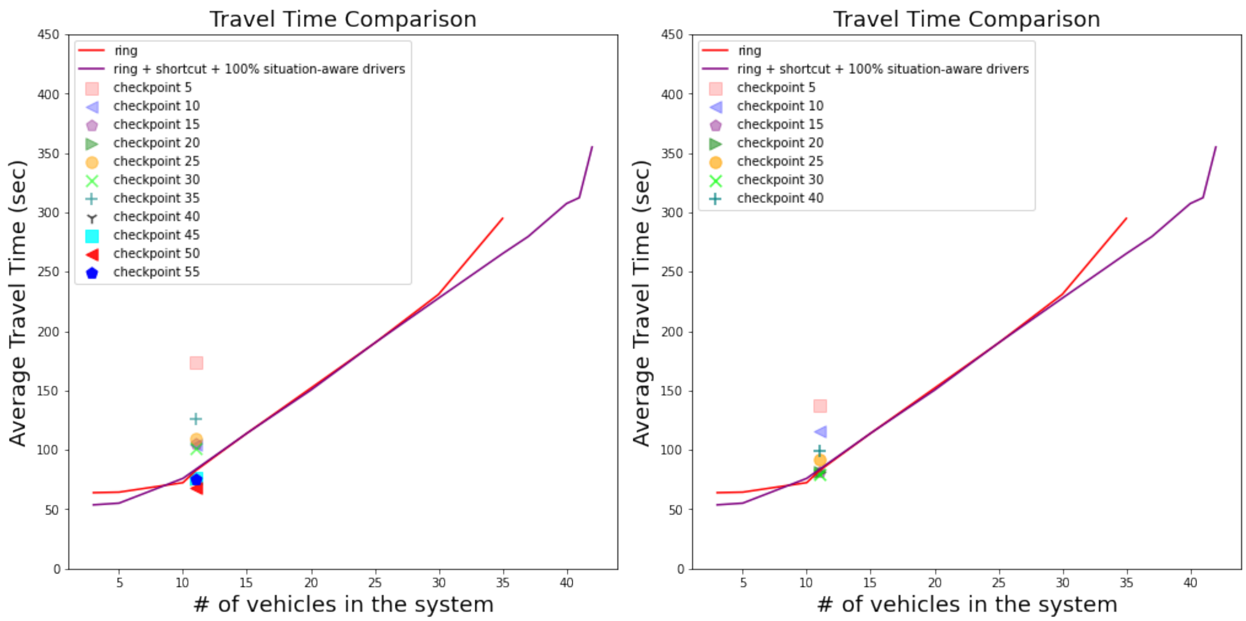
Similarly, when the AV is designated to only use the shortcut, the AV learned to stop on the shortcut at around Iteration 30, forcing all human-driven vehicles to only use the outer rotary. This strategy again avoided the need of vehicles to yield to others at the merging junction, thus resulting in a lower travel time. In Iteration 45, the AV has not only learned the blocking strategy, but is able to identify when human-driven vehicles do happen to choose the shortcut that is supposed to be blocked. When that happens, the AV would accelerate to clear out the road for these human-driven vehicles, until the next cycle when it would come back to the shortcut and block it again.

The progress in travel time improvements of different iterations of both experiments are shown in Figure 8.

It's worth noting that because our AV was trained while only following a predefined route, the optimal strategy that it had learned in either experiment would only be a local optimum rather than system optimum. This can be found from the training results from Fig. 9.

IV. Relation with mathematical traffic assignment

The BP in traffic assignment usually needs a cost function that influence the final traffic equilibrium. The cost function is positively related to the number of cars in that road. However, when a new road is added, the cost function remains unchanged for the original road, like the



(a) RL Scenario: AV only uses rotary.

(b) RL Scenario: AV only uses the shortcut.

Fig. 8: Travel time results of different checkpoints in RL experiments

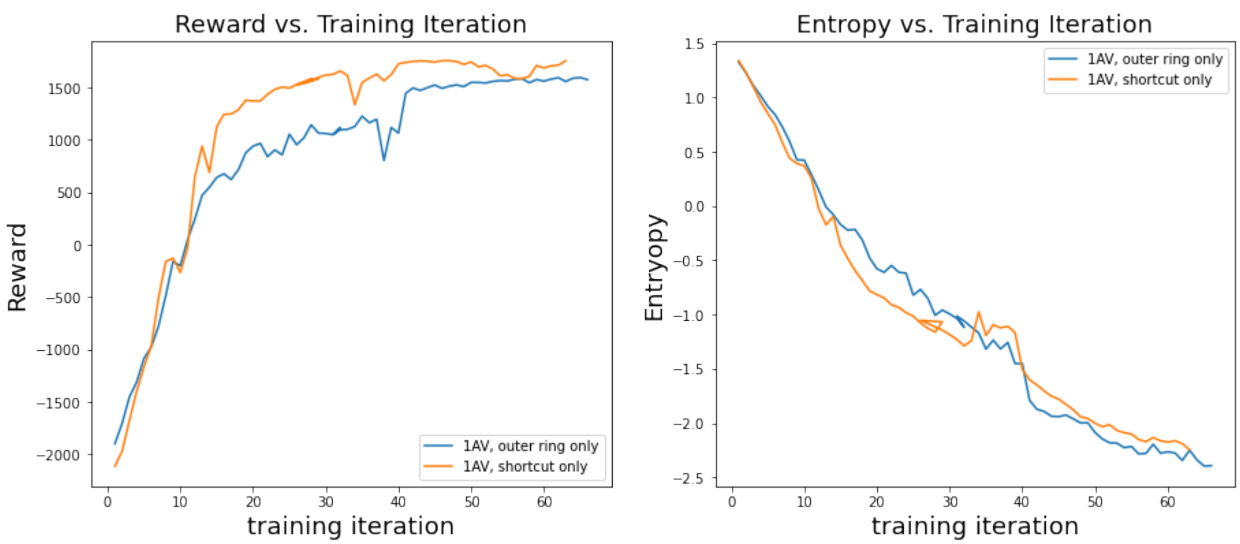


Fig. 9: Reward and Entropy of RL Experiments

case shown in Fig. 1. That means it does not take the effect of junction into consideration. In the simulation environment, vehicles in the merging point of the junction will propagate the stop-and-go wave and cause the queue in part of the road network, which leads to the increase of travel time. Therefore, when the shortcut is added, cost functions for each road also need to change and incorporate the vehicle interactions in the junction, which is intuitively a non-linear relation with the number of vehicles in the road segment. On the other hand, the fact that AV learns to block the road tells us that traffic operation managers might need to guide drivers to other roads when BP appears.

V. Conclusion

In this paper, we have designed an simple rotary road network with and without a shortcut to discuss the BP in the dynamic environment. Different from existing research about BP that assigns the traffic instantaneously, we use the Flow Project to simulate the driver behaviors in the dynamic setting. We find that BP exists within certain number of vehicles, typically 10 or 11 vehicles in our case. After finding the BP, we add an AV and apply RL algorithm PPO to let it optimize the system efficiency under the BP. The AV learns that wave-attenuation is not the optimal solutions but to block one of the roads. This is an inspiring result that we can suggest traffic operation manager to guide drivers when similar congestion happens.

Our project is far from perfect. In the future, we need to retrain with multiple seeds and multiple AVs. We can even discuss how to let AV chooses which route to take itself and extend our work to other complex road network.

VI. Acknowledgement

We want to thank the guidance and support of Prof. Cathy Wu and teaching assistants Vindula Jayawardana and Dajiang Suo. We can not finish this project without your help and suggestions!

References

[1] R. Steinberg and W. I. Zangwill, "The prevalence of braess' paradox," *Transportation Science*, vol. 17, no. 3, pp. 301–318, 1983.

[2] M. Frank, "The braess paradox," *Mathematical Programming*, vol. 20, no. 1, pp. 283–302, 1981.

[3] A. Nagurney and L. S. Nagurney, "The braess paradox," *International Encyclopedia of Transportation. Elsevier, forthcoming. Invited chapter, pre-print available at https://supernet. isenberg. umass. edu/articles/braess-encyc. pdf*, 2021.

[4] C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: Architecture and benchmarking for reinforcement learning in traffic control," *arXiv preprint arXiv:1710.05465*, vol. 10, 2017.

[5] E. Vinitsky, A. Kreidieh, L. Le Flem, N. Kheterpal, K. Jang, C. Wu, F. Wu, R. Liaw, E. Liang, and A. M. Bayen, "Benchmarks for reinforcement learning in mixed-autonomy traffic," in *Conference on robot learning*. PMLR, 2018, pp. 399–409.

[6] N. Mehr and R. Horowitz, "How will the presence of autonomous vehicles affect the equilibrium state of traffic networks?" *IEEE Transactions on Control of Network Systems*, vol. 7, no. 1, pp. 96–105, 2019.

[7] E. A. Belkina, E. A. Zhestov, and A. V. Shestakov, "Methods for resolving the braess paradox in the presence of autonomous vehicles," , vol. 13, no. 2, pp. 281–294, 2021.

[8] A. Nagurney, "The negation of the braess paradox as demand increases: The wisdom of crowds in transportation networks," *EPL (Europhysics Letters)*, vol. 91, no. 4, p. 48002, 2010.

[9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[10] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, J. Gonzalez, K. Goldberg, and I. Stoica, "Ray rllib: A composable and scalable reinforcement learning library," *arXiv preprint arXiv:1712.09381*, 2017.