Braess’ Paradox

Julian Romero

January 22, 2008
Outline

The Paradox

Severity

Linear Latency

General Latency

Extensions

Conclusion
Braess’ Paradox

- Introduced by Dietrich Braess in 1968.
- Adding costless edges doesn’t necessarily decrease traffic costs.

(a) The Whole Graph

(b) The Optimal Subgraph
Braess’ Paradox

- Cost of (a) is 2, Cost of (b) is $3/2 \Rightarrow \text{PoA} = 4/3$.
- Worst Case scenario. (PoA is bounded by $4/3$ with linear cost functions.)

(a) The Whole Graph

(b) The Optimal Subgraph
Examples

- Stuttgart, Germany invested in new route in city center to ease traffic.
- Traffic got worse, didn’t improve traffic until it was blocked from traffic.
- Earth Day 1990 - NYC closed 42nd street.
- Many were worried, actually improved.
- Seoul, South Korea - Invested $380m to tear down one of three main bridges into town.
- Santa Cruz added 2 lanes to highway.
- Carpool Lanes?
Other Work

- Steinberg and Zangwill (1983) - Show that Braess’ Paradox occurs about half the time in random graphs.
- Pas and Principio (1997) - Show that Braess’ Paradox only occurs demand for travel fall in an intermediate range.
- Tumer and Wolpert (2000) - Show that Braess’ Paradox can be avoided with COIN.
- In experimental lab, Braess’ paradox has been shown, but not for larger Braess’ graphs.
Application to Other Models

- Loss Networks
- Queuing Networks
- Used to explain Downs-Thomson Paradox
- Electric Networks
- Simple model of water pipes exhibits no Braess’ paradox, but more complicated model does (not sure about experimental data).
- Resolving Newcomb’s Problem
- String and Springs network
Game

- Not uncommon in game theory to give more options and make worse off.

\[
\begin{array}{cc}
  & L & M \\
  U & 4,4 & 5,1 \\
  M & 1,5 & 3,3 \\
\end{array}
\]
Game

- Not uncommon in game theory to give more options and make worse off.

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>4,4</td>
<td>5,1</td>
</tr>
<tr>
<td>M</td>
<td>1,5</td>
<td>3,3</td>
</tr>
</tbody>
</table>

\[
\begin{array}{ccc}
  & L & M & R \\
  U & 4,4 & 5,1 & -10,10 \\
  M & 1,5 & 3,3 & -10,10 \\
  D & 10,-10 & 10,-10 & -9,-9 \\
\end{array}
\]

Romero
**Game**

- Not uncommon in game theory to give more options and make worse off.

```
\[\begin{array}{cc}
U & L & M \\
L & 4,4 & 5,1 \\
M & 1,5 & 3,3
\end{array}\]
```

- “This is not a real paradox but only a situation which is counterintuitive.”

```
\[\begin{array}{ccc}
 & L & M & R \\
U & 4,4 & 5,1 & -10,10 \\
M & 1,5 & 3,3 & -10,10 \\
D & 10,-10 & 10,-10 & -9,-9
\end{array}\]
```
On the Severity of Braess’ Paradox: Designing Networks for Selfish Users is Hard

Tim Roughgarden

- Typically network designers look to find smallest or cheapest network satisfying conditions.
- This paper takes a given network, and tries to determine if cost can be improved by removing links.
- Can we determine what networks are susceptible to Braess’ Paradox, and would benefit by eliminating links?
Model

- Directed Network $G \in (V, E)$.
- Unique source $s$ and sink $t$.
- $\mathcal{P}$ is the set of $s$-$t$ paths.
- $f: \mathcal{P} \to \mathbb{R}^+$
- Flow on edges $f_e = \sum_{P: e \in P} f_P$.
- Traffic flow $r$.
- Feasible flow, $\sum_{P \in \mathcal{P}} f_P = r$.
- Each edge has latency function $\ell_e(\cdot)$.
- $\ell_e$ is non-negative, continuous, and non-decreasing.
- Latency on path, $\ell_P(f) = \sum_{e \in P} \ell_e(f_e)$
- An instance is the triple $(G, r, \ell)$. 
Definitions

- A flow $f$ feasible for $(G, R, \ell)$ is at Nash equilibrium, or is a Nash flow, if for all $P_1, P_2 \in P$ with $f_{P_1} > 0$ and $\delta \in (0, f_{P_1}]$, we have,

$$\ell_{P_1}(f) \leq \ell_{P_2}(\tilde{f}) \text{ and } \tilde{f}_P = \begin{cases} f_P - \delta & \text{if } P = P_1 \\ f_P + \delta & \text{if } P = P_2 \\ f_P & \text{otherwise} \end{cases}$$

- Cost of flow,

$$C(f) = \sum_{P \in P} \ell_P(f)f_P$$

- A $c$-approximation run in polynomial time, and returns no more than $c$ times the optimal solution.
Properties of Nash Flows

- In equilibrium \( \ell_{P_1}(f) = \ell_{P_2}(f) \) (latency equal on all paths).
  - Call this common latency \( L(G, R, \ell) \).
- Equilibrium exists for all instances \( (G, r, \ell) \).
- Equilibrium “unique”, i.e. \( \ell_P(f) = \ell_P(f') \).
- At equilibrium
  \[
  C(f) = r \cdot L(G, r, \ell)
  \]
- For every instance, \( L(G, r, \ell) \) is non-decreasing in \( r \).
### Linear Latency Functions

- \( \ell_e(x) = a_e x + b_e \) with \( a_e, b_e \geq 0 \).

- Prop: \( f^* \) feasible flow, \( f \) Nash flow. For any instance \((G, r, \ell)\) with linear latency functions,

\[
C(f) \leq \frac{4}{3} C(f^*)
\]

- Trivial algorithm - network will all candidate edges.
- The trivial algorithm is a \( 4/3 \)-approximation algorithm for linear latency network design.
- Theorem - Assuming \( P \neq NP \), for every \( \varepsilon > 0 \), there is no \( (\frac{4}{3} - \varepsilon) \)-approximation algorithm for Linear Latency Network Design
Linear Latency

- Call instance $(G, r, \ell)$
  - paradox-free if,
    \[ L(G, r, \ell) \leq L(H, r, \ell) \]
    for all sub graphs $H$ of $G$
  - paradox-ridden if,
    \[ L(G, r, \ell) = \frac{4}{3} L(H, r, \ell) \]
    for some subgraph $H$ of $H$

- Corollary - Given an instance $(G, r, \ell)$ that has linear latency functions and is either paradox-free or paradox-ridden, it is NP-hard to decide whether or not $(G, r, \ell)$ is paradox-ridden.
General Latency Functions

- Examine continuous and non-decreasing latency functions.
- PoA is unbounded for these functions.

- Compare only Nash-flows rather than all feasible flows.
- Theorem - For every instance \((G, r, \ell)\) with \(n\) vertices, the trivial algorithm returns a solution of value at most \([n/2]\) times that of an optimal solution.
Braess’ Graph

- Define class of “Braess’ Graphs”
- Vertices - $V^k = \{s, v_1, \ldots, v_k, w_1, w_k, t\}$
- Edges - $E^k -$
  - $(s, v_i), (v_i, w_i), (w_i, t), (v_i, w_{i-1}), (v_1, t), (s, w_k)$
- Latency Functions
  - $A - e = (v_i, w_i)$
    \[ \ell^k_e(x) = 0 \]
  - $B - e = (v_i, w_{i-1}), (s, w_k), (v_1, t)$
    \[ \ell^k_e(x) = 1 \]
  - $C - e = (w_i, t), (s, v_{k-i+1})$
    \[ \ell_e(x) = \begin{cases} 
      0 & x = \frac{k}{k+1} \\
      i & x = 1 \\
      \text{cont., non-decr.} & \text{otherwise}
    \end{cases} \]
Proposition - For every integer \( n \geq 2 \), there is an instance \((G, r, \ell)\) in which \( G \) has \( n \) vertices and a subgraph \( H \) with

\[
L(G, r, \ell) = \left\lfloor \frac{n}{2} \right\rfloor \cdot L(H, r, \ell)
\]
# Hardness

- **Theorem** - Assuming $P \neq NP$, for every $\varepsilon > 0$ there is no $(\lfloor n/2 \rfloor - \varepsilon)$-approximation algorithm for General Latency Network Design.

- **Corollary** - Given an instance $(G, r, \ell)$ with general latency functions that is paradox-free or paradox-ridden, it is NP-hard to decide whether or not $(G, r, \ell)$ is paradox-ridden.
Now consider polynomial latency functions

\[ \ell(x) = a_p x^p + a_{p-1} x^{p-1} + \cdots + a_0 \quad \text{with} \quad a_i \geq 0. \]

Compare Nash flow to any feasible flow.

Proposition - There is a constant \( c_1 > 0 \) such that for every \( p \geq 2 \) and every instance \( (G, r, \ell) \) with polynomial latency functions of degree \( p \) that admits a Nash flow \( f \) and a feasible flow \( f^* \),

\[ C(f) \leq c_1 \frac{p}{\ln p} \cdot C(f^*) \]
Conclusions

- Braess’ paradox is impossible to recognize efficiently.
- The effect is bounded with linear or polynomial latency functions.
- Effect can become unbounded for general continuous non-decreasing latency functions.
- Important to recognize that effect of Braess’ Paradox is unbounded.