

Some
computational
Issues in Nash
Equilibria and the
Routing Game



Fundamental computational issues concerned with NE



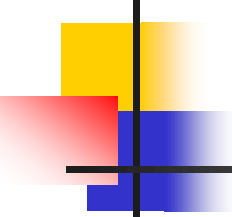
1. Finding (efficiently) a **mixed/pure** (if any) NE
2. Establishing the **quality** of a NE, as compared to a **cooperative system**, namely a system in which agents can collaborate (recall the *Prisoner's Dilemma*)
3. In a repeated game, establishing **whether** and in **how many** steps the system will eventually **converge** to a NE (recall the *Battle of the Sexes*)
4. **Verifying** that a strategy profile is a NE, **approximating** a NE, NE in **resource** (e.g., time, space, message size) **constrained** settings, **breaking** a NE by colluding, etc...

(interested in a Thesis, or even in a PhD?)



Finding a NE in mixed strategies

- How do we select the correct probability distribution? It looks like a problem in the continuous...
 - ...but it's not, actually! It can be shown that such a distribution can be found by selecting for each player a **best** possible subset of pure strategies (so-called **best support**), over which the probability distribution can actually be found by solving a system of algebraic equations (which are in general exponential in the number of players)
 - ⇒ In the practice, the problem can be solved by a simplex-like technique called the *Lemke-Howson algorithm*, which however is **exponential** in the worst case!
 - Remark:** Interestingly, for 2-player **zero-sum** games the solution can instead be found in polynomial time!
- It can be shown that there exist games for which finding a NE in mixed strategies is a **hard computational task** (but we cannot talk about **NP-hardness** here, since a solution is guaranteed to exist)



Finding a NE in pure strategies

- **Exhaustive search:** by definition, it is easy to see that an entry (p_1, \dots, p_N) of the payoff matrix is a NE if and only if p_i is the maximum i th element of the row $(p_1, \dots, p_{i-1}, \{p(s): s \in S_i\}, p_{i+1}, \dots, p_N)$, for each $i=1, \dots, N$.
- However, with N players, an explicit (i.e., in normal-form) representation of the payoff functions is exponential in N
- ⇒ Exhaustive search for finding a pure NE is then **exponential** in the number of players (even if it is still **polynomial** in the input size, but the normal-form representation needs not be a minimal-space representation of the input!).
- ⇒ Alternative cheaper methods are sought: for many games of interest, a NE can be found in poly-time w.r.t. to the number of players (e.g., by using the powerful **potential method**). But the question is: does there exist a general method which guarantees to be **always** polynomial in N ?
- As we will see, the answer is NO (if $P \neq NP$): in **pure strategies**, **finding a NE is NP-hard** for many games of interest (in this case, we can talk about NP-hardness since a solution is **not** guaranteed to exist)

On the quality of a NE



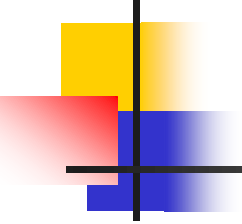
- How inefficient is a NE in comparison to an idealized situation in which the players would collaborate selflessly (in other words, the distributed system become **cooperative**), with the common goal of maximizing the overall **social welfare**, i.e., a **social-choice function C** which depends on the payoff of **all** the players (e.g., C is **the sum** of all the payoffs)?
- **Example:** in the *Prisoner's Dilemma* (PD) game, the DSE (and NE) incurs a total of **10** years in jail for the players. However, if the prisoners would cooperate by not implicating reciprocally, then they would stay a total of only **2** years in jail!

A worst-case perspective: the Price of Anarchy (PoA)

- **Definition (Koutsopias & Papadimitriou, 1999):** Given a game G and a **social-choice function** C , let S be the set of all NE. If the payoff represents a **cost** (resp., a **utility**) for a player, let OPT be the outcome of G **minimizing** (resp., **maximizing**) C . Then, the **Price of Anarchy (PoA)** of G w.r.t. C is

$$PoA_G(C) = \sup_{s \in S} \frac{C(s)}{C(OPT)} \left(\text{resp., } \inf_{s \in S} \frac{C(s)}{C(OPT)} \right)$$

- Notice that **PoA** ≥ 1
- **Example:** in the PD game, $PoA_{PD}(C) = 10/2 = 5$



A first case study for the existence and quality of a NE: selfish routing on the Internet

- Internet components are made up of heterogeneous nodes and links, and the network architecture is open-based and dynamic
- Internet users behave **selfishly**: they generate traffic, and their only goal is to download/upload data as fast as possible!
- But the more a link is used, the more it is slower, and there is no central authority “optimizing” the data flow...
- So, why does Internet eventually work in such a **jungle???**

The Internet routing game



Internet can be modelled by using game theory: it is a (congestion) game in which

players	→	users
strategies	→	paths over which users can route their traffic

Non-atomic Selfish Routing:

- There is a **large number** of (selfish) users generating a **large amount** of traffic;
- Every user controls an **infinitesimal fraction** of the traffic;
- The traffic of a user is routed over a **single path** in one shot.



Mathematical model (multicommodity flow network)

- A directed graph $G = (V, E)$ and a set of N players
- A set of *commodities*, i.e., source-sink pairs (s_i, t_i) , for $i=1, \dots, k$
- Each player is associated with a commodity
- Let N_i be the amount of players associated with (s_i, t_i) , for each $i=1, \dots, k$; then, the *rate* of traffic between s_i and t_i is $r_i = N_i/N$, with $0 \leq r_i \leq 1$ and $\sum_{i=1, \dots, k} r_i = 1$
- A set Π_i of paths in G between s_i and t_i for each $i=1, \dots, k$, and the corresponding set of all paths $\Pi = \bigcup_{i=1, \dots, k} \Pi_i$
- **Strategy for a player**: a **path** joining its commodity
- **Strategy profile**: a **flow vector** f specifying the rate of traffic f_p routed on each path $p \in \Pi$ (notice that $0 \leq f_p \leq 1$, and that for every $i=1, \dots, k$ we have $\sum_{p \in \Pi_i} f_p = r_i$)



Mathematical model (2)

- For each $e \in E$, the amount of flow absorbed by e w.r.t. f is $f_e = \sum_{P \in \Pi: e \in P} f_P$ (notice that $0 \leq f_e \leq 1$)
- For each edge e , a real-value latency function $l_e(x): [0, 1] \rightarrow \mathbb{R}^+$ of its absorbed flow x (this is a monotonically non-decreasing function which expresses how e gets congested when a fraction $0 \leq x \leq 1$ of the total flow f uses e)
- **Cost** of a player: the **latency** of its used **path** $P \in \Pi$:
$$c(P) = \sum_{e \in P} l_e(f_e)$$
- **Cost** (or **average latency**) of a **flow** f (**social-choice function**): $C(f) = \sum_{P \in \Pi} f_P \cdot c(P) = \sum_{P \in \Pi} f_P \cdot \sum_{e \in P} l_e(f_e) = \sum_{e \in E} f_e \cdot l_e(f_e)$

Observation: Notice that the game is not given in **normal form!**

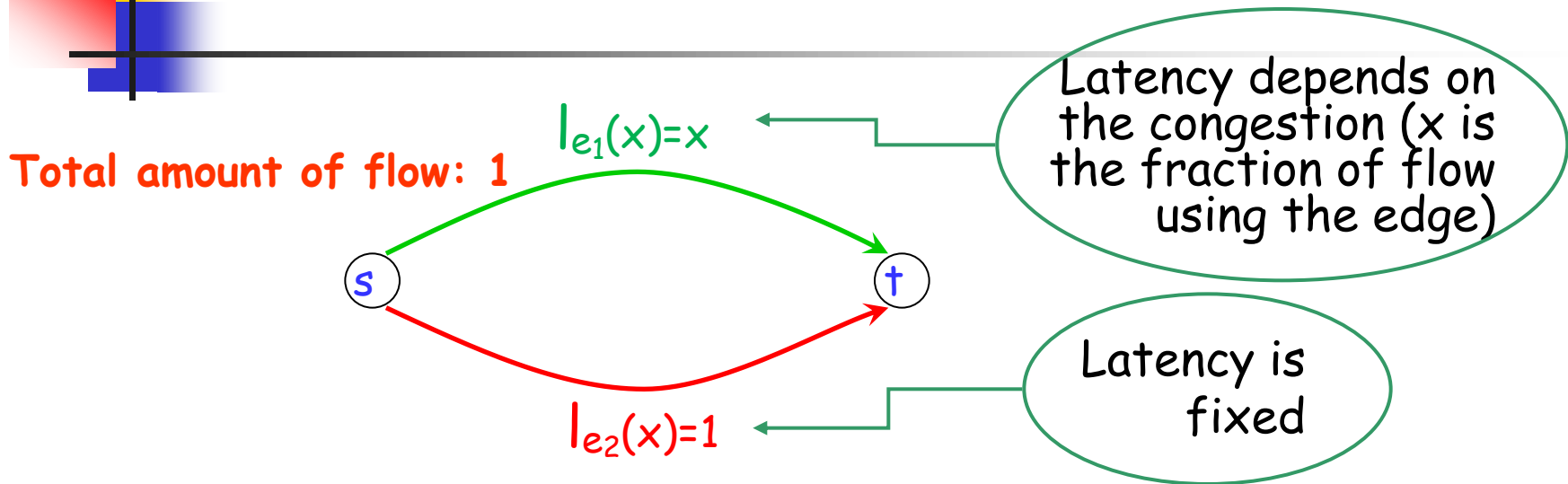


Flows and NE

Definition: A flow f^* is a **Nash flow** if no player can improve its cost (i.e., the **cost** of its used path) by changing unilaterally its path.

QUESTION: Given an instance $(G, s = ((s_1, t_1), \dots, (s_k, t_k)), r = (r_1, \dots, r_k), l = (l_{e_1}, \dots, l_{e_m}))$ of the non-atomic selfish routing game, **does it admit one or more Nash flows?** And in the positive case, what is the **PoA** of the game?

Example: Pigou's game [1920]



- Is there any Nash flow for this game?
YES! That in which all the flow travels on the upper edge \Rightarrow the cost of this flow is $C(f) = 1 \cdot l_{e_1}(1) + 0 \cdot l_{e_2}(0) = 1 \cdot 1 + 0 \cdot 1 = 1$
- Are there any other Nash flows? NO, and actually that one was a DSE!
- What is the PoA of this game? The optimal solution is the minimum of $C(x) = x \cdot x + (1-x) \cdot 1 \Rightarrow C(x) = x^2 - x + 1 \Rightarrow C'(x) = 2x - 1 \Rightarrow$ OPT for $C'(x) = 0$, i.e., $x = 1/2 \Rightarrow C(\text{OPT}) = 1/2 \cdot 1/2 + (1-1/2) \cdot 1 = 0.75$
 $\Rightarrow \text{PoA}(C) = 1/0.75 = 4/3$



Existence of a Nash flow

- **Theorem (Beckmann et al., 1956):** If for each edge e the function $x \cdot l_e(x)$ is **convex** (i.e., its graphic lies below the line segment joining any two points of the graphic) and **continuously differentiable** (i.e., its derivative exists at each point in its domain and is continuous), then the Nash flow of (G, s, r, l) exists and is unique, and is equal to the **optimal min-cost flow** of the following instance:

$$(G, s, r, \lambda(x) = [\int_0^x l(t) dt] / x).$$

- **Remark:** The **optimal min-cost flow** can be computed in polynomial time through convex programming methods.

Flows and Price of Anarchy

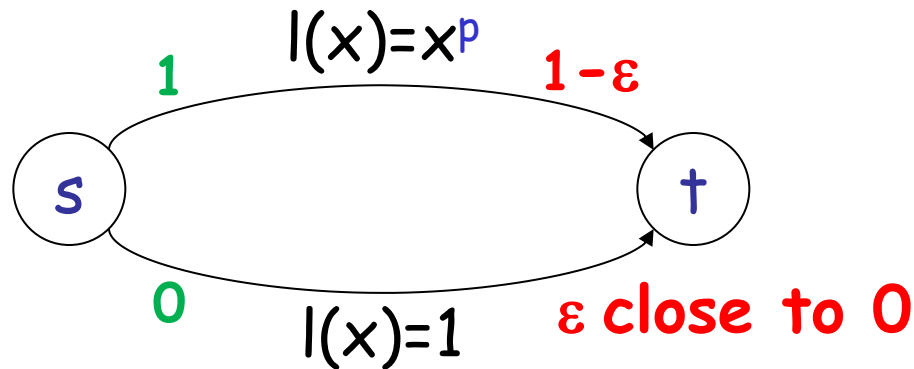


- Theorem 1: In a network with **linear** latency functions, the cost of a Nash flow is at most **4/3** times that of the min-cost flow \Rightarrow every **instance** of the non-atomic selfish routing satisfying this constraint has **PoA $\leq 4/3$** .
- Theorem 2: In a network with **degree-p polynomials** latency functions, the cost of a Nash flow is $O(p/\log p)$ times that of the min-cost flow.

(Roughgarden & Tardos, JACM'02)

A bad example for non-linear latencies

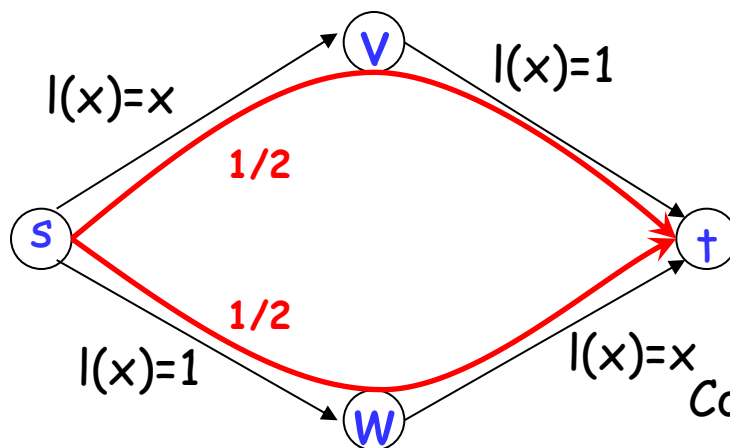
Assume $p \gg 1$



A **Nash flow** (of cost $C = 1 \cdot 1^p + 0 \cdot 1 = 1$) is arbitrarily more expensive than the **optimal flow** (of cost $C = (1 - \epsilon) \cdot (1 - \epsilon)^p + \epsilon \cdot 1 \approx 0$)

Improving the PoA: the Braess's paradox

- Does it help adding edges to improve the PoA?
- **NO!** Let's have a look at the **Braess Paradox (1968)**



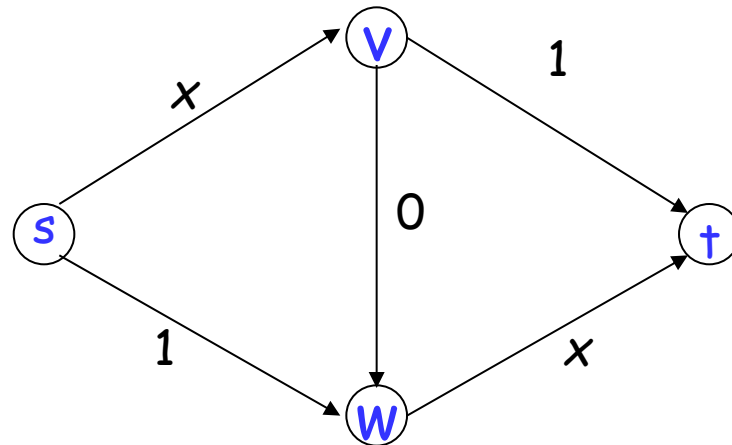
Let's assume that the players split. Then, the cost for each player (i.e., latency of a path) is $x+1 = 1/2+1 = 1.5$

$$\text{Cost of the flow} = 2 \cdot (1.5 \cdot 1/2) = 1.5$$

(notice this a (unique) NE and it is also an **optimal flow**)

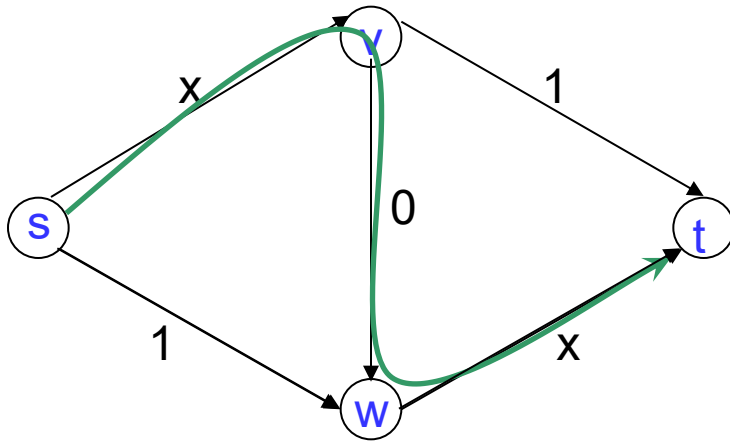
The Braess's paradox (2)

To reduce the cost of the flow, we try to add a no-latency road between v and w . Intuitively, this should not worsen things!



The Braess's paradox (3)

However, each user is tempted to change its route now, since the path $s \rightarrow v \rightarrow w \rightarrow t$ has less cost (indeed, $x \leq 1$)



If only **a single** user changes its route, then its cost decreases from **1.5** to approximately 1, i.e.:

$$c(s \rightarrow v \rightarrow w \rightarrow t) = x + 0 + x \approx 0.5 + 0.5 = 1$$

But the problem is that **all the users** will decide to change!

The Braess's paradox (4)

- So, the cost of the flow f that now entirely uses the path $s \rightarrow v \rightarrow w \rightarrow t$ is:

$$C(f) = 1 \cdot 1 + 1 \cdot 0 + 1 \cdot 1 = 2 > 1.5$$

- Even worse, this is a (unique) NE (the cost of the path $s \rightarrow v \rightarrow w \rightarrow t$ is 2, and the cost of the two paths not using (v,w) is also 2)!
- The optimal min-cost flow is equal to that we had before adding the new road and so, the PoA is

$$PoA = \frac{2}{1.5} = \frac{4}{3}$$

Notice it is $4/3$, as in the Pigou's example, and it is equal to the upper bound we gave for linear latency functions

Convergence towards a NE (in pure strategies games)



- Ok, we know that selfish routing is not so bad at its NE, but are we really sure this point of equilibrium will be eventually reached?
- **Convergence Time:** number of moves made by the players to reach a NE from an initial arbitrary state
- **Question:** *Is the convergence time (polynomially) bounded in the number of players?*



Convergence towards the Nash flow

- ☺ **Positive result:** If players obey to a *best response dynamics* (i.e., each player at each step greedily selects a strategy which **maximizes** its personal utility) then the **non-atomic selfish routing game** will converge to a NE. Moreover, for many instances (i.e., for prominent graph topologies and/or commodity specifications), the convergence time is **polynomial**.
- ☹ **Negative result:** However, there exist instances of the **non-atomic selfish routing game** for which the convergence time is **exponential** (under some mild assumptions).