Detecting Edges Causing Braess's Paradox via Simulation

Toshiki Kojima[1], Ahmad Bazzi[1], and Yoshikuni Onozato[1]

[1]Graduate School of Engineering, Gunma University, Japan

1 Introduction

Braess's paradox is a situation that might occur when new paths are added to a network. One might think that adding additional paths would improve transportation time; however, Braess's paradox shows that the transportation speed can be degraded by adding certain paths because the drivers follow a game theoretic strategy without coordinating between each other.

A brief introduction regarding how network traffic is affected by game theoretic decisions and the occurrence of Braess's paradox can be found in reference [1]. A proof of Braess's Paradox is shown in reference [2], while several guidelines on how to avoid Braess's Paradox is presented in reference [3]. Braess's ratio with respect to the maximum experienced latency of a flow particle is introduced in reference [4]. They discuss how Braess's paradox instance can be found if the Braess's ratio is strictly greater than one.

In this paper we propose a method to find the edge causing Braess's paradox, if any, in order to eliminate it. The proposed method is simulated and the results of the simulation are provided. The proposed technique can be applied to larger networks in order to improve the transportation time by eliminating the edges causing Braess's Paradox.

2 Eliminating Braess's Paradox

In this section, we present our proposed method to optimize a traffic pattern. The steps are as follows:

1. First, check whether Braess's paradox is occurring.
2. If it is occurring, create new networks by deleting one of the edges used in the traffic path.
3. Find the Nash equilibriums and the average travel time of each network.
4. Compare the time cost of each of the created networks with the time of the original network, and adopt the least expensive one.

For example, consider the routes shown in Fig. 1 and suppose that 4000 cars want to get from city 1 to city 4. Each edge is labeled by the time cost it incurs the travelers. The edge between vertices 2 and 3 has zero cost. The Nash equilibrium of this network is established when all the cars use the route through 2 and 3 (the path $1 \to 2 \to 3 \to 4$) resulting in a travel time of 80 minutes [1].

We seek to study the result of eliminating the different edges used in the used traffic path; hence we eliminate $1 \to 2$, $2 \to 3$ and $3 \to 4$, one edge at a time. We have the following scenarios:

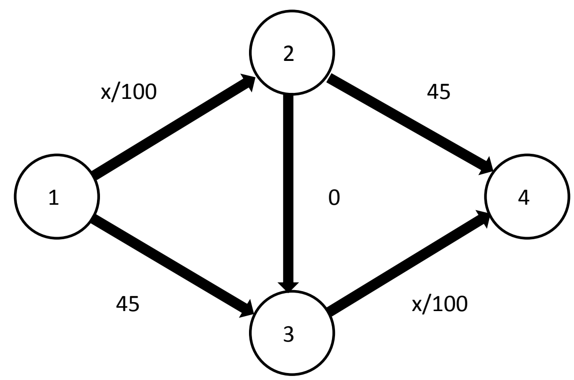

Fig. 1. A network suffering from Braess's paradox

- When the city government eliminates edge 1–2 the traffic pattern in the case of Nash equilibrium of the new the network is when the cars use the lower route through 3, then the total travel time of each driver becomes 85 minutes, since $4000/100 + 45 = 85$.
- When the city government eliminates edge 2–3 the traffic pattern in the case of Nash equilibrium is when all the drivers balance themselves evenly between the two routes, then the total travel time of each driver becomes 65 minutes, since $2000/100 + 45 = 65$.
- When the city government eliminates edge 3–4 the traffic pattern in the case of Nash equilibrium is when all the cars use the upper route through 2, then the total travel time for everyone is 85 minutes, since $4000/100 + 45 = 85$.

Next, we compare the time cost of the new cases with the time of the original network to find the one with the minimum time. In this case we adopt the traffic pattern with edge 2–3 eliminated as it allows the least cost.

Since the occurrence of Braess's paradox is caused by adding new routes to a network and consequently allowing new strategies, we can avoid it by using the proposed method. This is achievable because the edge that causes Braess's paradox is always used in the traffic route.

In this example, we use directed graph, but in the simulation, we use undirected graph as shown in Fig. 2. The main difference between directed and undirected graphs is that the driver can move from 3 to 2.

3 Simulation

In this simulation, we take the model shown in Fig. 2. In this model, the edges 1–3 and 2–4 are insensitive to congestion as they have a fixed cost. However, the edges 1–2 and 3–4 are sensitive to congestion, as they have a variable cost directly proportional to the number of cars. When Braess's paradox occurs, we compare the percentage of reduction of delay with each change in the network.

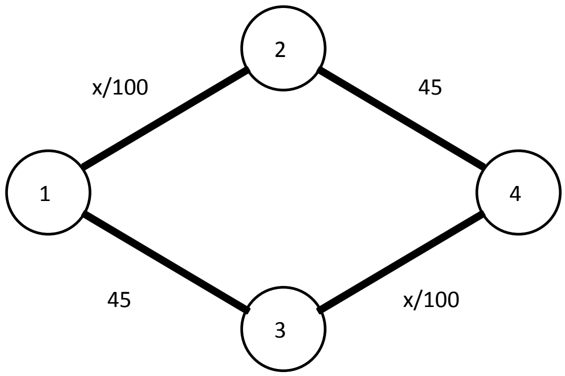

Fig. 2. The used simulation model

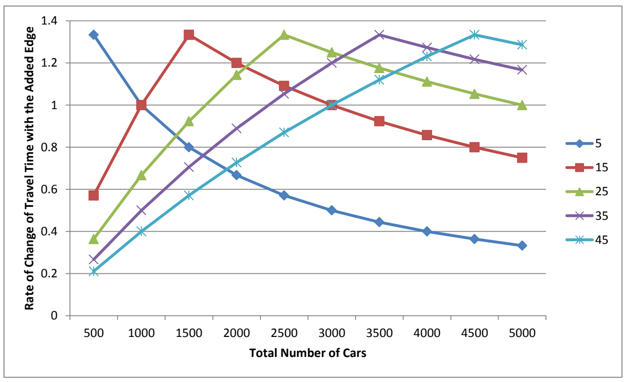

Fig. 3. Ratio of delay before applying our proposed method

It is important to note that the edges with a fixed cost are independent of the number of passing cars; therefore, the cost of using them is constant regardless of the number of cars. Cars using edges with a variable cost —x/100 in this case— will suffer a delay proportional to the number of the cars using the same edge.

The simulator is written in C++ and it follows the assumptions below:

- The model uses undirected graphs.
- Each driver will choose a strategy to minimize his cost.
- Each car has information regarding the route chosen by other cars.

4 Results

Fig. 3 shows the simulation results before applying our proposed method. The vertical axis indicates the ratio of delay while the horizontal axis indicates the number cars. In particular, when the value indicated on the vertical axis surpasses 1, this means that Braess's paradox is occurring.

Fig. 4 shows the simulation results after applying our proposed method. As a result, the value indicated by the vertical does not go above 1. In other words, Braess's paradox has been avoided.

Fig. 5 shows the reduction in the average travel time after applying the proposed method. The vertical axis is a value obtained by travel time when Braess's paradox is avoided divided by the travel time when the paradox is occurring. The horizontal axis indicates the number of cars. It is clear that average travel time is reduced by a maximum of 25%.

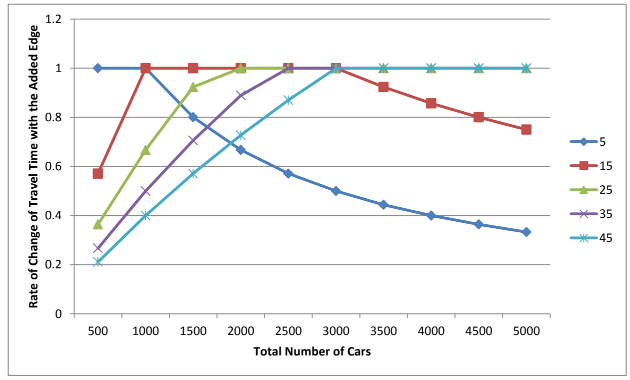

Fig. 4. Ratio of delay after applying our proposed method

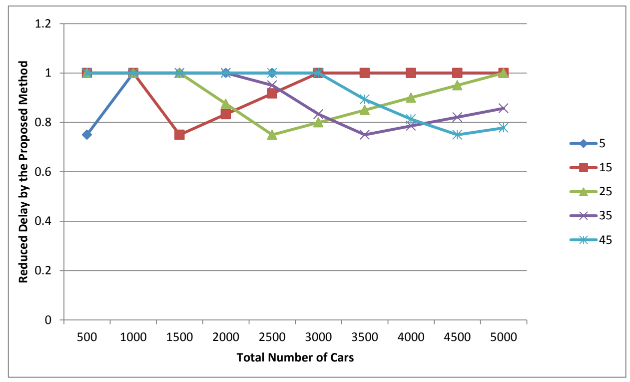

Fig. 5. Rate of decrease in the incurred delay

5 Conclusion

In this paper, we propose a method to avoid Braess's paradox and prove its efficiency through simulation. This method can be used to find the road that should be closed in order to avoid congestion in the network.

References

[1] Easley, D., & Kleinberg, J., Modeling Network Traffic using Game Theory, Networks, Crowds, and Markets: Reasoning about a Highly Connected World, 229–247, 2010

[2] Valiant, G., & Roughgarden, T., Braess's paradox in large random graphs, Proceedings of the 7th ACM conference on Electronic commerce, New York, USA, 296–305, 2006

[3] Altman, E., El Azouzi, R., & Pourtallier, O., Avoiding paradoxes in routing games, Proceedings of the 17th International Teletraffic Congress, 24–28, 2001

[4] Macko, M., Larson, K., & Steskal, ., Braess's Paradox for Flows over Time, Algorithmic Game Theory, Springer Berlin / Heidelberg, 262–275, 2010