École Polytechnique Fédérale de Lausanne
Semester project
Master of Civil Engineering, Autumn 2024-2025

Identification of Braess links in traffic networks

Julien Ars

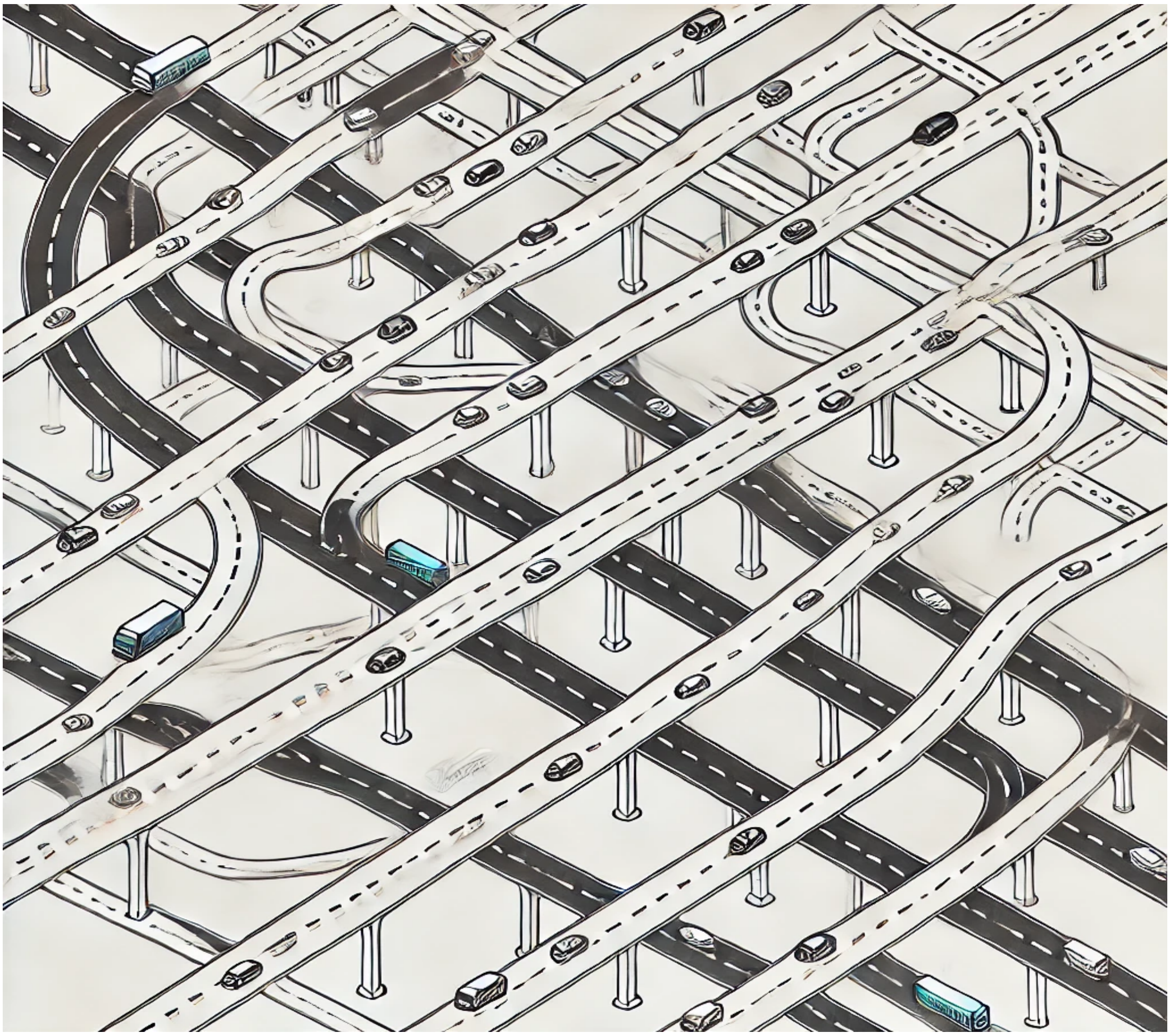


Figure 1: Artistic Intelligence rendering of the concept of Braess' paradox

Introduction

In traffic research, Braess' Paradox is a counterintuitive phenomenon where the addition of a new road can lead to an increase in overall travel time, rather than an expected improvement, due to the self-optimizing behavior of individual drivers. This project delves into the identification of such paradoxical links by closely analyzing traffic flows under two key conditions: User Equilibrium (UE) and System Optimum (SO). In User Equilibrium, the system is solved under the condition that drivers try to minimise their own traject time, while at System Optimum, the total traject time is minimised. These slightly different definitions yield different traffic repartition, and one could formulate an hypothesis that these differences in flow between both condition could highlight the presence of what is called Braess links. The primary objective of this research will be to verify this hypothesis, that is whether certain links, which are used under User Equilibrium but not under System Optimum by the flow for a given Origin - Destination pair, could contribute to improved travel efficiency if restricted to this OD pair.

In order to do so, we will solve the system both at User Equilibrium and at System Optimum, using a classical Frank-Wolf optimisation algorithm. In order to get OD-based links flows, we will apply the principle of Entropy Maximisation, that is that we will try to get the description of user's trajectories that yield the maximum entropy. For this, we will apply Entropy Maximisation through Alternative Representation of Bushes (EMARB) [1]. Then, we will compare OD based flow at User Equilibrium and System Optimum, and identify links that satisfy the condition of our hypothesis. Finally, we will solve the system at User Equilibrium while banning those links and compare the total travel time. An initial research will then be done to try to identify explaining factors for the results, altough further exploration of the results would probably still be necessary.

Understanding the dynamics behind Braess' Paradox is critical for optimizing urban mobility, traffic management, and infrastructure planning. By investigating these effects on a controlled network model, this study aims to provide insight into how rerouting strategies or infrastructure modifications could enhance overall travel times and alleviate congestion.

Contents

1 Objective

The central question guiding this research is whether Braess links can be systematically identified within a traffic network by comparing traffic flow distributions under User Equilibrium (UE) and System Optimum (SO) conditions. In particular, it investigates if a link used in UE but not in SO would improve total travel time if its access were restricted for a given Origin-Destination (OD) pair:

$$x_{ij}^{UE,\ od} > 0 \text{ and } x_{ij}^{SO,\ od} = 0 \Rightarrow \text{Link (i, j) is a Braess link for OD (o, d)}$$

Where $x_{ij}^{UE/SO,\ od}$ represent the flow on the link from node i to node j, that is caused by the trafic going from origin o to destination d, at User Equilibrium, respectively at System Optimum.

By addressing these questions, the study seeks to establish a methodological framework for recognizing inefficient links and assessing the benefits of their removal.

2 Methodology

The study follows a structured methodological approach. First, the network is solved under UE and SO conditions using the Frank-Wolf optimisation algorithm to determine link traffic distributions. Next, OD-based flows are extracted by applying Entropy Maximization through Alternative Representation of Bushes (EMARB). Following this, UE and SO link flows are compared for each OD to pinpoint links suspected of exhibiting Braess' Paradox characteristics. Lastly, identified Braess links are banned for the relevant OD pairs, and overall travel times are compared to determine the net impact on network efficiency.

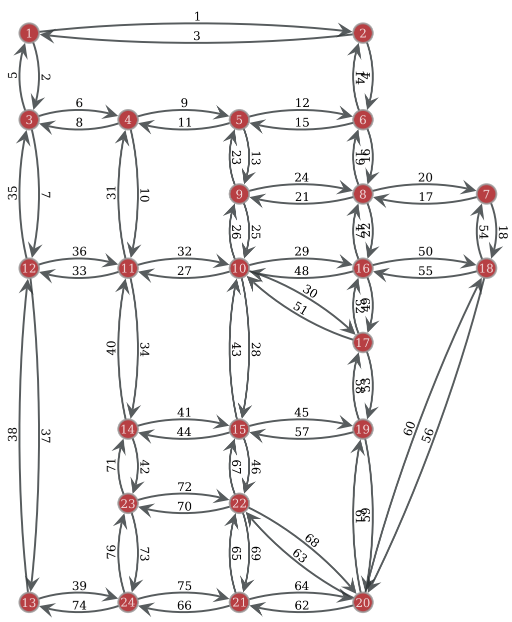


Figure 2: The Sioux Falls network, with node and link labels

2.1 The study network

The study is conducted on the Sioux Falls network (Figure 2) [2], a commonly used model in transportation studies. This network consists of 24 nodes and 76 directed links. The limited scale of this network allows for the evaluation of our hypothesis without needing excessive computational power.

2.2 The demand

The demand on the network is implemented from [2]. The values range between 100 and 5000, as seen in Figure 3

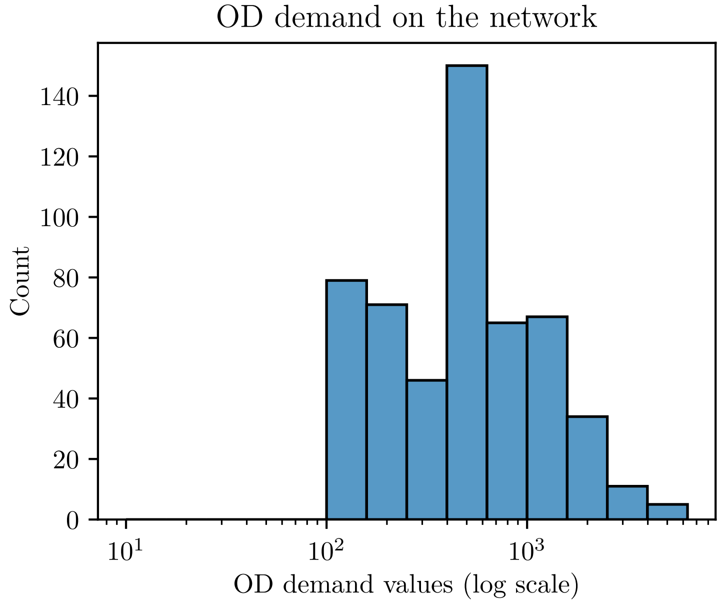


Figure 3: histogram of the values of the demand, by OD

2.3 Implementation

The different algorithms have been implemented in Python, using the graph-tool library for networks manipulation [3].

2.4 Solving the network

The code for the following process can be seen in appendix A.2.

In order to solve the network for both UE and SO, we use a classical Frank-Wolf algorithm. The travel time is estimated using the classical Bureau of Public Roads (BTR) cost function :

$$t_a(x_a) = t_a^0 * \left(1 + \alpha * \left(\frac{x_a}{c_a}\right)^\beta\right) \tag{1}$$

Where :

> $t_a(x_a)$ is the travel time for link a with flow x_a

> t_a^0 is the free flow travel time for link a

> α is a parameter ($\alpha = 0.15$ [2])

> β is a parameter ($\beta = 4$ [2])

> x_a is the capacity of link a

At UE, we run FrankWolf to obtain the flows $\boldsymbol{x}^{UE}$ that minimise the following function : [4]

$$z(\boldsymbol{x}^{UE}) = \sum_a \int_0^{x_a} t_a(\omega)\mathrm{d}\omega \tag{2}$$

At SO, we instead replace the travel time with the marginal travel time $\tilde{t}_a = \frac{\mathrm{d}t_a}{\mathrm{d}x_a}$, such that :

$$\tilde{z}(\boldsymbol{x}^{SO}) = \sum_a \int_0^{x_a^{SO}} \tilde{t}_a(\omega)\mathrm{d}\omega = \sum_a t_a(x_a^{SO}) \tag{3}$$

Convergence criterion We implemented as convergence criterion the relative change in travel time, that is we stop the algorithm when :

$$\left| \frac{t(\boldsymbol{x}^n) - t(\boldsymbol{x}^{direction})}{t(\boldsymbol{x}^n)} \right| \leq \epsilon \tag{4}$$

Where :

> $t(\boldsymbol{x}) = \sum_a t_a(x_a)$ at UE and $t(\boldsymbol{x}) = \sum_a \tilde{t}_a(x_a)$ at SO

> $x^{direction}$ is the result of the direction search step (and therefore is an upper bound of x^{n+1}).

Determining ϵ is a dilema of good results precision versus the computational time needed for high precision. Considering the computational time but also the relative error obtained, with the relative error defined as $\text{error}_{\text{rel}} = \frac{x_a - x_a^0}{x_a^0}$, with x_a^0 being the best known solution according to [2], we choose to use $\epsilon = 10^{-6}$ for the initial UE and SO computation, and then $\epsilon = 10^{-4}$ for the multiple travel time comparison needed to verify the hypothesis. Figure 5 shows the error distributions for those two precision.

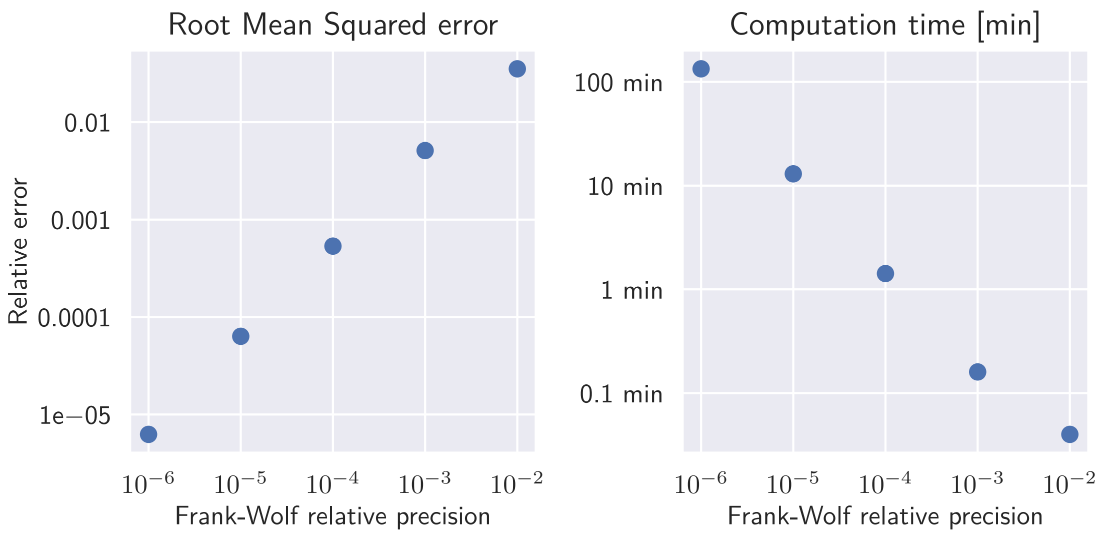


Figure 4: Frank Wolf precision comparison (UE)

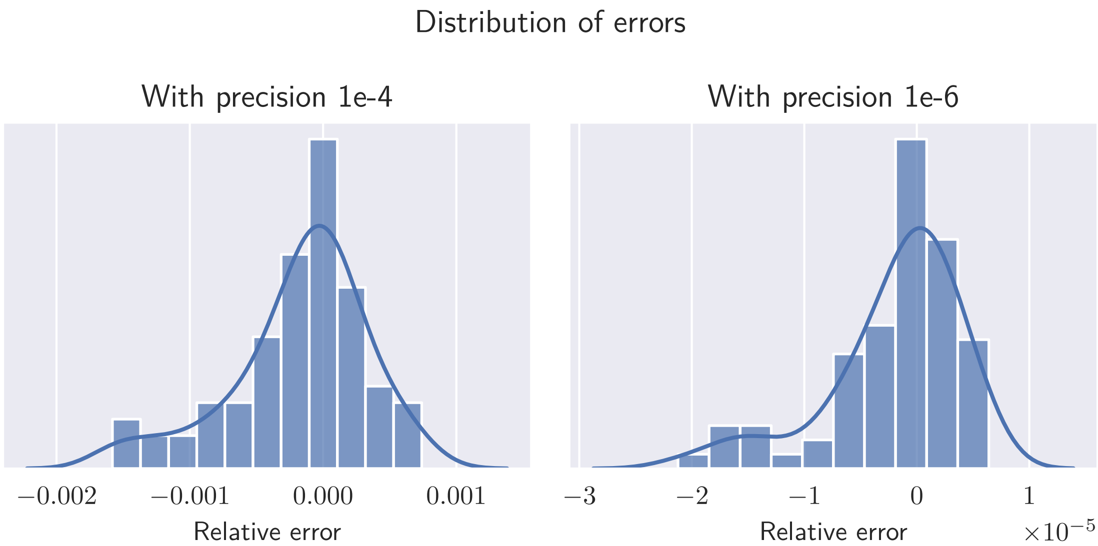


Figure 5: Frank Wolf relative errors distributions (UE)

2.5 OD based flows

In order to obtain the OD based flows, we apply the Entropy Maximization through Alternative Representation of Bushes (EMARB) algorithm [1]. The main idea of the algorithm is to iterate through the nodes and for each node to equilibrate the flows going to this node between the different origins. This gives us origin-based bushes, from which, for each OD, paths are computed and aggregated into OD-based link flows.

Validation of the implementation In order to validate our implementation of the algorithm, we ran it over the example in [1]. The reader is redirected to appendix A.5 for the process.

Convergence criterion The EMARB algorithm's convergence criterion is expressed in absolute values: every time the algorithm run around a node, it remembers the sum of the absolute changes in every origin-based link flow, and it stops once this change is smaller than a parameter ϵ for every node. We have, in the end, used an $\epsilon = 0.001$, which is the smallest we ran in an understandable computational time. The differences in origin-based link flows are shown in Figure 6.

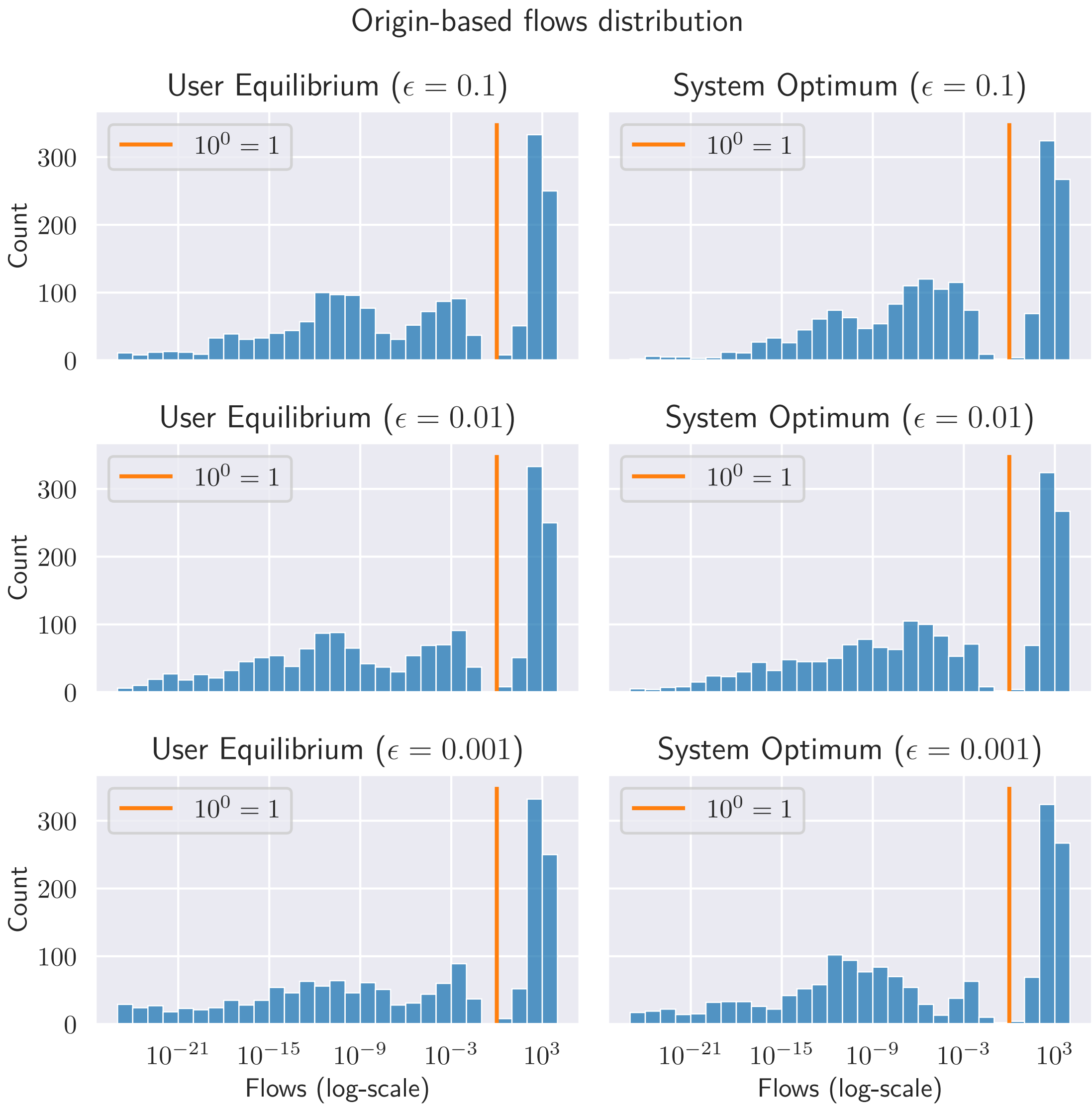


Figure 6: Impact of the choice of ϵ on the orders of magnitude of origin-based link flows

One can see that the effect of a lower ϵ on the scale of the flows is important on the values smaller than 1. However, keeping the lower values would not make much sense. As such, we will consider any value inferior to $\epsilon = 0.001$ to be zero.

In order to get the paths and link flows for each OD, we implement a depth-first search method to enumerate the possible paths, as proposed in the online appendix of [1]. In order to avoid listing many paths with very low flow, we implement a threshold of 0.1 (a thousand less than the minimal OD demand on the network) on path flow under which we do not consider the path. This can yield small differences between the demand for a given OD and the sum of the flows on the paths of this OD. The path flows are then aggregated into OD-based link flows.

Figure 7: OD-based link flows distribution

2.6 Testing the hypothesis

The code for this section is in appendix A.3.

With the OD-based link flows both at UE and SO, we can now determine the links and

ODs that correspond to the left side of our hypothesis:

$$x_{ij}^{UE,\ od} > 0 \text{ and } x_{ij}^{SO,\ od} = 0 \Rightarrow \text{Link (i, j) is a Braess link for OD (o, d)}$$

Then we try, for each OD pair with at least one such link, to ban the links that correspond to our hypothesis. In order to analyse the results, we then also compute and compile other variables that could give insight into the improvement in total travel times, namely:

- Demand for that OD
- Number of removed links (equivalent to the number of identified link for that OD)
- Flow at UE for the given OD on the links (min, mean, max)
- Percentage of the OD UE paths impacted by the ban
- Percentage of the OD demand impacted by the ban (using the paths)
- Betweeness centrality of the links (min, mean, max) [5]
- Total flow on the links impacted (all OD) (min, mean, max)

This set of variables explore both the importance of the identified links for the given OD and in the general network, as well as the importance of the OD to the network. However, this only include a few topological measures (betweeness centrality) but further research into centrality measures and their potential as explanatory variables is probably needed.

For comparison, we also computed the effect of removing the most common links from the network.

3 Results

The outputs of the different algorithms, such as link flows, paths, etc... can be found in appendix A.4.

Visualisations of trafic flows at different scales (total, origin-based, OD-based) can be found in appendix A.2.

The code for the following analysis of the results can be found in appendix A.3.

3.1 Links identified

Comparing OD-based flows at UE and at SO, we identified 260 suspected Braess links in 68 different OD pairs (out of 528 OD pairs with demand). The number of links per OD is visible in Table 1, while the number of times a given link is identified is visible in Figure 8.

Table 1: Number of links concerned, per OD

Destination Origin	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0
2	0	0	0	0	2	0	0	0	2	2	0	0	0	0	2	0	7	0	6	0	0	6	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	3	0
5	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	5	0	3	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	3	0
7	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
10	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	2	2	4	4	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	4	0	0	2	0	0	0	0	0
13	0	0	0	0	0	0	5	0	0	0	0	0	0	3	9	0	0	0	11	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
15	0	2	0	0	0	0	0	0	0	0	2	2	9	0	0	0	0	0	0	0	0	0	0	0
16	0	0	2	2	2	0	0	0	2	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0
17	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	5	0	0	9
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	6	0	0	0	5	0	0	0	0	2	2	11	0	0	0	0	0	0	0	2	0	0	2
20	6	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	2	0	0	0	0	0	0	0
21	0	0	0	0	5	0	0	0	0	0	4	0	0	0	0	0	5	0	2	0	0	0	0	0
22	0	6	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	2	0	0	0	0	0

We can see that the OD pairs with the most links identified are from node 19 to node 13 and vice-versa. On the other hand, links around node 14 are identified the most frequently as suspected Braess links, with other hotspots around node 6 and node 20.

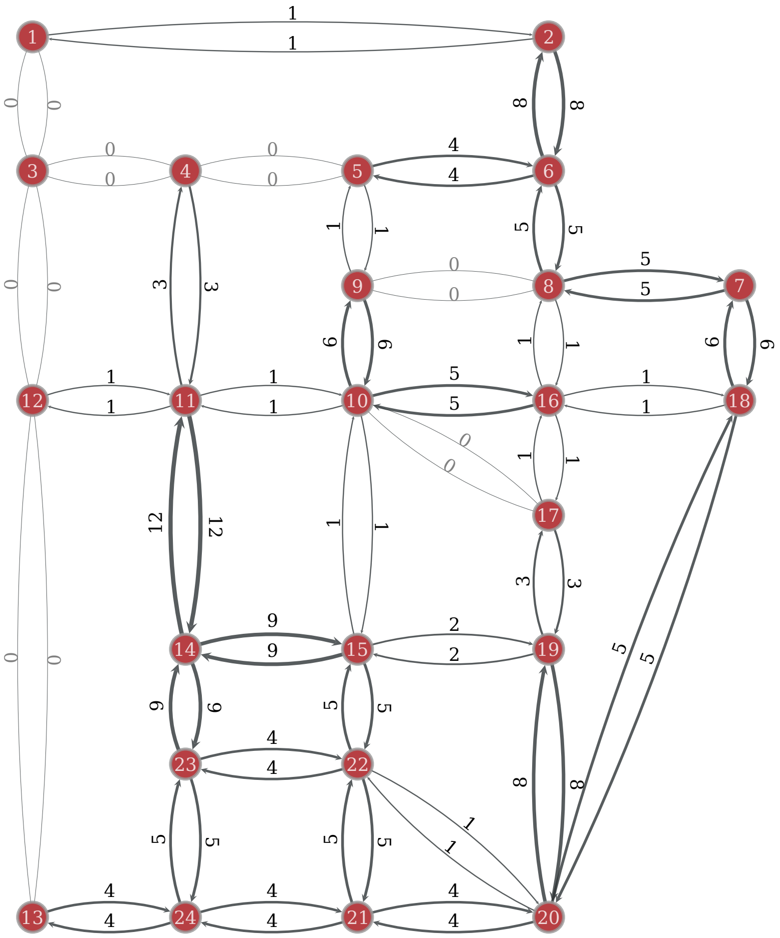
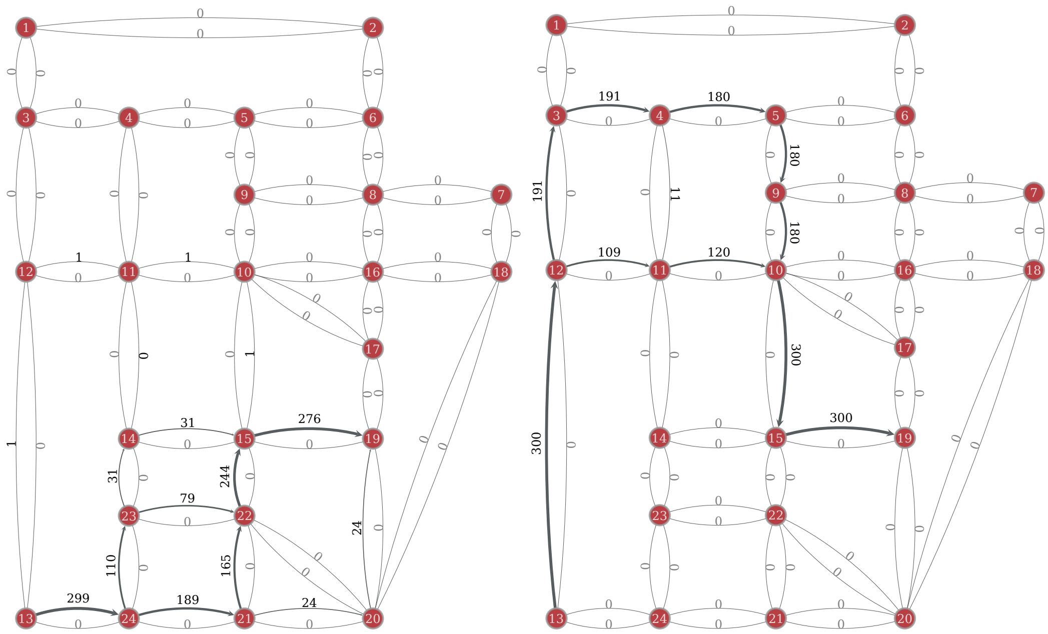


Figure 8: Number of OD pairs for which each link is identified

3.1.1 Visualisation of OD 13-19

Figure 9 presents the flows between origin node 13 and destination node 19, one of the OD pairs (the other one being the inverse of this one) with the most links identified. We indeed notice the SO flow is going through different links than the UE flow.



(a) at User Equilibrium (b) at System Optimum

Figure 9: Flows from node 13 to node 19

3.1.2 Analysis of link 34

Similarly, we can take a closer look at one of the most frequently identified links, link 34 from node 11 to node 14. Figure 10 highlights the ODs for which this link was identified as a potential Braess link (in red), as well as the ODs which uses link 34 but for which it was not identified as a potential Braess link (in blue). We can see that there seems to be a predisposition for ODs far from the link (coming from nodes 4, 5, 6 and going to node 23, or from nodes 15 to 22 to node 11) to seem to use link 34 as a Braess link, while ODs nearer the link (for example, going to node 14) do not validate our hypothesis condition.

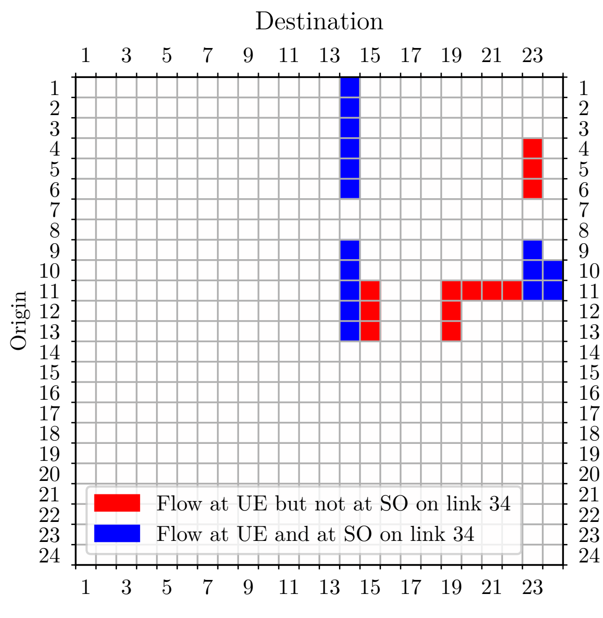


Figure 10: ODs that use link 34

3.2 Impact of banning links for a given OD

In this section, we study the impact of banning, for each OD pair, all links that corresponded to our condition. We will firstly analyse and visualise the results themselves, then we will try to match the impact of banning links with potential explaining variables.

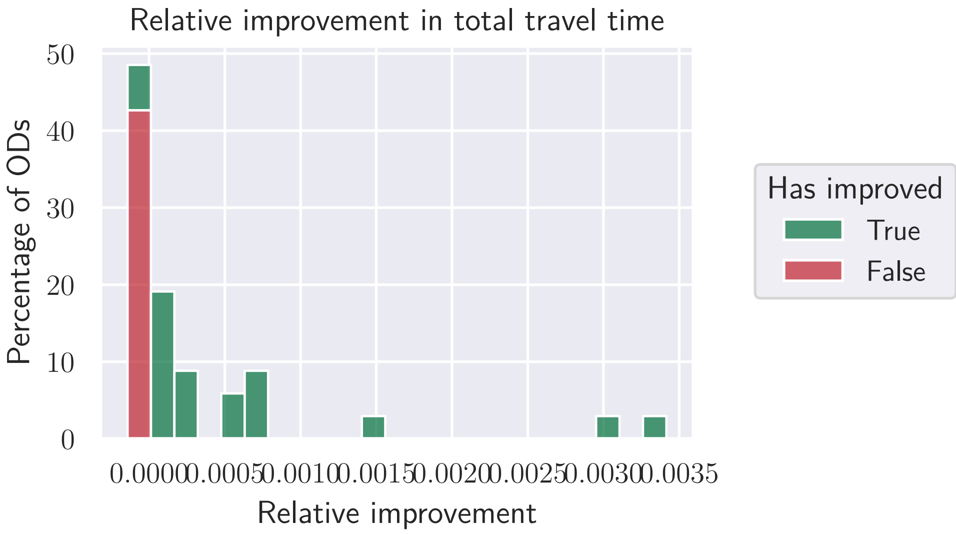


Figure 11: Total Travel Time improvement (relative)

Figure 11 shows the distribution of the improvement in total travel time, relative to the original travel time. It can be seen that in 57% of the ODs, banning links according to our hypothesis improved the travel time. However, we can also compute the difference an improvement in precision would bring in travel time. Going from 10^{-4} to 10^{-6} (in terms of the convergence criterion for Frank-Wolf) would increase total travel time by 0.00008 times, and 16% of the ODs have positive improvements smaller than this value. This is to keep in mind going further, as the significance of these small improvements could be questioned. An interesting fact is that when removing a link was counterproductive, the impact was relatively small, but when the impact is positive, it can grow quite big.

We can also take a look at the relationship between OD pairs and improvement, as shown in Figure 12. We can see that ODs that had the most improvements are linked to node 13, but we can not bring much results from this alone, it could also be that node 13 has the highest demands.

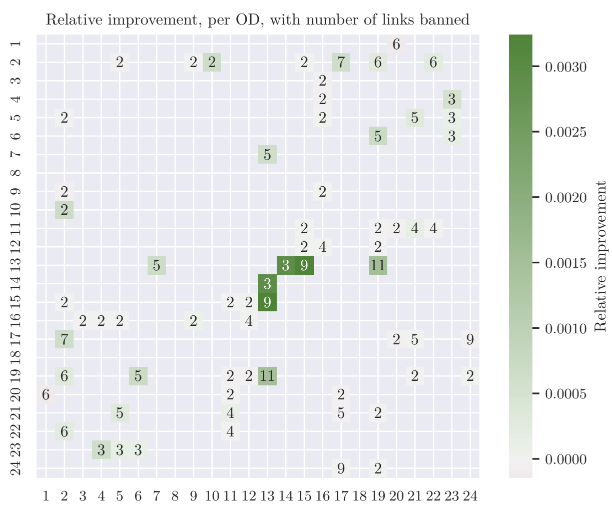


Figure 12: Relative improvement, per OD

3.2.1 Comparing with OD demand

One of the variables most susceptible of influencing the improvement in total travel time is the demand for the concerned OD. After visualising the relationship between both variables, we see indeed a dependent relationship (Figure 13, left). In order to see better the effects, making the reasonable hypothesis of a linear relationship, we normalise the improvement with the OD demand :

$$\text{Improvement normalised} = \frac{\text{ttt}^0 - \text{ttt}}{\text{OD demand}}$$

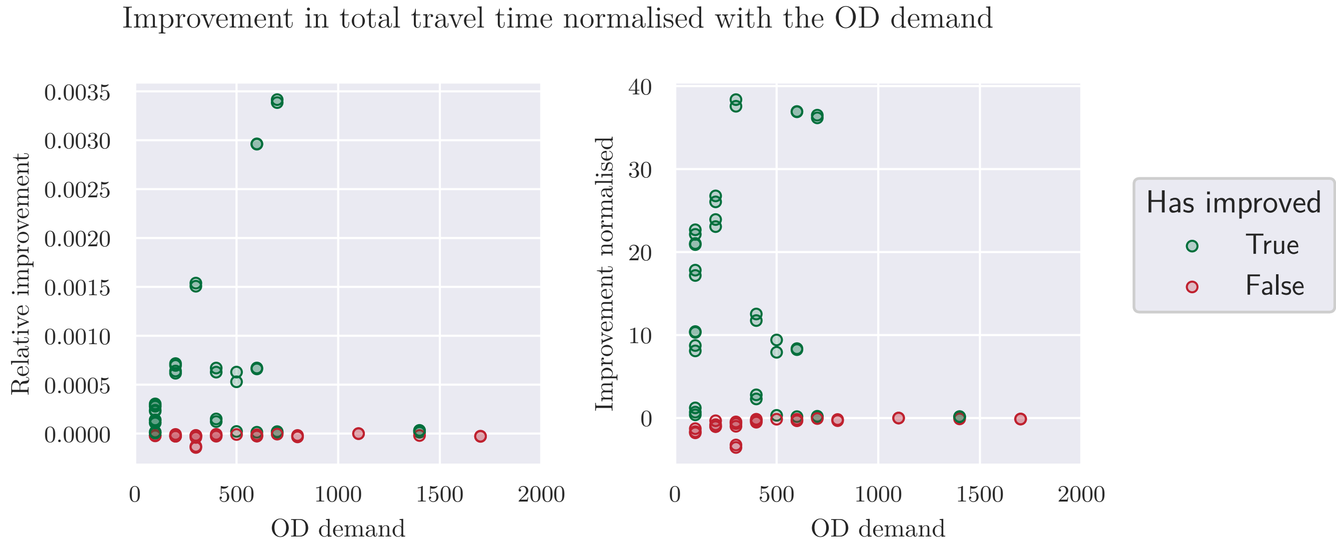


Figure 13: Improvement in Total Travel Time vs the OD demand

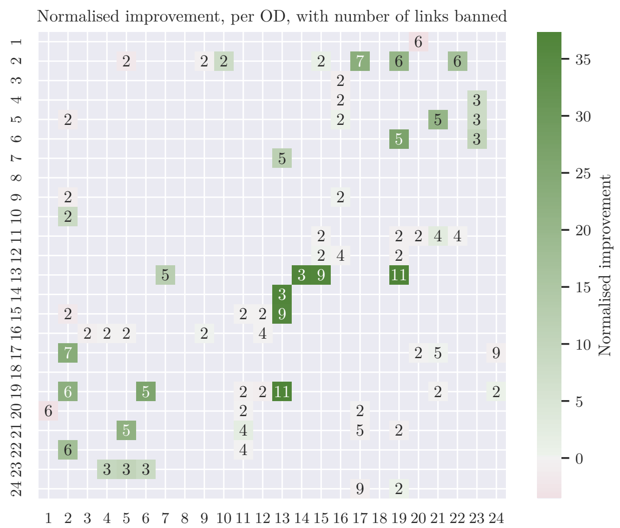


Figure 14: Relative improvement, per OD

Based on this improvement normalised, we can also update Figure 12 with the normalised demand, giving Figure 14. We notice that ODs with node 13 are still the best, but with also many interesting values in the top right and bottom left corners, which correspond to the farthest OD pairs.

3.2.2 Comparing with the number of removed links

Another interesting variable is the number of links we removed. This could serve as a proxy variable for the difference in paths between UE and SO for this OD, as well as to the distance between the origin and the destination. Comparing with the improvement normalised to the OD demand, we see indeed a tendency for improvements to be bigger for OD pairs with higher number of links removed, if there was indeed an improvement. There is still however one major outlier (corresponding to OD pairs 13 - 14), which could highlight the fact that the number of removed links is indeed a proxy variable for something more structural.

Figure 15: Improvement in Total Travel Time vs the number of removed links

3.2.3 Comparing with the flow redirected

One could speculate that the impact of banning links depend on the flow redirected, that is the OD-based flow at UE on the banned links. Since we banned multiple links for each OD pair, we aggregate these results through their minimum, mean and maximum values. Looking at Figure 16, we can see that there seems to be an upper bound to the improvement by the maximum flow on the links, however this effect is less visible once we normalise with the OD demand. That being said, an interesting computation to make would be to normalise this UE flow on banned links with the OD demand as well.

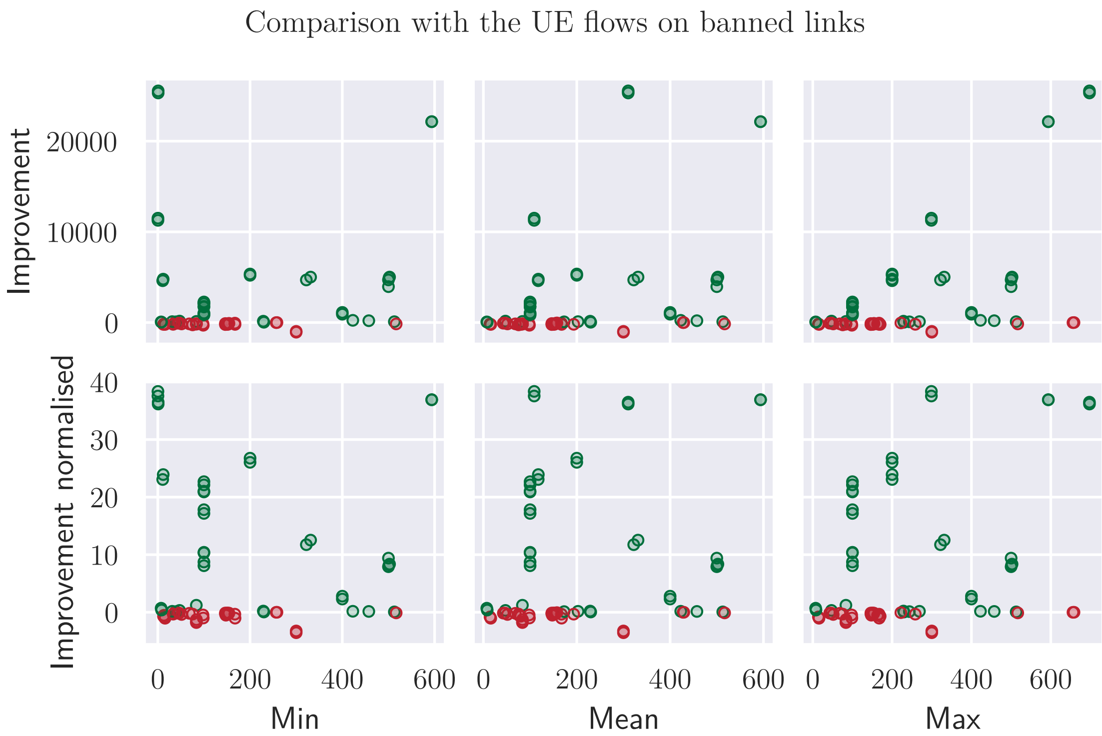


Figure 16: Comparison with the UE flow on banned links

An interesting visualisation is the cumulative histogram of these values, separated by whether there was an improvement (Figure 17). Here, we see that the proportion of improved ODs in high UE flow redirected is higher.

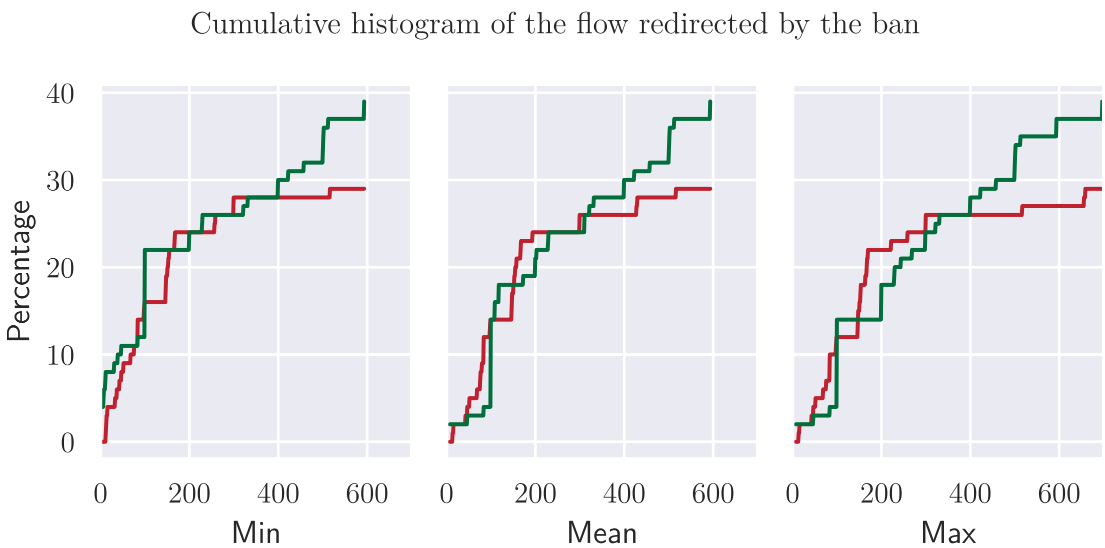


Figure 17: Comparison with the UE flow on banned links

3.2.4 Comparing with the demand and paths impacted

Now, we look at the percentage of the paths, and the percentage of OD demand that was redirected (using the paths to avoid considering multiple times the same flow on different links). As seen in Figure 18, ODs where the redirection impacted less than 80% of the trafic did not have meaningfull improvement. This is an interesting result, that could be expanded into further research. That being said, it gives a necessary condition, but not a sufficient condition as there are still ODs where all the demand was impacted with little to no improvement when banning links.

Figure 18: Comparison with the UE flow on banned links

3.2.5 Comparing with the betweenness centrality of the banned links

The betweenness centrality is an interesting centrality measure, that measures, for each edge, the ratio of shortest paths between node pairs that it is a part of [5]. It could provide more insight on the structure of the network and the importance of the impacted links in the network. However, on Figure 19, we do not see much more. It could maybe be argued that including an edge with high betweenness centrality improves the impact, but this needs to be researched further.

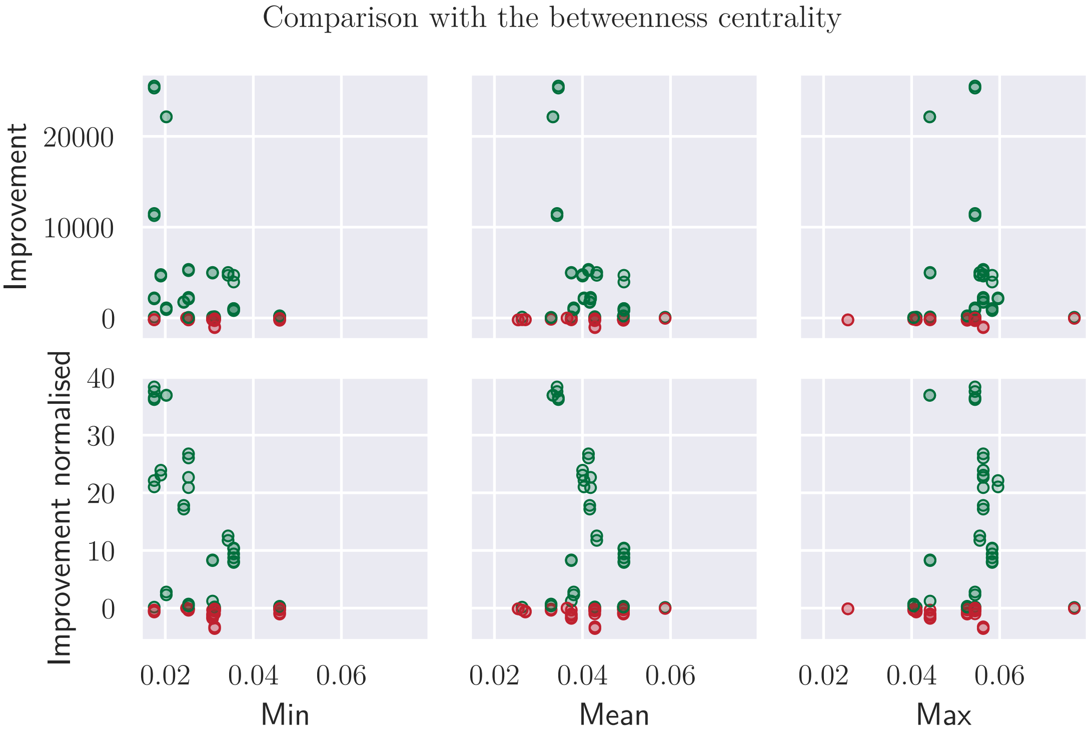


Figure 19: Comparison with the betweenness centrality

3.2.6 Comparing with the total flow on the banned links

Finally, we compare the results with the total flow on the banned links, regardless of the OD, which could be interpreted as an indice to the importance of the links. The results are however less meaningfull than for betweenness centrality.

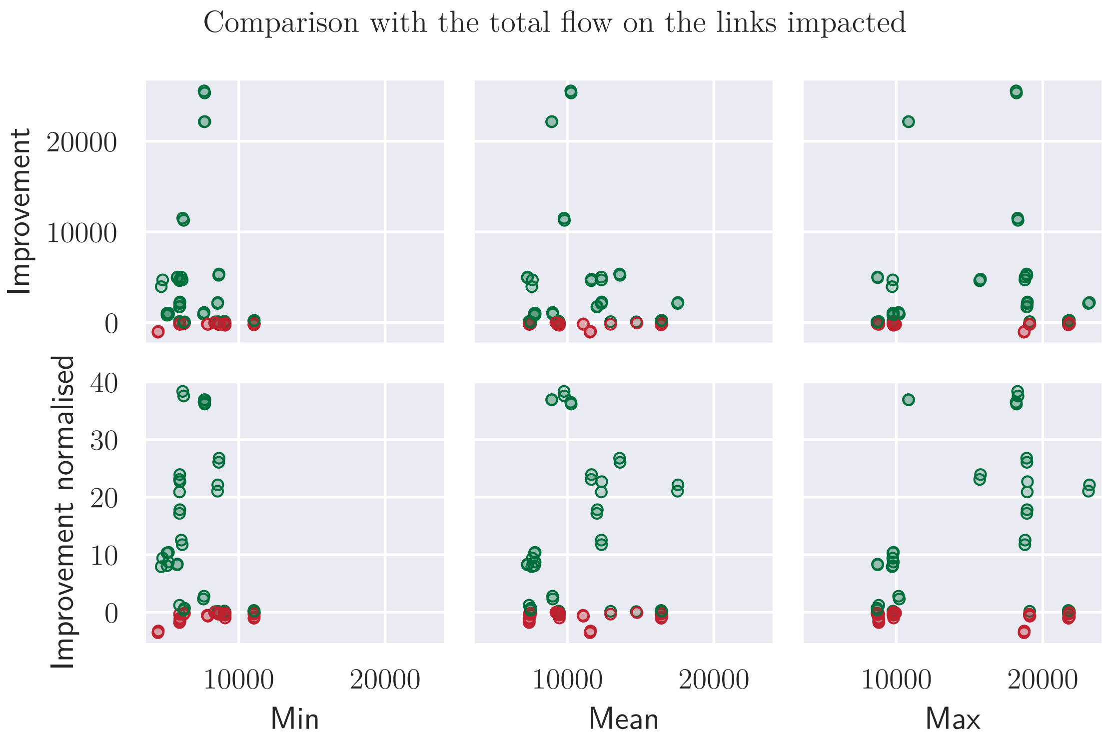
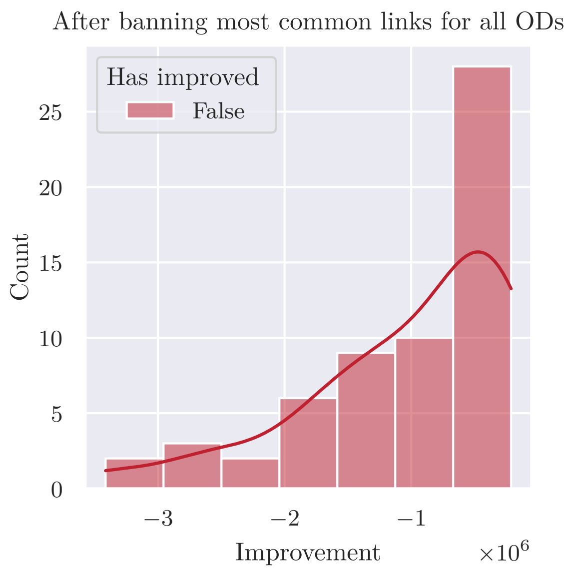


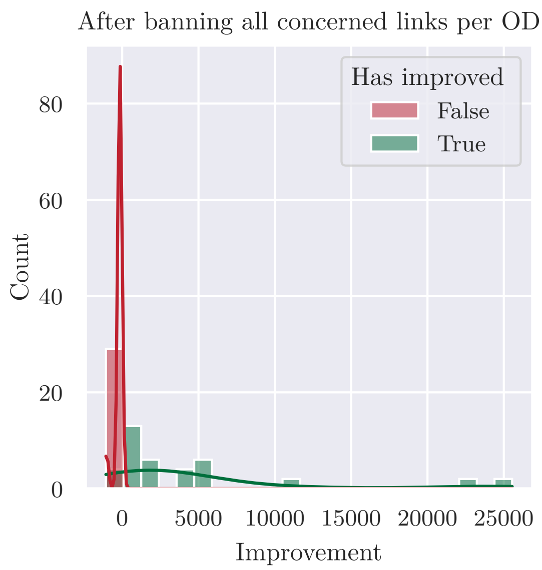
Figure 20: Comparison with the total flow on the banned links

3.3 Impact of banning the links for all ODs

In order to compare, we banned the most common links to all OD. This never brought improvements, and the scale of the lost travel time is bigger. This would be expected as we are impacting much more trafic.



(a) After banning all trafic from identified links



(b) After banning identified links for one OD

4 Conclusion

To conclude, our hypothesis proved to have interesting potential to identify Braess links. On the classic SiouxFalls network, we successfully identified potential Braess links and demonstrated that banning these potential links to certain OD pairs can, in certain cases, lead to improved total travel times. However, the variability in results suggests that further investigation is necessary to refine the identification process and establish more reliable indicators of paradoxical links.

Potential areas for future research include exploring potential explanatory variables that influence the occurrence of Braess' Paradox. Exploring the impact of the structure of the network as well as looking at the effect of banning individual links (instead of a group of link) could yield interesting results and allow us to get more precise insight. The study could also be expanded to Mixed Equilibrium models to see if the targetness of the hypothesis improve. As computational power has been a limiting factor, improving the algorithms or having more computational power at disposition could bring better results, as we have seen that some of our results are in the range of a precision error. Better computational efficiency could also allow to compare with other, more complex, networks.

This study provides a valuable framework to look into counterintuitive traffic behaviors and optimization strategies. The different algorithm implementation as well as the multiple visualisation routines developped could help further this research to a more definite result.

A Appendix

The code and the outputs are availlable on Github, at https://github.com/merlebleue/Braess-Links-Identification. In particular :

A.1 Algorithms

The algorithms can be found in the following files :

- FrankWolf.py : Frank Wolf, including shortest path algorithm and cost functions.
- EntropyMaximisation.py : EMARB, including backward and forward entropy maximisation.
- RemoveBraess.py : The removal of links from the network.
- Network.py : The implementation of the Network class, which supersedes the Graph class from graph-tool [3], and implements loading a network out of the TransportationNetworks files as well as drawing, saving, exporting and loading routines.

A.2 Initial network calculations

The initial network calculations, that is trafic at UE and SO code is availlable in the Initial network calculations.ipynb notebook.

A.3 Hypothesis testing

The testing of the hypothesis, including removal of links and analyse of the results, is availlable in the hypothesis testing.ipynb notebook.

A.4 Code outputs

Outputs of the different algorithms can be found in the exports folder.

A.5 Example implementation of EMARB

The testing of our implementation of EMARB on the example from [1] is availlable in the example network.ipynb notebook.

References

[1] Jun Xie and Yu (Marco) Nie. A new algorithm for achieving proportionality in user equilibrium traffic assignment. *Transportation Science*, 53(2):566–584, 2019. (pages 1, 4, 4, 6, 19).

[2] Transportation Networks for Research Core Team. Transportation networks for research. `https://github.com/bstabler/TransportationNetworks`, 2023. Accessed October 2024. (pages 1, 2, 2, 3, 3).

[3] Tiago P. Peixoto. The graph-tool python library. *figshare*, 2014. (pages 2, 19).

[4] Yossi Sheffi. *Urban Transportation Networks: Equilibrium Analysis With Mathematical Programming Methods*. 01 1984. (page 3).

[5] Mark Newman. *Networks*. Oxford University Press, 07 2018. (pages 7, 15).