Experimentally Exploring Braess's Paradox

This report serves two purposes:

1. To summarize one consequence that results from a big idea that is well-explored in the cross-disciplinary realm of economics and theoretical computer science: the Price of Anarchy. This tells us that Braess's Paradox dilates affine-costed traffic by at most 4/3.

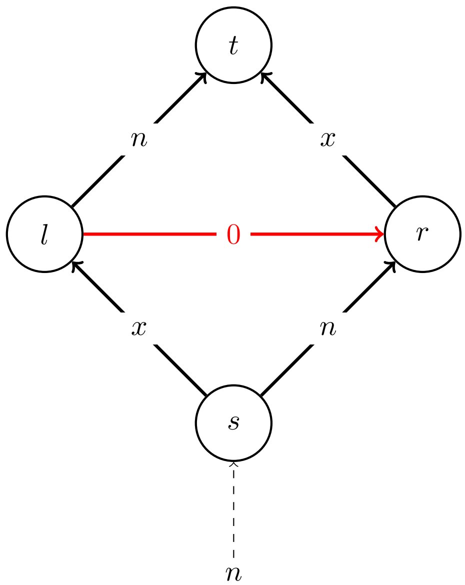
2. To suggest further avenues one may pursue the exploration of Braess's Paradox experimentally so as to attain a better understanding of this curious phenomenon.

Our story begins with Dr Z's challenge problem. Therein, the famous Braess's Paradox is explained, and a challenge proposed. We sketch a rough idea exhibiting a tight solution (using none of our own ideas) to the challenge.

Most if not all of these ideas are a rewriting of ideas whose origins are attributable to Roughgarden. Roughgarden has been involved in solving many fundamental problems linked to Braess's Paradox (see The POA of Selfish Routing). In fact, the paper [1] which first established the bound on the soon-to-be-mentioned ratio was one of three works awarded the 2012 Gödel Prize.

In the latter half of this report, we try to experimentally ratify and/or extend some ideas in the current literature, and through the results (or lack thereof) of the experiments, suggest other directions one may take to explore Braess's Paradox.

1 Network achieving $4/3$ dilation



Without the red arc, the average travel time is $3n/2$; half the cars take the path s, l, t and the other half take the path s, r, t. With the red arc, the average travel time is $2n$; all the cars take the path s, l, r, t. Dividing the latter by the former yields a ratio of $4/3$.

This alone yields a better ratio than the one posed in Dr Z's challenge problem. In fact, as we will presently show, this is the best ratio achievable.

2 A 4/3 dilation is tight for affine costs

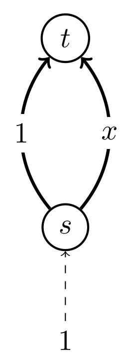
We model traffic as an indiscrete st-flow. We first define the notion of Price of Anarchy, and later relate it to the notion of Braess Ratio. The goal is to show Braess Ratio $\leq$ Price of Anarchy $\leq 4/3$.

2.1 Price of Anarchy

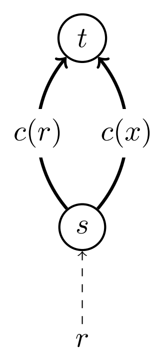
Informal Definition 1 (Price of Anarchy (POA)). *The POA of a network is*

$$\frac{\textit{Wardrop Equilibrium Average Travel Time}}{\textit{Optimal Average Travel Time}}.$$

The maximum POA over all networks with affine costs with non-negative coefficients can be achieved by an instance of what's known as a Pigou-like Network:



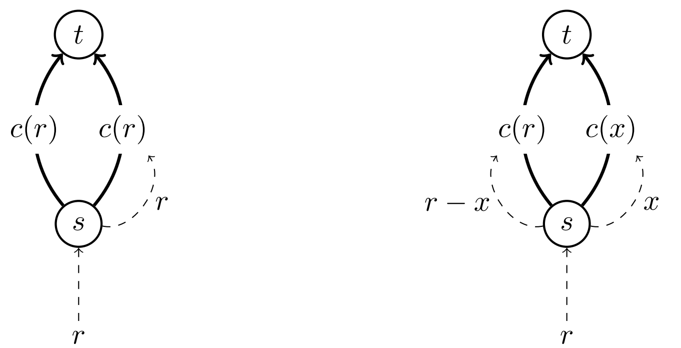
This result is generalizable to broader classes of monotonically non-decreasing costs; a Pigou-like Network (that is, the same graph topology with costs coming from some class $\mathcal{C}$, say for example, of bounded degree polynomials with non-negative coefficients) achieves the maximum POA over all networks whose costs come from $\mathcal{C}$.

$\sup_{c\in\mathcal{C},r}$(POA )

Given a class of monotone costs $\mathcal{C}$, the POA over Pigou-like Networks can be formulated like so:

$$\alpha(\mathcal{C}) = \sup_{\substack{c \in \mathcal{C} \\ 0 \leq r \\ 0 \leq x \leq r}} \frac{r \cdot c(r)}{x \cdot c(x) + (r - x) \cdot c(r)}, \tag{1}$$

where the numerator is a Wardrop Equilibrium sending all traffic along the non-constant edge.



Numerator flow　　　　Denominator flow

We want to use an explicit formulation like equation 1 to bound the POA of any other network, but first, we can remove the constraint $x \leq r$ since $\mathcal{C}$ is a class of monotone costs:

$$\begin{aligned} r < x \implies & \frac{r \cdot c(r)}{x \cdot c(x) + (r - x) \cdot c(r)} \\ \leq & \frac{r \cdot c(r)}{x \cdot c(x) + (r - x) \cdot c(r)} && \text{(monotonicity of } c\text{)} \\ = & \frac{r \cdot c(r)}{r \cdot c(x)} \\ = & \frac{c(r)}{c(x)} \\ \leq & 1, && \text{(monotonicity of } c\text{)} \end{aligned}$$

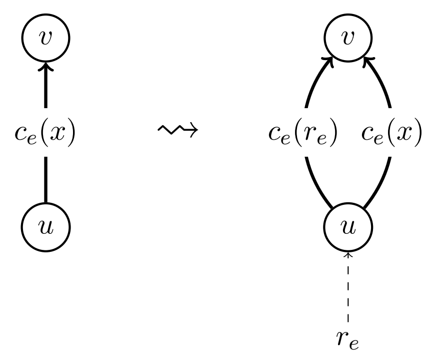
and thus formulate the POA over Pigou-like Networks using $\mathcal{C}$ like so:

$$\alpha(\mathcal{C}) = \sup_{\substack{c \in \mathcal{C} \\ 0 \leq r \\ 0 \leq x}} \frac{r \cdot c(r)}{x \cdot c(x) + (r - x) \cdot c(r)}. \tag{2}$$

Formulation 2 is helpful as we will look at an equilibrium flow r_e along an edge e and a socially optimal flow x_e along the same edge, and relate those quantities to the bound given by a Pigou-like Network spanning the endpoints of e; there is no guarantee that the flows $x_e \leq r_e$ hence our removal of the constraint $x \leq r$. We will then sum over all edges to get the 4/3 POA bound. Let us now formalize this.

Theorem 1. *Let $\mathcal{C}$ be the set of affine costs with non-negative coefficients. Let $G = (V, E)$ be any network whose arcs have costs coming from $\mathcal{C}$. Then the POA of G is at most 4/3.*

Proof sketch. Fix an edge $(u, v) = e \in E$. Let r_e be the Wardrop Equilibrium flow going through e. Let x_e be the socially optimal flow going through e. Let $c_e(x)$ be the cost of going through e. Consider the Pigou-like Network with total flow r_e going from u to v using c_e.



By equation 2 we have that

$$\alpha(\mathcal{C}) \geq \frac{r_e \cdot c_e(r_e)}{x_e \cdot c_e(x_e) + (r_e - x_e) \cdot c_e(r_e)}.$$

and so, multiplying both sides by the denominator (and swapping sides),

$$r_e \cdot c_e(r_e) \leq \alpha(\mathcal{C})(x_e \cdot c_e(x_e) + (r_e - x_e) \cdot c_e(r_e)). \tag{3}$$

Then

$$\begin{aligned}
&\text{Wardrop Equilibrium Average Travel Time} \\
=&\sum_{e\in E} r_e \cdot c_e(r_e) \\
\leq&\alpha(\mathcal{C}) \sum_{e\in E} (x_e \cdot c_e(x_e) + (r_e - x_e)\cdot c_e(r_e)) && \text{(Inequality 3)} \\
=&\alpha(\mathcal{C}) \sum_{e\in E} x_e \cdot c_e(x_e) + \alpha(\mathcal{C}) \sum_{e\in E} (r_e - x_e)\cdot c_e(r_e) \\
\leq&\alpha(\mathcal{C}) \sum_{e\in E} x_e \cdot c_e(x_e) && \text{(Lemma 1)} \\
=&\alpha(\mathcal{C}) \cdot \text{Optimal Average Travel Time} \\
=&\frac{4}{3} \cdot \text{Optimal Average Travel Time.} && \text{(Lemma 2)}
\end{aligned}$$

□

We now tie up loose ends left in the proof of Theorem 1 by showing Lemma 1 and Lemma 2.

Lemma 1. *Let r be a Wardrop Equilibrium flow and x be any other flow of the same value on some network $G = (V, E)$. Let r_e, x_e stand for the respective flows along arc $e \in E$. Let $c_e \in \mathcal{C}$ be the cost on the arc $e \in E$. Then*

$$\sum_{e\in E} (r_e - x_e) \cdot c_e(r_e) \leq 0$$

Proof sketch. Let P, P' be any pair of *st* paths, and r_P be the Wardrop Equilibrium flow along P. Then observe that if $r_P > 0$, we have

$$\sum_{e\in P} c_e(r_e) \leq \sum_{e\in P'} c_e(r_e)$$

or else traffic along P will be incentivized to move to P' which is precluded by the traffic being at a Wardrop Equilibrium. That is to say,

Observation 1. *In a Wardrop Equilibrium, the cost of every path for which there is positive flow is the same (say C):*

$$r_P > 0 \implies c_P(r_P) = C$$

and

Observation 2. *In a Wardrop Equilibrium, the cost of every path is at least as much as the cost of a path for which there is positive flow:*

$$c_P(r_P) \geq C.$$

We are now ready to conclude

$$\begin{aligned}
\sum_{e \in E} r_e \cdot c_e(r_e) = & \sum_{P \in \mathcal{P}} r_P \cdot c_P(r_P) \\
= & r \cdot C && \text{(Observation 1)} \\
= & x \cdot C \\
\leq & \sum_{P \in \mathcal{P}} x_P \cdot c_P(r_P) && \text{(Observation 2)} \\
= & \sum_{e \in E} x_e \cdot c_e(r_e)
\end{aligned}$$

and so

$$\sum_{e \in E} (r_e - x_e) \cdot c_e(r_e) \leq 0.$$

□

Lemma 2. *Let $\mathcal{C}$ be the class of affine costs with non-negative coefficients. Then $\alpha(\mathcal{C}) = 4/3$.*

Proof sketch.

$$\sup_{\substack{0 \leq a \\ 0 \leq b \\ 0 \leq r \\ 0 \leq x}} \frac{r(ar + b)}{(r - x)(ar + b) + x(ax + b)} = 4/3$$

□

2.2 Braess Ratio

Informal Definition 2 (Braess Ratio)**.** *The Braess Ratio of a network $G = (V, E, c)$ is*

$$\max_{e \in E} \frac{\textit{Wardrop Equilibrium Average Travel Time of } G}{\textit{Wardrop Equilibrium Average Travel Time of } G - e}.$$

The Wardrop Equilibrium flow of $G - e$ is a feasible flow in G and so it is clear that Braess Ratio $\leq$ Price of Anarchy $\leq 4/3$.

The 4/3 ratio achieved by the simple network in Section 1 is thus tight.

3 What next?

In view of the aforementioned result, we now propose some directions we may further explore Braess's Paradox. The remainder of this report will comprise of some exposition on a selection of several of these possible directions.

3.1 Improving traffic

3.1.1 Adding an arc

Given a network, how can we add an arc to best reduce the average travel time? Some constraints need to be enforced to the extent that the solution is not as easy as joining s to t with a 0 cost arc.

3.1.2 Changing the cost of an arc

Given a network, how can we change the cost of one arc to best reduce the average travel time? This is not completely straightforward since reducing the cost of an arc may actually worsen the average travel time. Think about the original network which inspired this project: changing the cost of the (l, r) arc to something very high (say ∞) restores the damaged average travel time.

3.2 Occurrence of Braess's Paradox in random graphs

Roughgarden and Valiant show that Braess's Paradox occurs with high probability in the Erdös-Renyi model for random graphs in [2] (see here). They leave open the problem of exploring this phenomenon in other models of random graphs (where a typical graph is "sufficiently dense and uniform") and, even more curiously, models of sparse or non-uniform graphs.

3.3 Braess's Paradox games

Analyze the following games. Note that the games are impartial and so we have at our disposal tools like Sprague-Grundy numbers in order to characterize winning positions.

3.3.1 Network labelling

We start with a directed graph. Alice and Bob take turns labelling the arcs with affine costs. The game ends when all the arcs have been labelled. Bob wins if the removal of any of the arcs yields lower travel time. Alice wins otherwise.

The initial version of this game is included in the modified Maple package. Alice and Bob take turns, choose a random linear function, and assign that cost to the first existing edge which doesn't have a cost yet. The program then outputs who wins based on whether the network is Braess or not.

The next version will be interactive, allowing the players to choose an existing edge with no cost function and choose a cost function to assign.

3.3.2 Network building

We start with n isolated vertices. Alice and Bob take turns to draw an arc between the vertices. The game ends when s has a path to every vertex and every vertex has a path to t. Bob wins if there is a cost labelling of the arcs such that the removal of any of the arcs yields lower travel time. Alice wins otherwise.

3.4 Quartic costs

The U.S. Bureau of Public Roads at least once upon a time used quartic costs to model traffic (see book-page 358). While the POA of quartic costs has been figured out, I am not sure if we can match it using the canonical Braess's Paradox network. Try to get as close to it as possible on the canonical Braess Paradox network, and try to find other simple networks whose Braess Ratio can get close to the POA.

4 Occurrence of Braess's Paradox in random graphs

[2] shows Braess's Paradox manifests with high probability for large enough random networks under the Erdös-Renyi model. We will focus more on their $1/x$ model where arcs are given the costs x or 1. Their result (theorem 4.1 in [2]) is as follow:

Theorem 2. *Let $p, q, \varepsilon \in (0, 1)$ be constants. With high probability, a sufficiently large random network from $G(n, p)$ with costs being x with probability q and 1 otherwise (with probability $1 - q$) admits a traffic rate such that the Braess Ratio of the network is at least*

$$\frac{4 - 3pq}{3 - 2pq} - \varepsilon.$$

We will explore this theorem experimentally, under some relaxations, and under alternative models for random networks.

As our code models traffic discretely as a whole number of cars rather than as some amount of continuous flow, we will take some liberty in rounding the flow of npq used in [2] to $\lceil npq \rceil$ cars.

Secondly, we relax the precision to which we measure the Braess Ratio. Instead of pinning down the ratio, we simply check whether it is greater than 1 (i.e. it is Braess). Insofar as our experimentation is concerned, our focus is on whether a network is bad rather than how bad a network is.

4.1 Watts-Strogatz Model

As an alternative to the Erdös-Renyi model, we can consider random graphs under the Watts-Strogatz model. The latter starts with some base graph structure (for example a grid network), and perturbs each arc (u, v) from the base graph with some probability to (u, w) where w is taken uniformly. The arc-rewiring of the Watts-Strogatz model constrains randomness enough to capture what is known as small-world phenomena [3], which is present in many real-world networks.

The original conception of the Watts-Strogatz model begins with a ring lattice as a base graph; that is, vertices v_i and v_j are connected when $|i - j|$ is small. We take some liberties in generating a graph under this model by starting off instead with a grid network (the "perfect" city layout), that is also directed down and to the right; our source will be the upper-left corner of the grid and our destination will be the lower-right corner of the grid.

Our Maple package includes a function RGRID for generating grids with arc costs coming from $\{1, x\}$. It also includes an experiment to determine how likely a random grid is to be Braess.

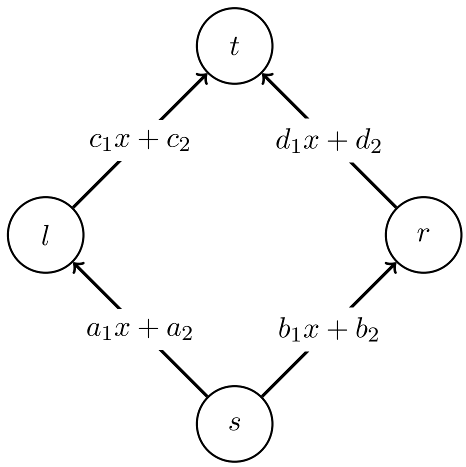
Finally, our Maple package includes a function WATTSSTROGATZ to generate a random graph starting with a grid network as a base, along with an experiment to get a handle on the likelihood of such graphs being Braess.

The accompanying experiments are unfortunately run for only feasible values of n^2 (the graph size); no occurrence of Braess Paradox is observed. Perhaps more efficient algorithms are needed in order to run experiments on much larger graphs, or perhaps this is an effect caused by discrete traffic; the results are contrary to our suspicion that Braess Paradox ought to be observed in such graphs.

5 Improving Traffic by Adding an Arc

In this section we attempt to avoid instances of Braess's Paradox. Given a network, we want a systematic way to add an arc that improves the average delay time. To ensure that our solution is not trivial, we never let the new arc directly join s to t.

We begin by considering the network originally provided to us in Dr Z's challenge problem but with arbitrary arc weights.



In this particular network, there is only one place to add an arc, i.e., the one joining l to r. So, we want to determine the weight of the arc from l to r so that traffic improves.

The weight of each arc is a linear function with nonnegative, real coefficients. If we let the weight of the new arc be e_1x+e_2, we find that traffic usually improves as long as $e_1 \leq \min\{a_1, b_1, c_1, d_1\}$ and $e_2 \leq \min\{a_2, b_2, c_2, d_2\}$. Experimenting with the range of the coefficients a_i, b_i, c_i, d_i and the number of cars driving on the network, we find that this method of adding an arc successfully improves traffic on at least 50% of these networks with randomly generated arc weights. Furthermore, the success rate of this method roughly increases (i.e. not monotonically) with the number of drivers. So far our experiments show that this method works at least 90% of the time when there are at least 15 cars.

We now this method to arbitrary directed networks without cycles in which each arc weight is a linear function with nonnegative, real coefficients. As before, we bound the linear coefficient of our new arc weight above the linear coefficients of the existing arcs. We similarly bound the constant coefficient of our new arc weight. We also search for the best place to add an arc with such a weight so that traffic decreases as much as possible. This method is not as successful on more complicated networks, but so far our experiments show that when there are at least 2 cars, it works at least %50 of the time.

The Maple package includes functions that add an arc using this method to an arbitrary network and also to the original network but with randomly generated arc weights. It also includes functions that test how successful this method is on randomly generated graphs. We need to experiment some more to get a better estimate of how often this method of adding an arc successfully improves traffic. We also need to explore other ways of adding an arc to improve traffic.

References

[1] Tim Roughgarden and Éva Tardos. "How bad is selfish routing?" In: *Journal of the ACM (JACM)* 49.2 (2002), pp. 236–259.

[2] Gregory Valiant and Tim Roughgarden. "Braess's paradox in large random graphs". In: *Random Structures & Algorithms* 37.4 (2010), pp. 495–515.

[3] Duncan J Watts and Steven H Strogatz. "Collective dynamics of 'small-world'networks". In: *nature* 393.6684 (1998), pp. 440–442.