Algorithmic Game Theory and Applications

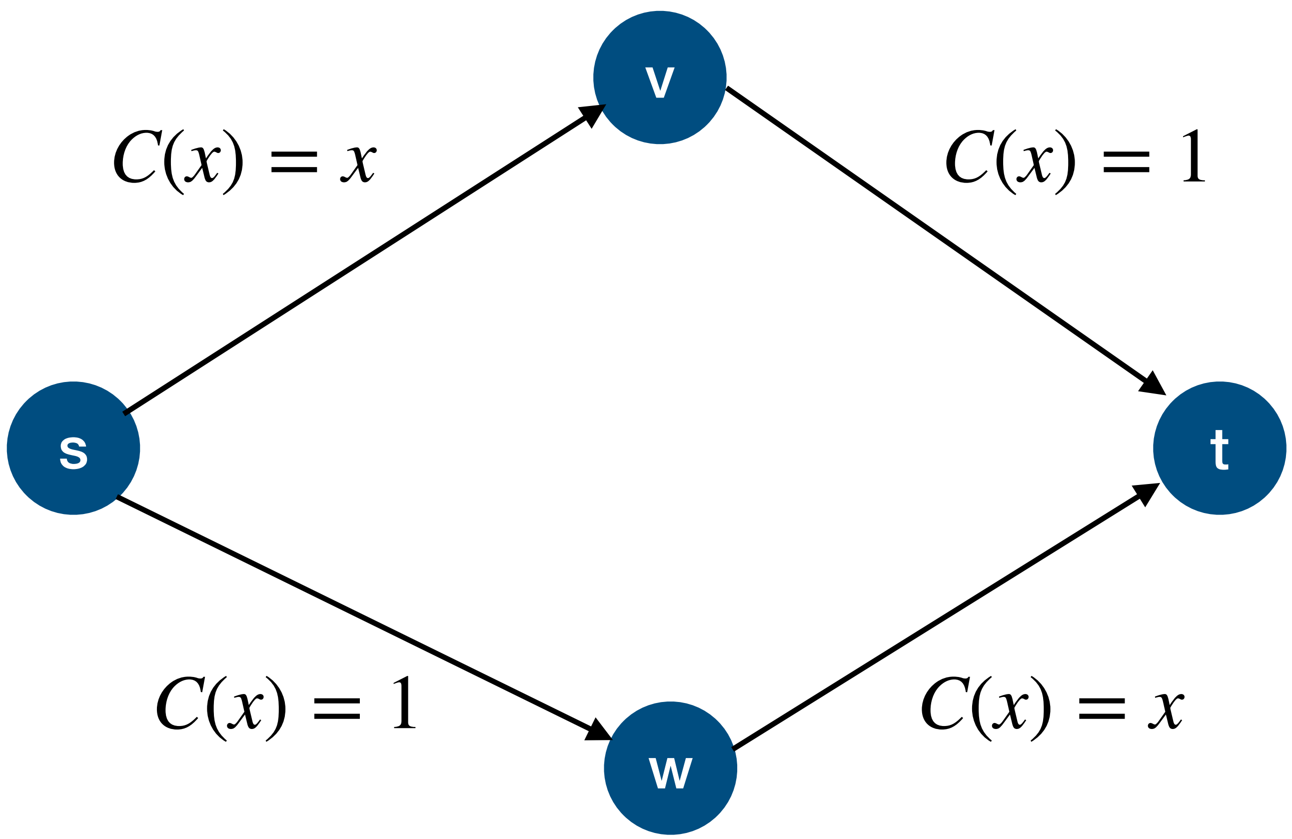
Congestion Games

Routing Traffic

We have *one unit of traffic* that goes from s to t.

There are two ways to get there, each contains a fast route and a slow route.

Every player controls a small part of the traffic, e.g., 1/100 of the whole unit.

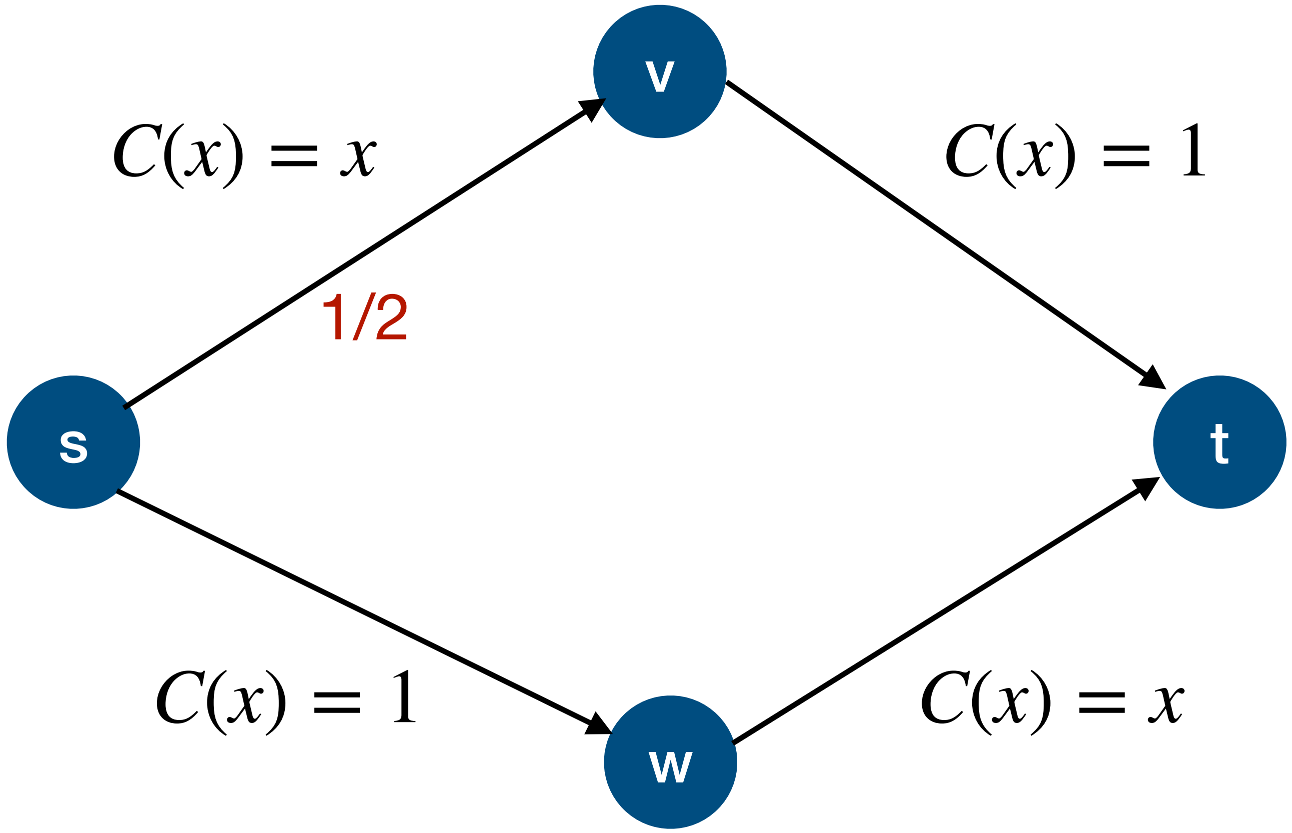


Routing Traffic

We have *one unit of traffic* that goes from s to t.

There are two ways to get there, each contains a fast route and a slow route.

Every player controls a small part of the traffic, e.g., 1/100 of the whole unit.

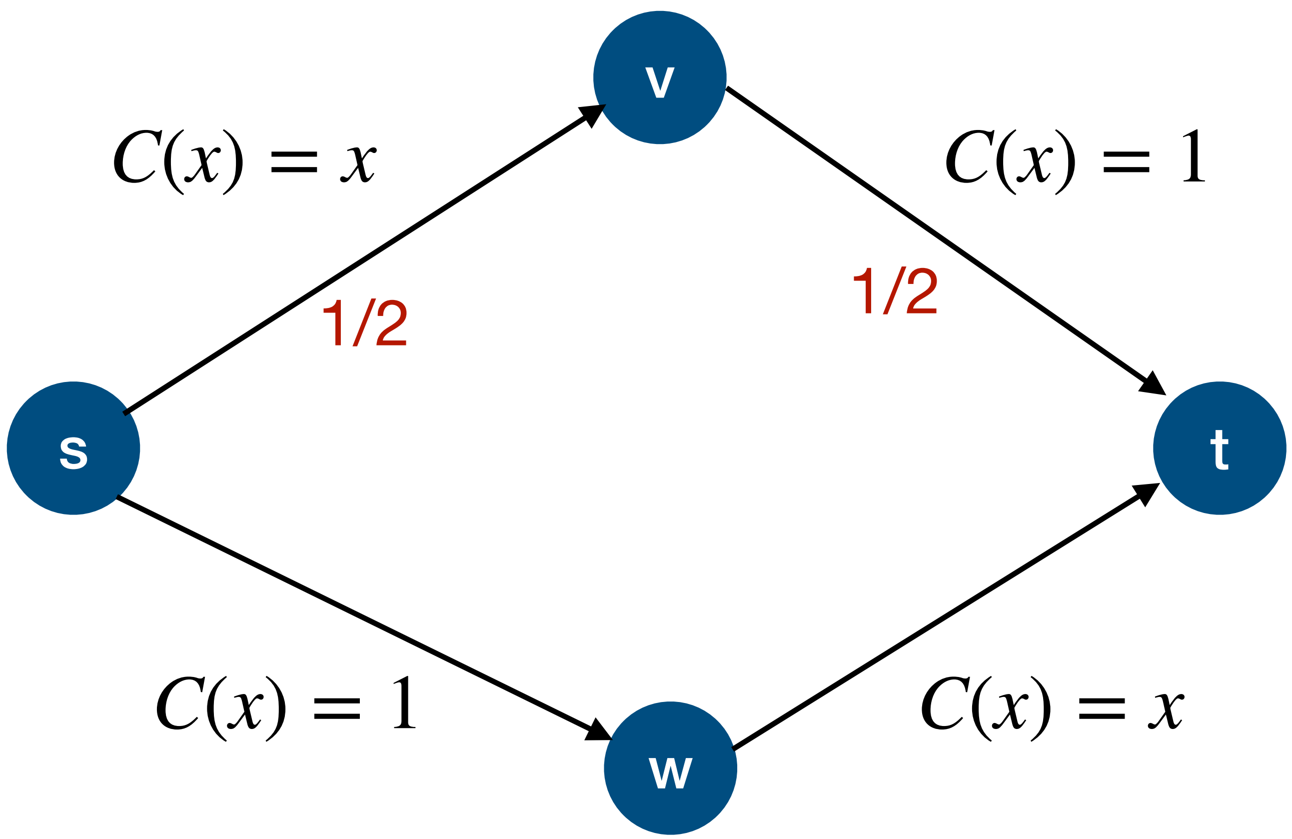


Routing Traffic

We have *one unit of traffic* that goes from s to t.

There are two ways to get there, each contains a fast route and a slow route.

Every player controls a small part of the traffic, e.g., 1/100 of the whole unit.

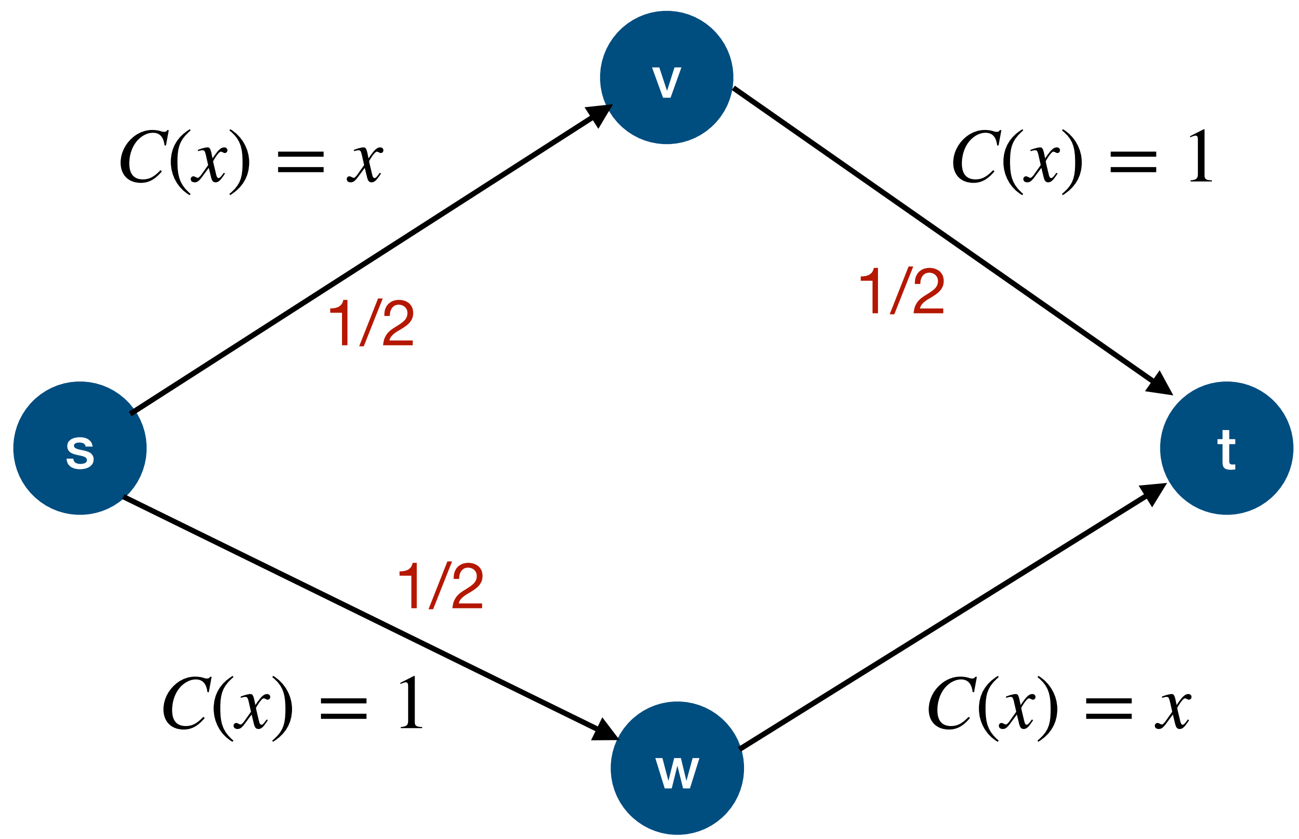


Routing Traffic

We have *one unit of traffic* that goes from s to t.

There are two ways to get there, each contains a fast route and a slow route.

Every player controls a small part of the traffic, e.g., 1/100 of the whole unit.

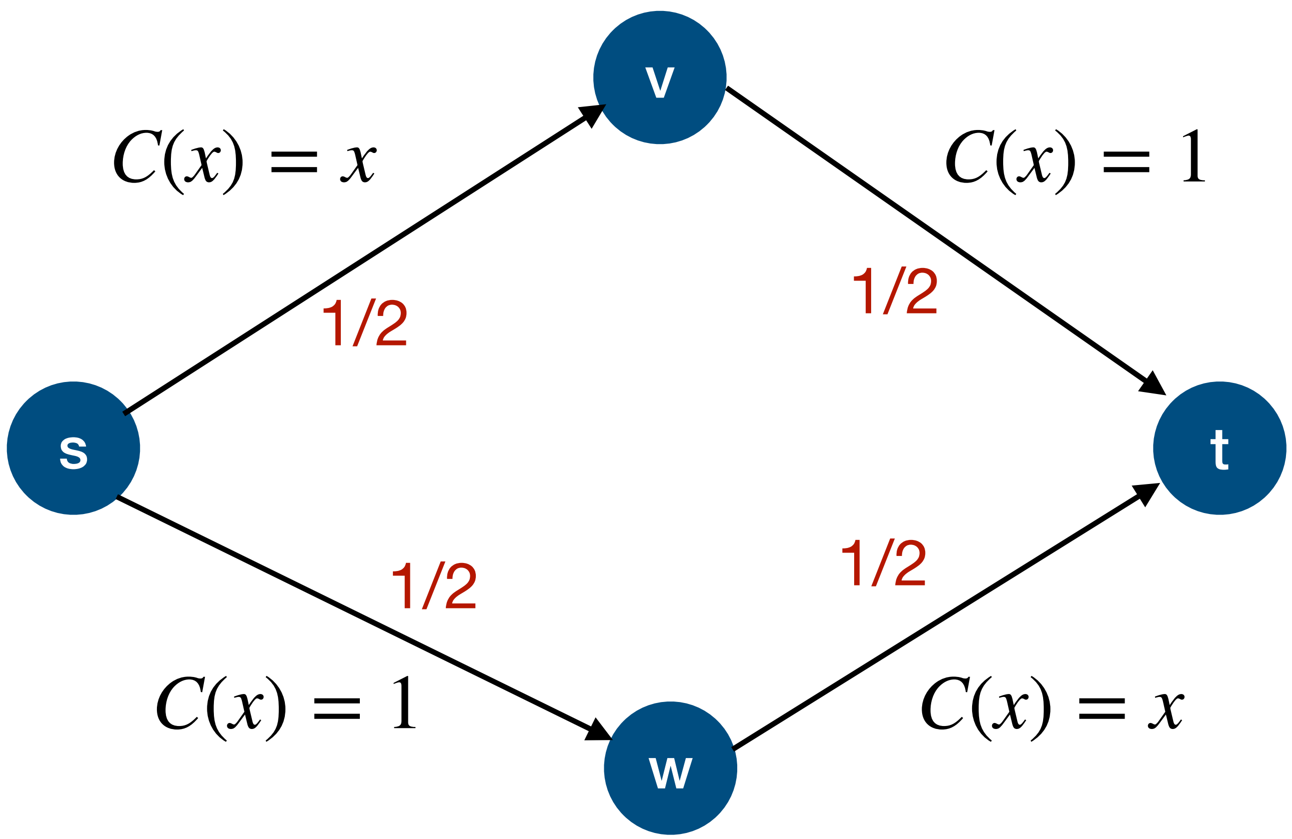


Routing Traffic

We have *one unit of traffic* that goes from s to t.

There are two ways to get there, each contains a fast route and a slow route.

Every player controls a small part of the traffic, e.g., 1/100 of the whole unit.

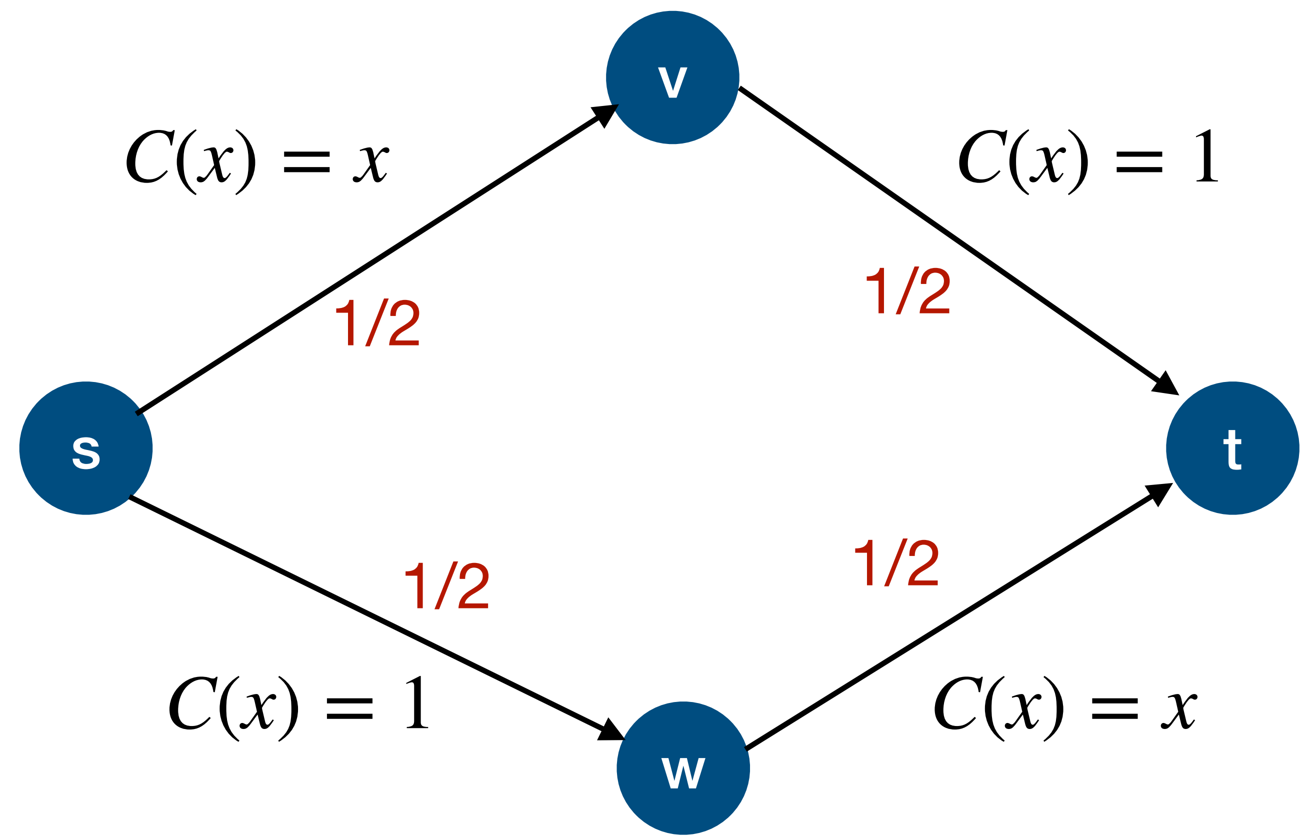


Routing Traffic

We have *one unit of traffic* that goes from s to t.

There are two ways to get there, each contains a fast route and a slow route.

Every player controls a small part of the traffic, e.g., 1/100 of the whole unit.



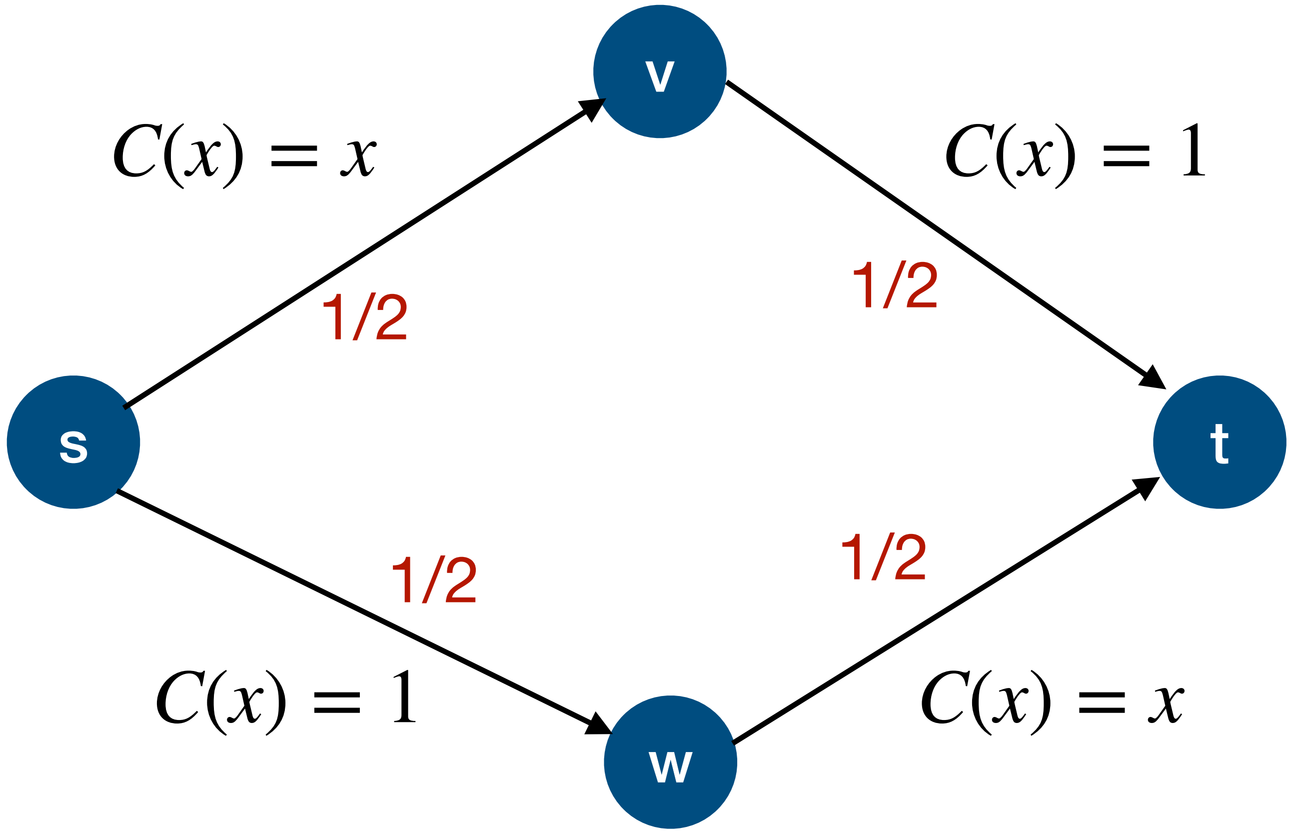
This is a Nash equilibrium!

Routing Traffic

We have *one unit of traffic* that goes from s to t.

There are two ways to get there, each contains a fast route and a slow route.

Every player controls a small part of the traffic, e.g., 1/100 of the whole unit.

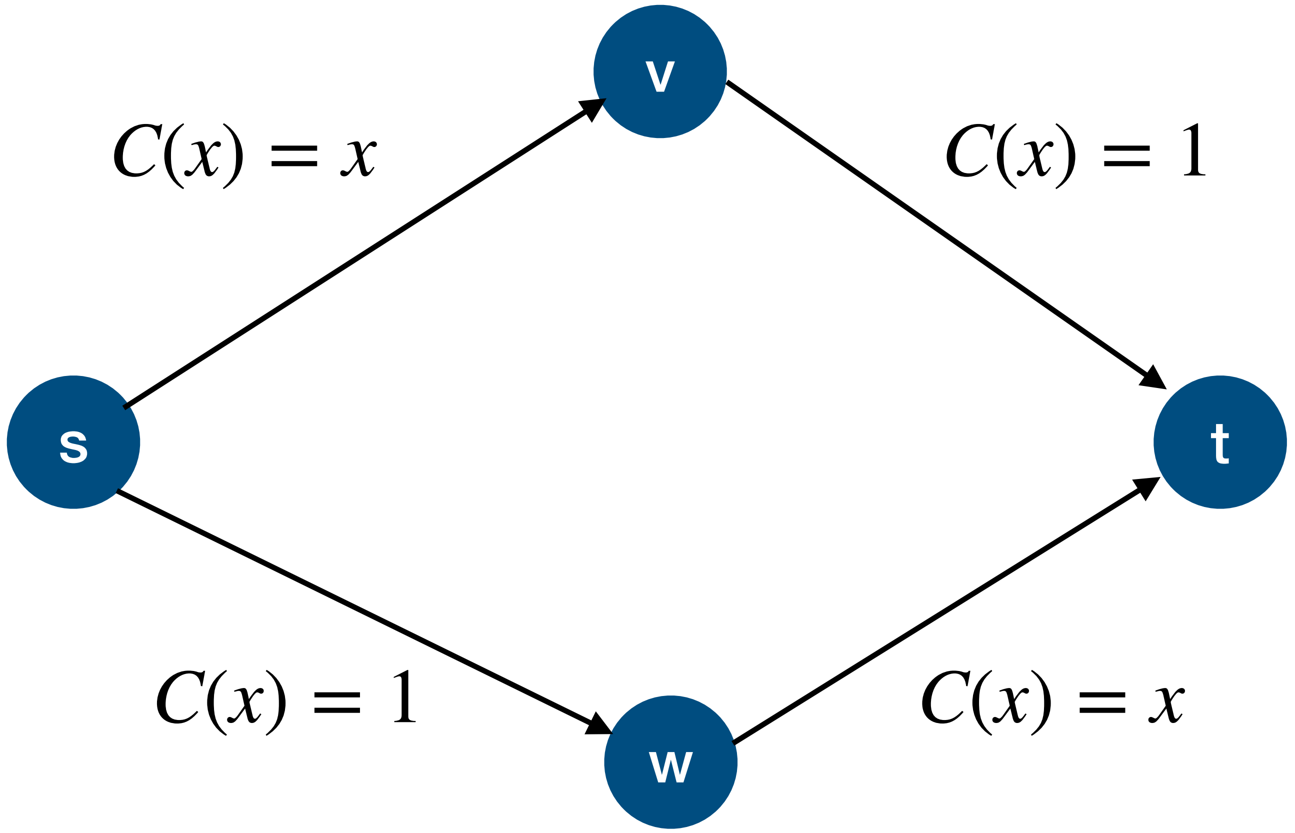


This is a Nash equilibrium!

Every commuter experiences a congestion of 1.5.

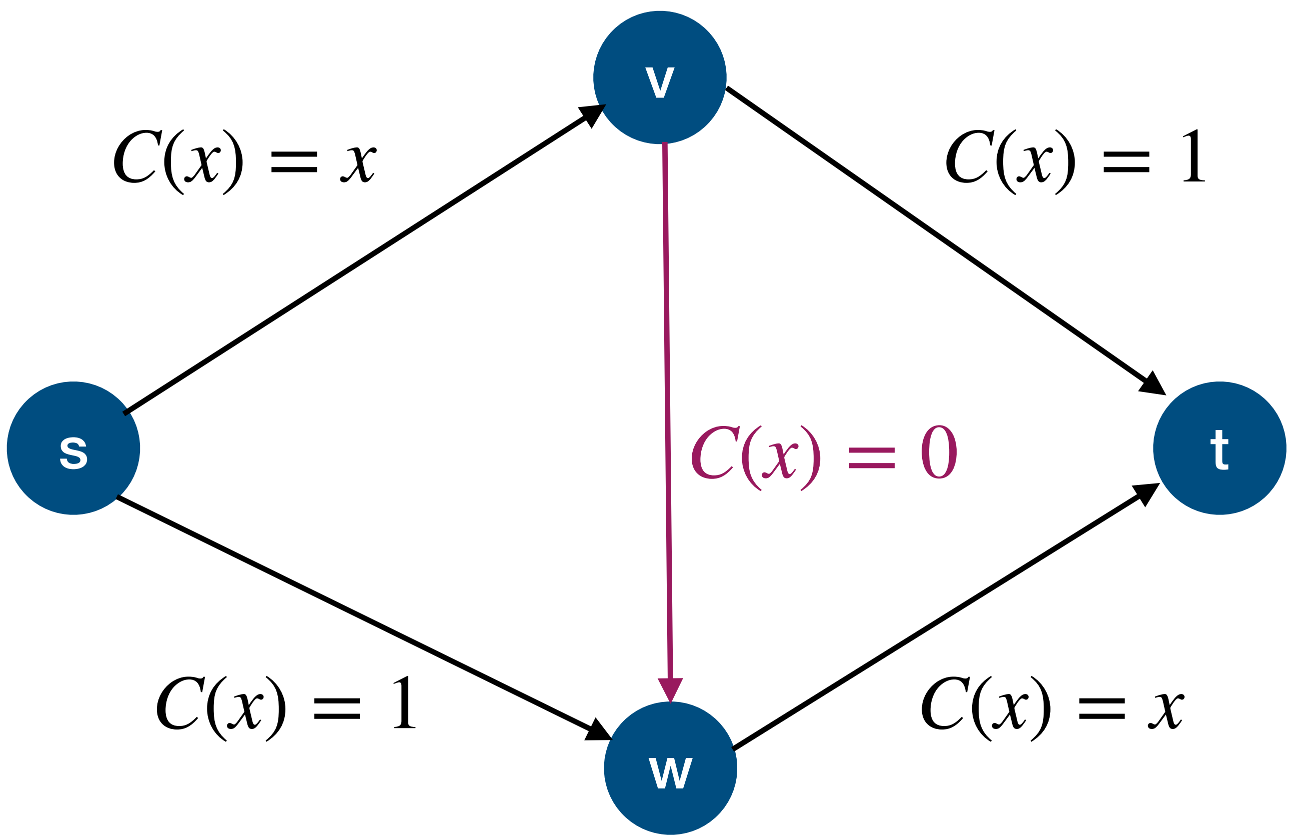
Routing Traffic

Now suppose that the government aims to prove the congestion on this network, by adding a super-fast multi-lane highway.



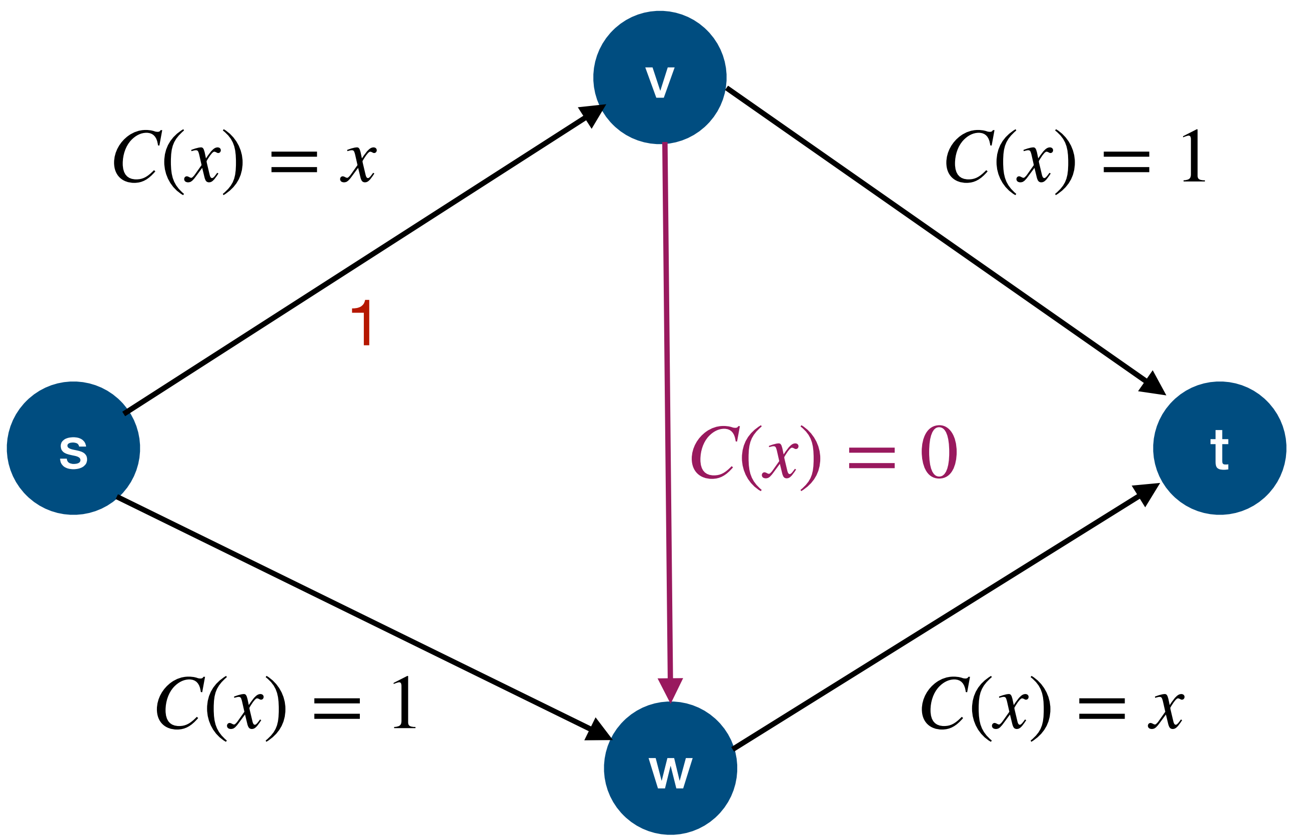
Routing Traffic

Now suppose that the government aims to prove the congestion on this network, by adding a super-fast multi-lane highway.



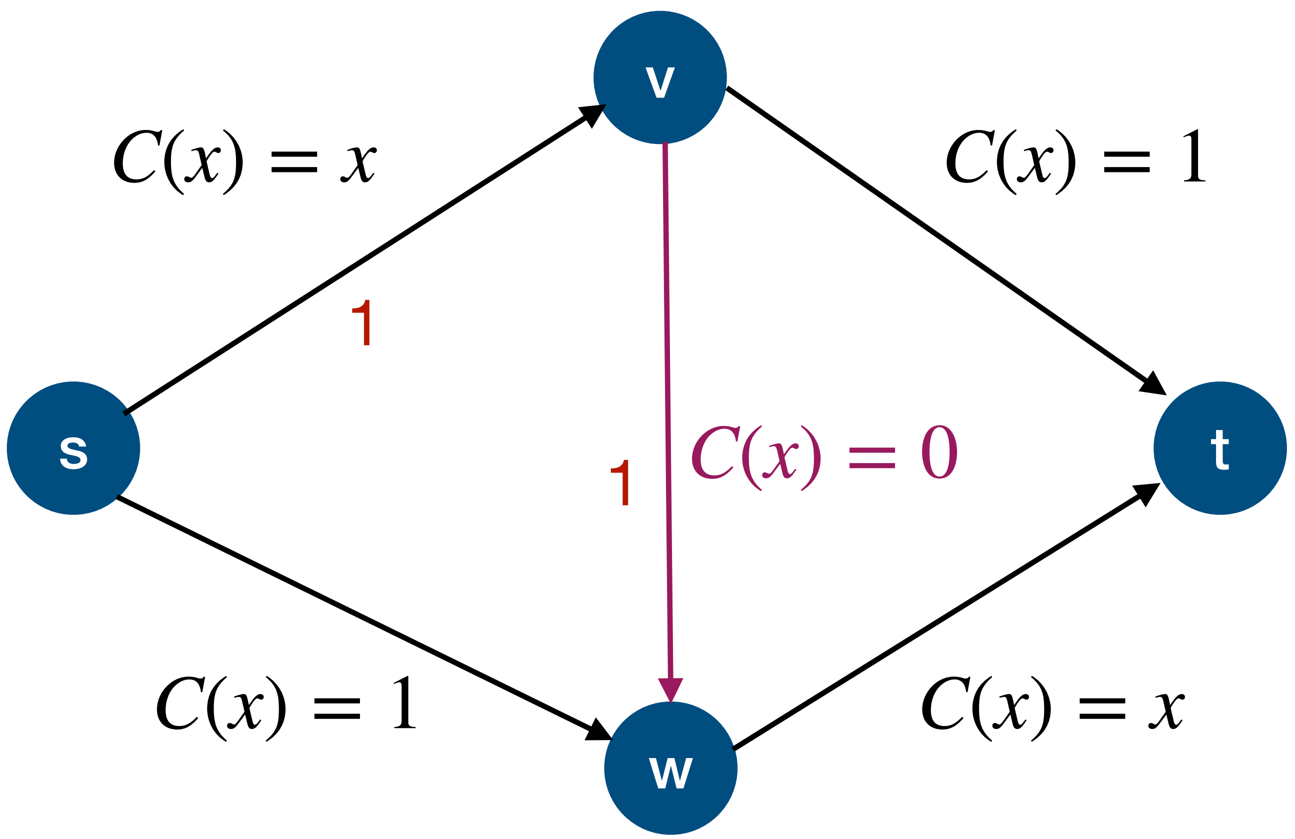
Routing Traffic

Now suppose that the government aims to prove the congestion on this network, by adding a super-fast multi-lane highway.



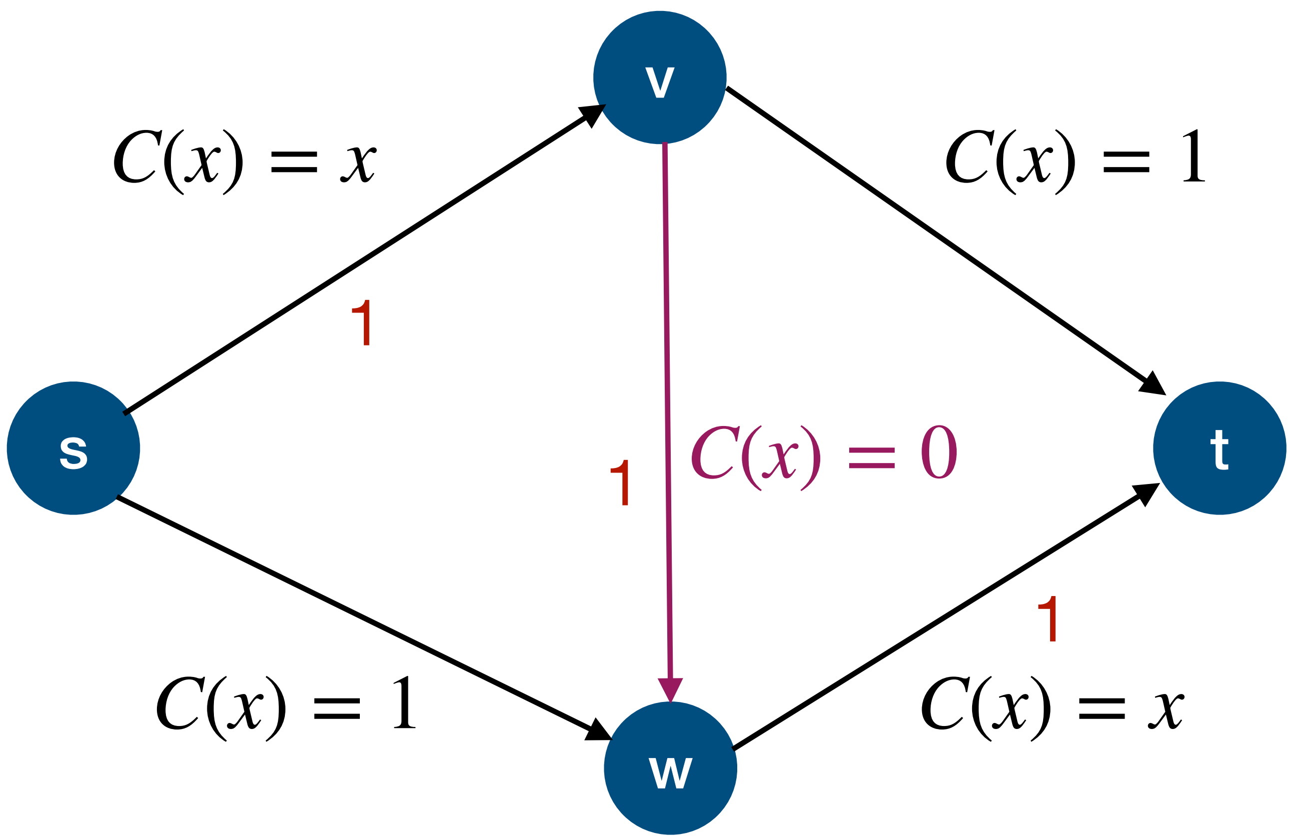
Routing Traffic

Now suppose that the government aims to prove the congestion on this network, by adding a super-fast multi-lane highway.



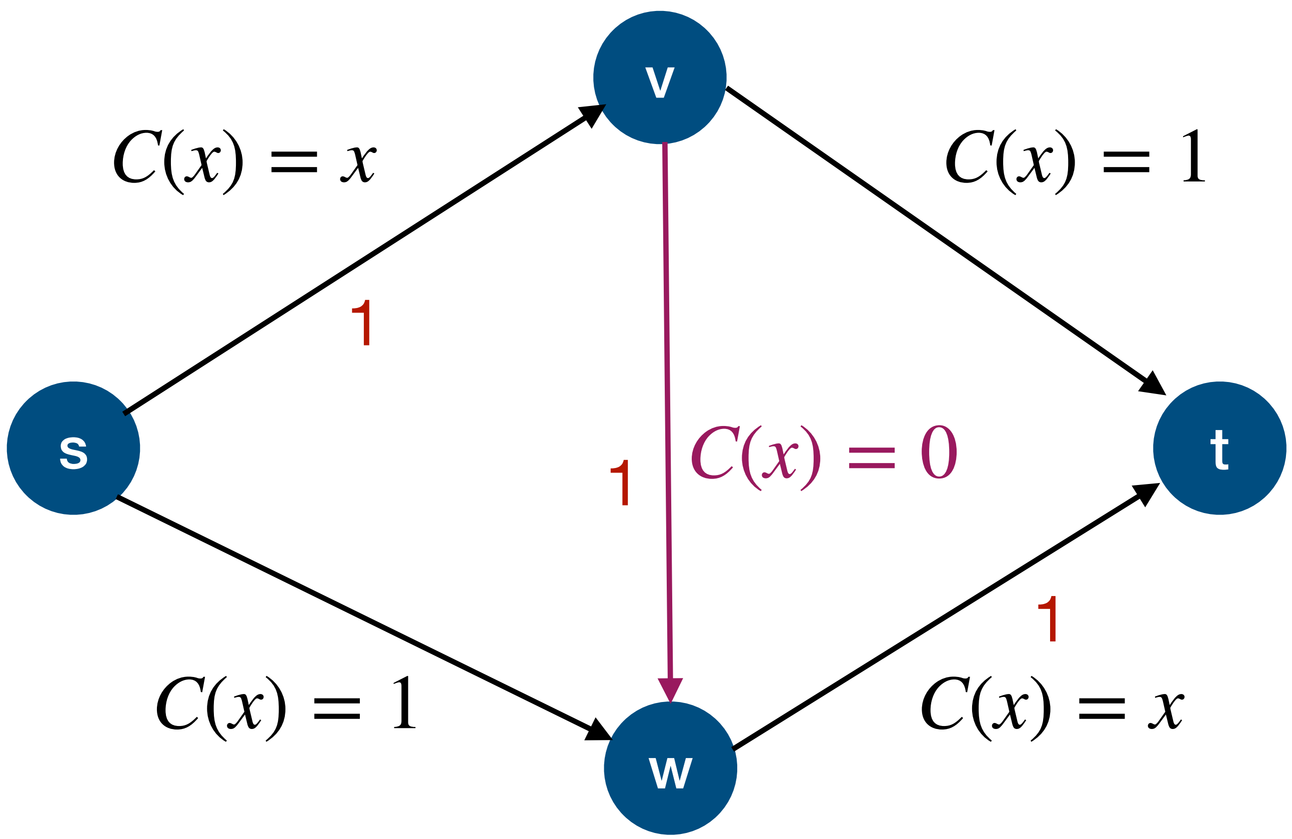
Routing Traffic

Now suppose that the government aims to prove the congestion on this network, by adding a super-fast multi-lane highway.



Routing Traffic

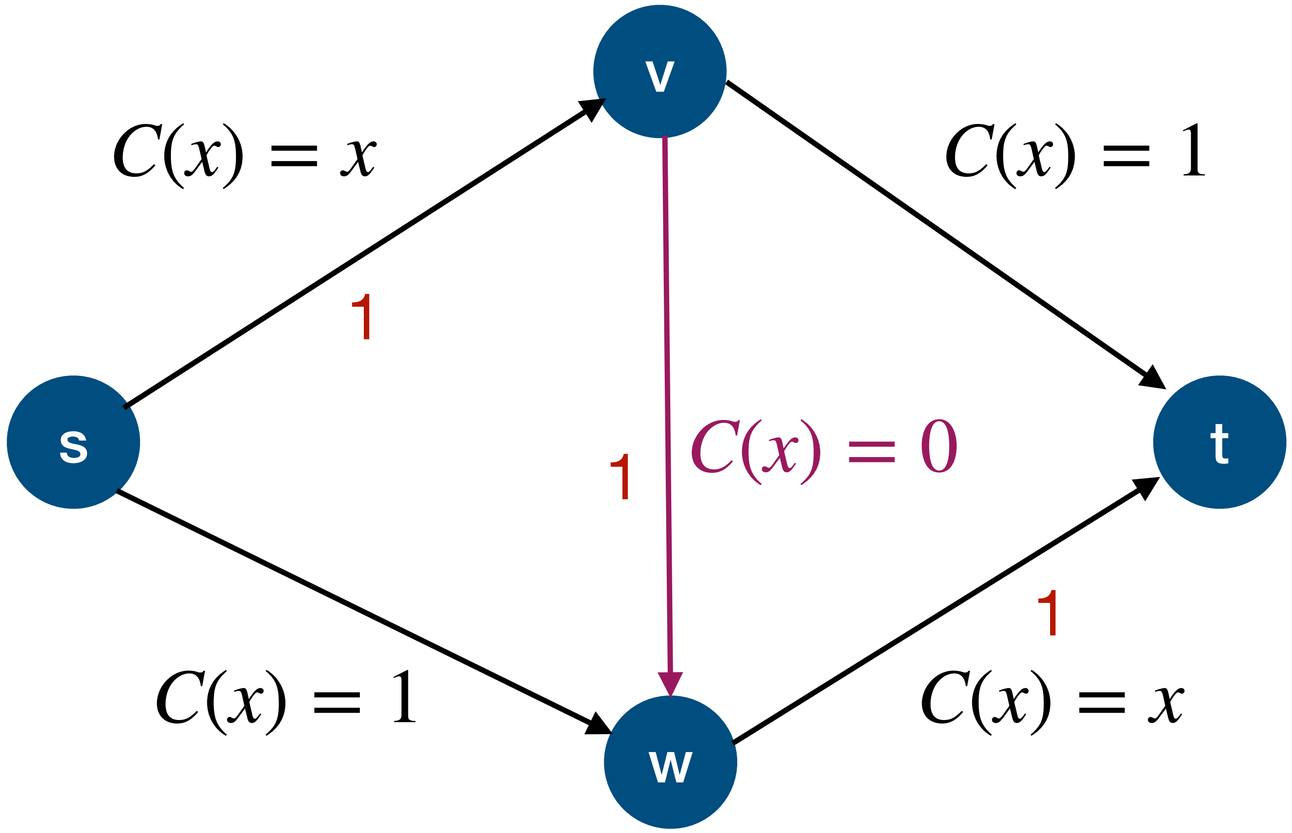
Now suppose that the government aims to prove the congestion on this network, by adding a super-fast multi-lane highway.



This is a Nash equilibrium!

Routing Traffic

Now suppose that the government aims to prove the congestion on this network, by adding a super-fast multi-lane highway.

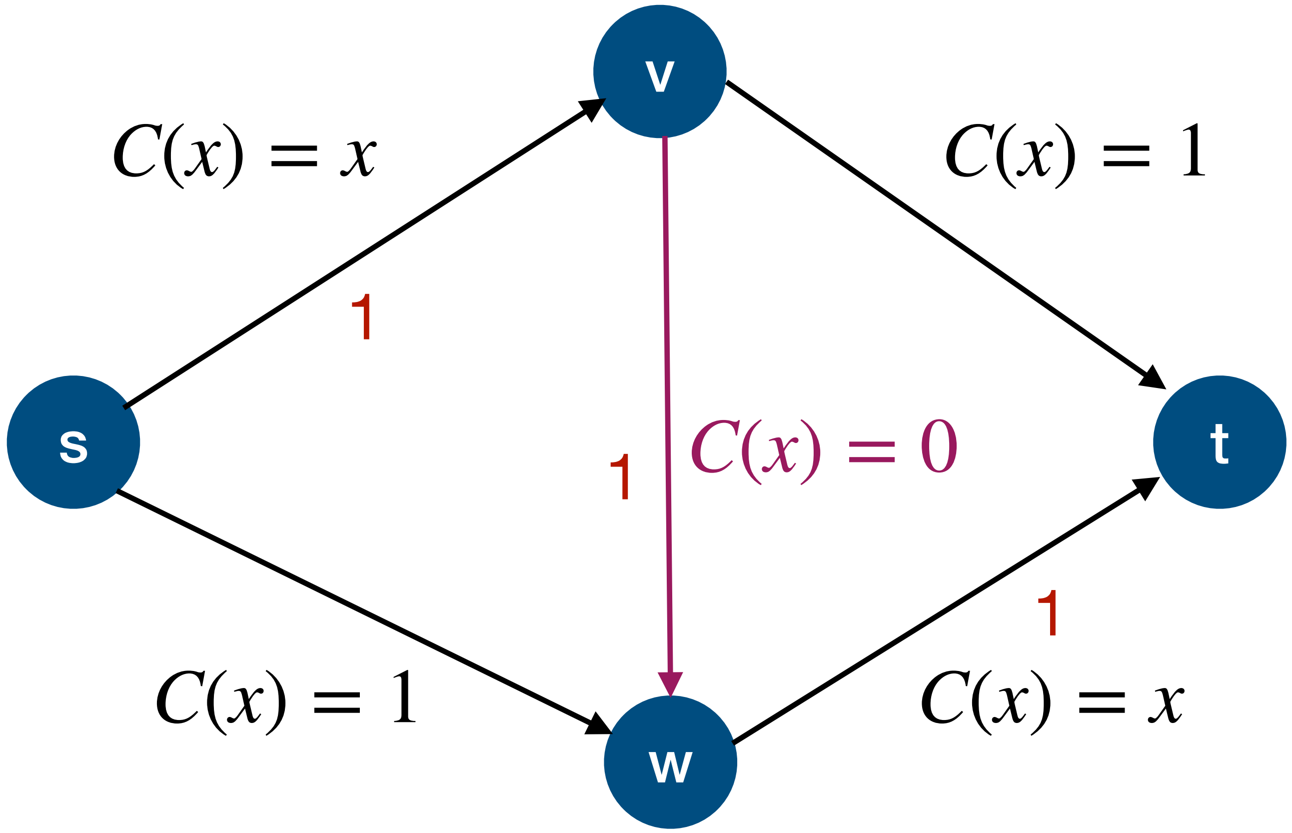


This is a Nash equilibrium!

Every commuter experiences a congestion of 2.

Routing Traffic

Now suppose that the government aims to prove the congestion on this network, by adding a super-fast multi-lane highway.



This is a Nash equilibrium!

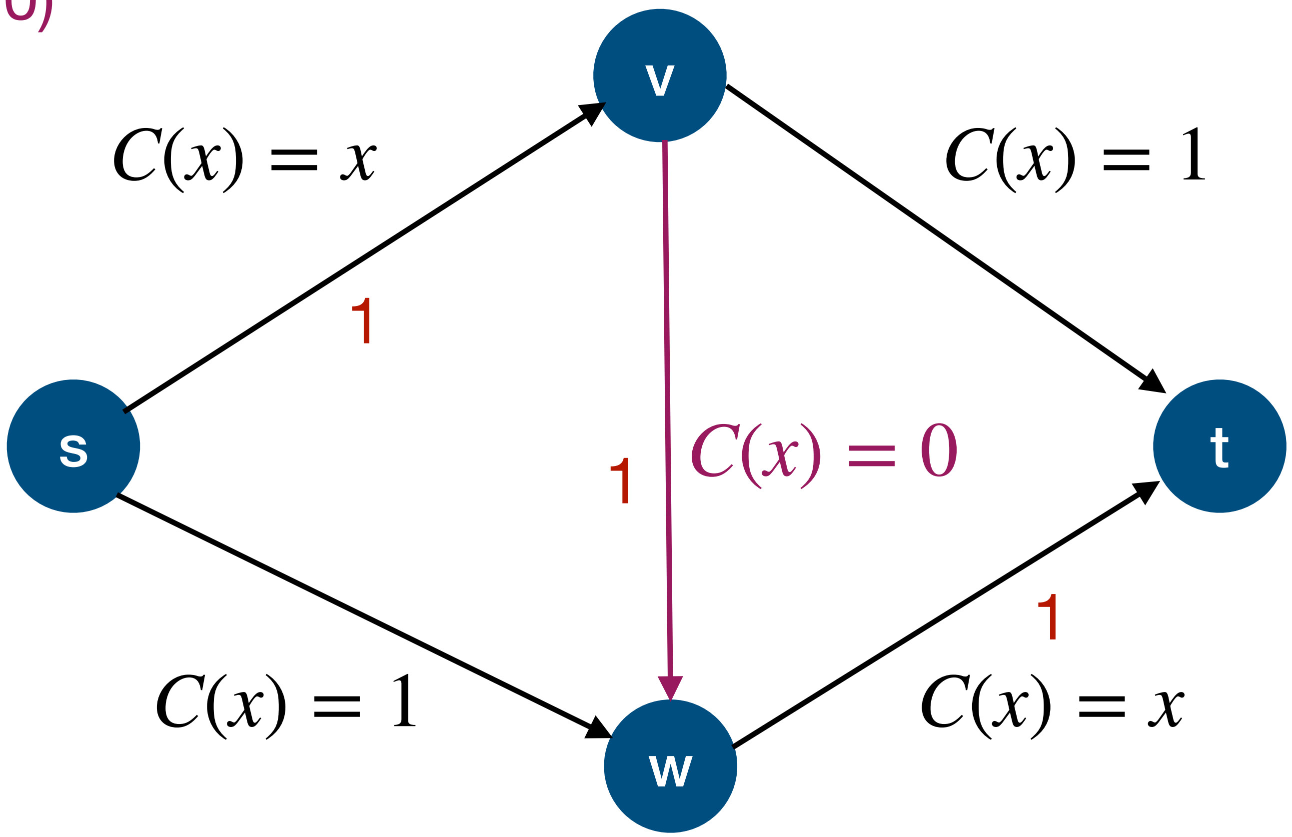
Every commuter experiences a congestion of 2.

Adding the high speed link made things worse!

Routing Traffic

Braess' Paradox (Pigou 1920)

Now suppose that the government aims to prove the congestion on this network, by adding a super-fast multi-lane highway.



This is a Nash equilibrium!

Every commuter experiences a congestion of 2.

Adding the high speed link made things worse!

Congestion Games

Congestion Games

Definition: An (atomic) congestion game is a tuple $G = (N, R, S, c)$ where

Congestion Games

Definition: An (atomic) congestion game is a tuple $G = (N, R, S, c)$ where

1. N is a set of n players.

Congestion Games

Definition: An (atomic) congestion game is a tuple $G = (N, R, S, c)$ where

1. N is a set of n players.

2. R is a set of m resources.

Congestion Games

Definition: An (atomic) congestion game is a tuple $G = (N, R, S, c)$ where

1. N is a set of n players.

2. R is a set of m resources.

3. $S = S_1 \times \ldots \times S_n$, where $S_i \subseteq 2^R \backslash \{0\}$ is the set of (pure) strategies of Player i.

Congestion Games

Definition: An (atomic) congestion game is a tuple $G = (N, R, S, c)$ where

1. N is a set of n players.

2. R is a set of m resources.

3. $S = S_1 \times \ldots \times S_n$, where $S_i \subseteq 2^R \backslash \{0\}$ is the set of (pure) strategies of Player i.

4. $c = (c_1, \ldots, c_r)$, where $c_k : \mathbb{N} \to \mathbb{R}$ is the cost function for resource $k \in R$.

Congestion Games

Definition: An (atomic) congestion game is a tuple $G = (N, R, S, c)$ where

1. N is a set of n players.

2. R is a set of m resources.

3. $S = S_1 \times \ldots \times S_n$, where $S_i \subseteq 2^R \backslash \{0\}$ is the set of (pure) strategies of Player i.

4. $c = (c_1, \ldots, c_r)$, where $c_k : \mathbb{N} \rightarrow \mathbb{R}$ is the cost function for resource $k \in R$.

In a congestion game, every player has a cost (or *disutility*) defined as

Congestion Games

Definition: An (atomic) congestion game is a tuple $G = (N, R, S, c)$ where

1. N is a set of n players.

2. R is a set of m resources.

3. $S = S_1 \times \ldots \times S_n$, where $S_i \subseteq 2^R \backslash \{0\}$ is the set of (pure) strategies of Player i.

4. $c = (c_1, \ldots, c_r)$, where $c_k : \mathbb{N} \rightarrow \mathbb{R}$ is the cost function for resource $k \in R$.

In a congestion game, every player has a cost (or *disutility*) defined as

$$\text{cost}_i(s) = \sum_{r \in R | r \in s_i} c_r(\#(r, s))$$

Congestion Games

Definition: An (atomic) congestion game is a tuple $G = (N, R, S, c)$ where

1. N is a set of n players.

2. R is a set of m resources.

3. $S = S_1 \times \ldots \times S_n$, where $S_i \subseteq 2^R \backslash \{0\}$ is the set of (pure) strategies of Player i.

4. $c = (c_1, \ldots, c_r)$, where $c_k : \mathbb{N} \to \mathbb{R}$ is the cost function for resource $k \in R$.

In a congestion game, every player has a cost (or *disutility*) defined as

$$\text{cost}_i(s) = \sum_{r \in R | r \in s_i} c_r(\#(r, s))$$

where $\#(r, s)$ is the number of players that took any strategy that involves resource r under strategy profile profile s.

Congestion Games

Definition: An (atomic) congestion game is a tuple $G = (N, R, S, c)$ where

1. N is a set of n players.

2. R is a set of m resources.

3. $S = S_1 \times \ldots \times S_n$, where $S_i \subseteq 2^R \backslash \{0\}$ is the set of (pure) strategies of Player i.

4. $c = (c_1, \ldots, c_r)$, where $c_k : \mathbb{N} \to \mathbb{R}$ is the cost function for resource $k \in R$.

In a congestion game, every player has a cost (or *disutility*) defined as

$$\text{cost}_i(s) = \sum_{r \in R | r \in s_i} c_r(\#(r, s))$$

where $\#(r, s)$ is the number of players that took any strategy that involves resource r under strategy profile profile s.

Observations and reasonable assumptions

In a congestion game, every player has a cost (or *disutility*) defined as

$$\text{cost}_i(s) = \sum_{r \in R | r \in s_i} c_r(\#(r, s))$$

where $\#(r, s)$ is the number of players that took any strategy that involves resource r under strategy profile profile s.

Observations and reasonable assumptions

In a congestion game, every player has a cost (or *disutility*) defined as

$$\text{cost}_i(s) = \sum_{r \in R | r \in s_i} c_r(\#(r, s))$$

where $\#(r, s)$ is the number of players that took any strategy that involves resource r under strategy profile profile s.

Observation: The cost of a player depends on how many players are using a resource, not on which those players are (*anonymity*).

Observations and reasonable assumptions

In a congestion game, every player has a cost (or *disutility*) defined as

$$\text{cost}_i(s) = \sum_{r \in R | r \in s_i} c_r(\#(r, s))$$

where $\#(r, s)$ is the number of players that took any strategy that involves resource r under strategy profile profile s.

Observation: The cost of a player depends on how many players are using a resource, not on which those players are (*anonymity*).

Reasonable Assumption: The cost c_r of a resource is *non-decreasing* in the number of players that use it (*monotonicity*).

Observations and reasonable assumptions

In a congestion game, every player has a cost (or *disutility*) defined as

$$\text{cost}_i(s) = \sum_{r \in R | r \in s_i} c_r(\#(r, s))$$

where $\#(r, s)$ is the number of players that took any strategy that involves resource r under strategy profile profile s.

Observation: The cost of a player depends on how many players are using a resource, not on which those players are (*anonymity*).

Reasonable Assumption: The cost c_r of a resource is *non-decreasing* in the number of players that use it (*monotonicity*).

But it is not unreasonable to not have this in some cases, e.g., the *El Farol Bar problem*.

Atomic Network Congestion Games

Atomic Network Congestion Games

Definition: An (atomic) network congestion game is a congestion game in which the resources are edges in a directed graph, and each player must choose a set of edges that forms a (simple) path from a given source s_i to a given sink t_i.

Atomic Network Congestion Games

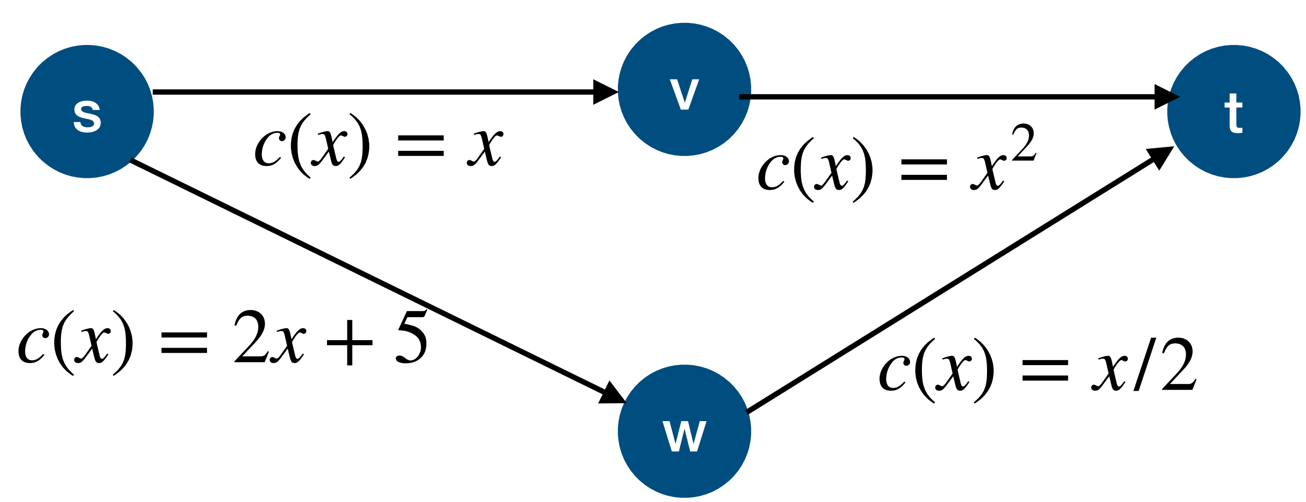
Definition: An (atomic) network congestion game is a congestion game in which the resources are edges in a directed graph, and each player must choose a set of edges that forms a (simple) path from a given source s_i to a given sink t_i.

On every edge there e is a cost function $c_e(x)$ which is a function of the number of players that have e in their chosen paths.

Atomic Network Congestion Games

Definition: An (atomic) network congestion game is a congestion game in which the resources are edges in a directed graph, and each player must choose a set of edges that forms a (simple) path from a given source s_i to a given sink t_i.

On every edge there e is a cost function $c_e(x)$ which is a function of the number of players that have e in their chosen paths.



Atomic Network Congestion Games

Definition: An (atomic) network congestion game is a congestion game in which the resources are edges in a directed graph, and each player must choose a set of edges that forms a (simple) path from a given source s_i to a given sink t_i.

On every edge there e is a cost function $c_e(x)$ which is a function of the number of players that have e in their chosen paths.

For example: $c_e(x)$ could be a linear function

$$c_e(x) = \alpha_e x + \beta_e$$

Best Response Dynamics

	L	C	R
U	-1, 1	1, -1	-2, -2
M	1, -1	-1, 1	-2, -2
D	-2, -2	-2, -2	2, 2

Best Response Dynamics

Recall: Best response := a strategy s_i that maximises the utility of Player i given the strategies s_{-i} of the other players.

	L	C	R
U	-1, 1	1, -1	-2, -2
M	1, -1	-1, 1	-2, -2
D	-2, -2	-2, -2	2, 2

Best Response Dynamics

Recall: Best response := a strategy s_i that maximises the utility of Player i given the strategies s_{-i} of the other players.

Start from an arbitrary (pure) strategy profile.

	L	C	R
U	-1, 1	1, -1	-2, -2
M	1, -1	-1, 1	-2, -2
D	-2, -2	-2, -2	2, 2

Best Response Dynamics

Recall: Best response := a strategy s_i that maximises the utility of Player i given the strategies s_{-i} of the other players.

Start from an arbitrary (pure) strategy profile.

If both players are best responding, then we are at a PNE.

	L	C	R
U	-1, 1	1, -1	-2, -2
M	1, -1	-1, 1	-2, -2
D	-2, -2	-2, -2	2, 2

Best Response Dynamics

Recall: Best response := a strategy s_i that maximises the utility of Player i given the strategies s_{-i} of the other players.

Start from an arbitrary (pure) strategy profile.

If both players are best responding, then we are at a PNE.

If there is one player that is not best responding, find a best response for the player and move to another strategy profile.

	L	C	R
U	-1, 1	1, -1	-2, -2
M	1, -1	-1, 1	-2, -2
D	-2, -2	-2, -2	2, 2

Best Response Dynamics

Recall: Best response := a strategy s_i that maximises the utility of Player i given the strategies s_{-i} of the other players.

Start from an arbitrary (pure) strategy profile.

If both players are best responding, then we are at a PNE.

If there is one player that is not best responding, find a best response for the player and move to another strategy profile.

Repeat until we reach an equilibrium.

	L	C	R
U	-1, 1	1, -1	-2, -2
M	1, -1	-1, 1	-2, -2
D	-2, -2	-2, -2	2, 2

Best Response Dynamics

Recall: Best response := a strategy s_i that maximises the utility of Player i given the strategies s_{-i} of the other players.

Start from an arbitrary (pure) strategy profile.

If both players are best responding, then we are at a PNE.

If there is one player that is not best responding, find a best response for the player and move to another strategy profile.

Repeat until we reach an equilibrium.

	L	C	R
U	-1, 1	1, -1	-2, -2
M	1, -1	-1, 1	-2, -2
D	-2, -2	-2, -2	2, 2

Best Response Dynamics

Recall: Best response := a strategy s_i that maximises the utility of Player i given the strategies s_{-i} of the other players.

Start from an arbitrary (pure) strategy profile.

If both players are best responding, then we are at a PNE.

If there is one player that is not best responding, find a best response for the player and move to another strategy profile.

Repeat until we reach an equilibrium.

	L	C	R
U	-1, 1	1, -1	-2, -2
M	1, -1	-1, 1	-2, -2
D	-2, -2	-2, -2	2, 2

Best Response Dynamics

Recall: Best response := a strategy s_i that maximises the utility of Player i given the strategies s_{-i} of the other players.

Start from an arbitrary (pure) strategy profile.

If both players are best responding, then we are at a PNE.

If there is one player that is not best responding, find a best response for the player and move to another strategy profile.

Repeat until we reach an equilibrium.

	L	C	R
U	-1, 1	1, -1	-2, -2
M	1, -1	-1, 1	-2, -2
D	-2, -2	-2, -2	2, 2

Best Response Dynamics

Recall: Best response := a strategy s_i that maximises the utility of Player i given the strategies s_{-i} of the other players.

Start from an arbitrary (pure) strategy profile.

If both players are best responding, then we are at a PNE.

If there is one player that is not best responding, find a best response for the player and move to another strategy profile.

Repeat until we reach an equilibrium.

	L	C	R
U	-1, 1	1, -1	-2, -2
M	1, -1	-1, 1	-2, -2
D	-2, -2	-2, -2	2, 2

Best Response Dynamics

Recall: Best response := a strategy s_i that maximises the utility of Player i given the strategies s_{-i} of the other players.

Start from an arbitrary (pure) strategy profile.

If both players are best responding, then we are at a PNE.

If there is one player that is not best responding, find a best response for the player and move to another strategy profile.

Repeat until we reach an equilibrium.

	L	C	R
U	-1, 1	1, -1	-2, -2
M	1, -1	-1, 1	-2, -2
D	-2, -2	-2, -2	2, 2

Best Response Dynamics

Recall: Best response := a strategy s_i that maximises the utility of Player i given the strategies s_{-i} of the other players.

Start from an arbitrary (pure) strategy profile.

If both players are best responding, then we are at a PNE.

If there is one player that is not best responding, find a best response for the player and move to another strategy profile.

Repeat until we reach an equilibrium.

	L	C	R
U	-1, 1	1, -1	-2, -2
M	1, -1	-1, 1	-2, -2
D	-2, -2	-2, -2	2, 2

Best Response Dynamics

Recall: Best response := a strategy s_i that maximises the utility of Player i given the strategies s_{-i} of the other players.

Start from an arbitrary (pure) strategy profile.

If both players are best responding, then we are at a PNE.

If there is one player that is not best responding, find a best response for the player and move to another strategy profile.

Repeat until we reach an equilibrium.

	L	C	R
U	-1, 1	1, -1	-2, -2
M	1, -1	-1, 1	-2, -2
D	-2, -2	-2, -2	2, 2

Best Response Dynamics

Recall: Best response := a strategy s_i that maximises the utility of Player i given the strategies s_{-i} of the other players.

Start from an arbitrary (pure) strategy profile.

If both players are best responding, then we are at a PNE.

If there is one player that is not best responding, find a best response for the player and move to another strategy profile.

Repeat until we reach an equilibrium.

	L	C	R
U	-1, 1	1, -1	-2, -2
M	1, -1	-1, 1	-2, -2
D	-2, -2	-2, -2	2, 2

Best Response Dynamics

The best response dynamics might not converge!

Recall: Best response := a strategy s_i that maximises the utility of Player i given the strategies s_{-i} of the other players.

Start from an arbitrary (pure) strategy profile.

If both players are best responding, then we are at a PNE.

If there is one player that is not best responding, find a best response for the player and move to another strategy profile.

Repeat until we reach an equilibrium.

	L	C	R
U	-1, 1	1, -1	-2, -2
M	1, -1	-1, 1	-2, -2
D	-2, -2	-2, -2	2, 2

Best Response Dynamics in Congestion Games

Best Response Dynamics in Congestion Games

Theorem (Rosenthal 1973): In any congestion game, the best response dynamics always converges to a pure Nash equilibrium.

Best Response Dynamics in Congestion Games

Theorem (Rosenthal 1973): In any congestion game, the best response dynamics always converges to a pure Nash equilibrium.

In particular, this implies that every congestion game has a pure Nash equilibrium.

Best Response Dynamics in Congestion Games

Theorem (Rosenthal 1973): In any congestion game, the best response dynamics always converges to a pure Nash equilibrium.

In particular, this implies that every congestion game has a pure Nash equilibrium.

The theorem also gives us an *algorithm* to find a PNE:

Best Response Dynamics in Congestion Games

Theorem (Rosenthal 1973): In any congestion game, the best response dynamics always converges to a pure Nash equilibrium.

In particular, this implies that every congestion game has a pure Nash equilibrium.

The theorem also gives us an *algorithm* to find a PNE:

- start from any arbitrary strategy profile,

Best Response Dynamics in Congestion Games

Theorem (Rosenthal 1973): In any congestion game, the best response dynamics always converges to a pure Nash equilibrium.

In particular, this implies that every congestion game has a pure Nash equilibrium.

The theorem also gives us an *algorithm* to find a PNE:

- start from any arbitrary strategy profile,
- run the best response dynamics until we reach a PNE.

Potential Games

Definition: A game is an (exact) potential game if there exists a potential function $\Phi : S_1 \times \ldots \times S_n \to \mathbb{R}$ such that for all $i \in N$, all $s_{-i} \in S_{-i}$ and $s_i, s_i' \in S_i$, we have that

$$\text{cost}_i(s_i, s_{-i}) - \text{cost}_i(s_i', s_{-i}) = \Phi(s_i, s_{-i}) - \Phi(s_i', s_{-i})$$

Potential Games

Definition: A game is an (exact) potential game if there exists a potential function $\Phi : S_1 \times \ldots \times S_n \to \mathbb{R}$ such that for all $i \in N$, all $s_{-i} \in S_{-i}$ and $s_i, s_i' \in S_i$, we have that

$$\text{cost}_i(s_i, s_{-i}) - \text{cost}_i(s_i', s_{-i}) = \Phi(s_i, s_{-i}) - \Phi(s_i', s_{-i})$$

Theorem (Rosenthal 1973): Every potential game has a pure Nash equilibrium.

Simple Proof

Theorem (Rosenthal 1973): Every potential game has a pure Nash equilibrium.

Simple Proof

Theorem (Rosenthal 1973): Every potential game has a pure Nash equilibrium.

Proof:

Simple Proof

Theorem (Rosenthal 1973): Every potential game has a pure Nash equilibrium.

Proof:

Let $s^* \in \arg\max_s \Phi(s)$, which implies that $\Phi(s^*) \geq \Phi(s')$ for any other strategy profile s'.

Simple Proof

Theorem (Rosenthal 1973): Every potential game has a pure Nash equilibrium.

Proof:

Let $s^* \in \arg\max_s \Phi(s)$, which implies that $\Phi(s^*) \geq \Phi(s')$ for any other strategy profile s'.

In particular, this also holds for $s' = (s_i', s_{-i})$. Since the game is a potential game, this means that $\text{cost}_i(s^*) \geq \text{cost}_i(s')$, and hence s^* is a pure Nash equilibrium.

Congestion Games are Potential Games

Congestion Games are Potential Games

Consider the following potential function:

Congestion Games are Potential Games

Consider the following potential function:

$$\Phi(s) = \sum_{r \in R} \sum_{j=1}^{\#(r,s)} c_r(j)$$

Congestion Games are Potential Games

Consider the following potential function:

$$\Phi(s) = \sum_{r \in R} \sum_{j=1}^{\#(r,s)} c_r(j)$$

Recall:

Congestion Games are Potential Games

Consider the following potential function:

$$\Phi(s) = \sum_{r \in R} \sum_{j=1}^{\#(r,s)} c_r(j)$$

Recall:

- $\#(r, s)$ is the number of players that use resource r in the strategy profile s.

Congestion Games are Potential Games

Consider the following potential function:

$$\Phi(s) = \sum_{r \in R} \sum_{j=1}^{\#(r,s)} c_r(j)$$

Recall:

- $\#(r, s)$ is the number of players that use resource r in the strategy profile s.
- $c_r(j)$ is the cost of resource r when it is being used by j players.

Congestion Games are Potential Games

Let $s = (s_i, s_{-i})$ and $s' = (s'_i, s_{-i})$.

We have $\Phi(s) - \Phi(s') =$

Congestion Games are Potential Games

Let $s = (s_i, s_{-i})$ and $s' = (s'_i, s_{-i})$.

We have $\Phi(s) - \Phi(s') = \sum_{r \in R} \sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{r \in R} \sum_{j=1}^{\#(r,s')} c_r(j)$

Congestion Games are Potential Games

Let $s = (s_i, s_{-i})$ and $s' = (s'_i, s_{-i})$.

We have $\Phi(s) - \Phi(s') = \sum_{r \in R} \sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{r \in R} \sum_{j=1}^{\#(r,s')} c_r(j)$

$$= \sum_{r \in R} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right)$$

Congestion Games are Potential Games

Congestion Games are Potential Games

Partition set set of resources R into 4 different subsets:

Congestion Games are Potential Games

Partition set set of resources R into 4 different subsets:

- $R_1 = \{r \in R : r \notin s_i \wedge r \notin s_i'\}$:
  r is not used in either strategy

Congestion Games are Potential Games

Partition set set of resources R into 4 different subsets:

- $R_1 = \{r \in R : r \notin s_i \wedge r \notin s_i'\}$:
  r is not used in either strategy

- $R_2 = \{r \in R : r \in s_i \wedge r \in s_i'\}$:
  r is used in the both strategies

Congestion Games are Potential Games

Partition set set of resources R into 4 different subsets:

- $R_1 = \{r \in R : r \notin s_i \wedge r \notin s_i'\}$:
  r is not used in either strategy

- $R_2 = \{r \in R : r \in s_i \wedge r \in s_i'\}$:
  r is used in the both strategies

- $R_3 = \{r \in R : r \in s_i \wedge r \notin s_i'\}$:
  r is used in s_i but not in s_i'.

Congestion Games are Potential Games

Partition set set of resources R into 4 different subsets:

- $R_1 = \{r \in R : r \notin s_i \wedge r \notin s_i'\}$:
  r is not used in either strategy

- $R_2 = \{r \in R : r \in s_i \wedge r \in s_i'\}$:
  r is used in the both strategies

- $R_3 = \{r \in R : r \in s_i \wedge r \notin s_i'\}$:
  r is used in s_i but not in s_i'.

- $R_4 = \{r \in R : r \notin s_i \wedge r \in s_i'\}$:
  r is used in s_i' but not in s_i.

Congestion Games are Potential Games

Let $s = (s_i, s_{-i})$ and $s' = (s'_i, s_{-i})$.

We have $\Phi(s) - \Phi(s') = \sum_{r \in R} \sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{r \in R} \sum_{j=1}^{\#(r,s')} c_r(j)$

$$= \sum_{r \in R} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right) = \sum_{i=1}^{4} \sum_{r \in R_i} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right)$$

Congestion Games are Potential Games

- $R_1 = \{r \in R : r \notin s_i \wedge r \notin s_i'\}$:
  r is not used in either strategy

Congestion Games are Potential Games

- $R_1 = \{r \in R : r \notin s_i \wedge r \notin s_i'\}$:
  r is not used in either strategy

$$\sum_{r \in R_1} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right)$$

Congestion Games are Potential Games

- $R_1 = \{r \in R : r \notin s_i \wedge r \notin s_i'\}$:
  r is not used in either strategy

$$\sum_{r \in R_1} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right)$$

We have $\#(r, s) = \#(r, s')$ (why?)

Congestion Games are Potential Games

- $R_1 = \{r \in R : r \notin s_i \wedge r \notin s_i'\}$:
  r is not used in either strategy

$$\sum_{r \in R_1} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right) = 0$$

We have $\#(r, s) = \#(r, s')$ (why?)

Congestion Games are Potential Games

- $R_2 = \{r \in R : r \in s_i \wedge r \in s_i'\}$:
  r is used in the both strategies

Congestion Games are Potential Games

- $R_2 = \{r \in R : r \in s_i \wedge r \in s_i'\}$:
  r is used in the both strategies

$$\sum_{r \in R_2} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right)$$

Congestion Games are Potential Games

- $R_2 = \{r \in R : r \in s_i \wedge r \in s_i'\}$:
  r is used in the both strategies

$$\sum_{r \in R_2} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right)$$

Again, we have $\#(r, s) = \#(r, s')$

Congestion Games are Potential Games

- $R_2 = \{r \in R : r \in s_i \wedge r \in s'_i\}$:
  r is used in the both strategies

$$\sum_{r \in R_2} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right) = 0$$

Again, we have $\#(r, s) = \#(r, s')$

Congestion Games are Potential Games

- $R_3 = \{r \in R : r \in s_i \wedge r \notin s_i'\}$:
  r is used in s_i but not in s_i'.

Congestion Games are Potential Games

- $R_3 = \{r \in R : r \in s_i \wedge r \notin s_i'\}$:
  r is used in s_i but not in s_i'.

$$\sum_{r \in R_3} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right)$$

Congestion Games are Potential Games

- $R_3 = \{r \in R : r \in s_i \wedge r \notin s_i'\}$:
  r is used in s_i but not in s_i'.

$$\sum_{r \in R_3} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right)$$

Now we have $\#(r, s) = \#(r, s') + 1$ (why?)

Congestion Games are Potential Games

- $R_3 = \{r \in R : r \in s_i \wedge r \notin s_i'\}$:
  r is used in s_i but not in s_i'.

$$\sum_{r \in R_3} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right) = \sum_{r \in R_3} c_r(\#(r,s))$$

Now we have $\#(r,s) = \#(r,s') + 1$ (why?)

Congestion Games are Potential Games

- $R_4 = \{r \in R : r \notin s_i \wedge r \in s_i'\}$:
  r is used in s_i' but not in s_i.

Congestion Games are Potential Games

- $R_4 = \{r \in R : r \notin s_i \wedge r \in s_i'\}$:
  r is used in s_i' but not in s_i.

$$\sum_{r \in R_4} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right)$$

Congestion Games are Potential Games

- $R_4 = \{r \in R : r \notin s_i \wedge r \in s_i'\}$:
  r is used in s_i' but not in s_i.

$$\sum_{r \in R_4} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right)$$

Now we have $\#(r, s) = \#(r, s') - 1$

Congestion Games are Potential Games

- $R_4 = \{r \in R : r \notin s_i \wedge r \in s_i'\}$:
  r is used in s_i' but not in s_i.

$$\sum_{r \in R_4} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right) = - \sum_{r \in R_4} c_r(\#(r, s'))$$

Now we have $\#(r, s) = \#(r, s') - 1$

Congestion Games are Potential Games

Let $s = (s_i, s_{-i})$ and $s' = (s_i', s_{-i})$.

We have $\Phi(s) - \Phi(s') = \sum_{r \in R} \sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{r \in R} \sum_{j=1}^{\#(r,s')} c_r(j)$

$$= \sum_{r \in R} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right) = \sum_{i=1}^{4} \sum_{r \in R_i} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right)$$

$$= \sum_{r \in R_3} c_r(\#(r, s)) - \sum_{r \in R_4} c_r(\#(r, s'))$$

Congestion Games are Potential Games

Let $s = (s_i, s_{-i})$ and $s' = (s_i', s_{-i})$.

We have $\Phi(s) - \Phi(s') = \sum_{r \in R} \sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{r \in R} \sum_{j=1}^{\#(r,s')} c_r(j)$

$$= \sum_{r \in R} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right) = \sum_{i=1}^{4} \sum_{r \in R_i} \left(\sum_{j=1}^{\#(r,s)} c_r(j) - \sum_{j=1}^{\#(r,s')} c_r(j) \right)$$

$$= \sum_{r \in R_3} c_r(\#(r,s)) - \sum_{r \in R_4} c_r(\#(r,s'))$$

$$= \sum_{r \in R_3} c_r(\#(r,s)) + \sum_{r \in R_2} c_r(\#(r,s)) - \sum_{r \in R_4} c_r(\#(r,s')) - \sum_{r \in R_2} c_r(\#(r,s'))$$

Congestion Games

Definition: An (atomic) congestion game is a tuple $G = (N, R, S, c)$ where

1. N is a set of n players.

2. R is a set of r resources.

3. $S = S_1 \times \ldots \times S_n$, where $S_i \subseteq 2^R \backslash \{0\}$ is the set of (pure) strategies of Player i.

4. $c = (c_1, \ldots, c_r)$, where $c_k : \mathbb{N} \to \mathbb{R}$ is the cost function for resource $k \in R$.

In a congestion game, every player has a cost (or *disutility*) defined as

$$\text{cost}_i(s) = \sum_{r \in R | r \in s_i} c_r(\#(r, s))$$

where $\#(r, s)$ is the number of players that took any strategy that involves resource r under strategy profile profile s.

Congestion Games

Definition: An (atomic) congestion game is a tuple $G = (N, R, S, c)$ where

1. N is a set of n players.

2. R is a set of r resources.

3. $S = S_1 \times \ldots \times S_n$, where $S_i \subseteq 2^R \backslash \{0\}$ is the set of (pure) strategies of Player i.

4. $c = (c_1, \ldots, c_r)$, where $c_k : \mathbb{N} \to \mathbb{R}$ is the cost function for resource $k \in R$.

In a congestion game, every player has a cost (or *disutility*) defined as

$$\text{cost}_i(s) = \sum_{r \in R | r \in s_i} c_r(\#(r, s))$$

where $\#(r, s)$ is the number of players that took any strategy that involves resource r under strategy profile profile s.

Congestion Games are Potential Games

Let $s = (s_i, s_{-i})$ and $s' = (s_i', s_{-i})$.

We have $\Phi(s) - \Phi(s') =$

$$= \sum_{r \in R_3} c_r(\#(r, s)) + \sum_{r \in R_2} c_r(\#(r, s))$$

$$- \sum_{r \in R_4} c_r(\#(r, s')) - \sum_{r \in R_2} c_r(\#(r, s'))$$

Congestion Games are Potential Games

Let $s = (s_i, s_{-i})$ and $s' = (s_i', s_{-i})$.

We have $\Phi(s) - \Phi(s') =$

$$\left.= \sum_{r \in R_3} c_r(\#(r, s)) + \sum_{r \in R_2} c_r(\#(r, s)) \right\} \text{cost}_i(s)$$

$$- \sum_{r \in R_4} c_r(\#(r, s')) - \sum_{r \in R_2} c_r(\#(r, s'))$$

Congestion Games are Potential Games

Let $s = (s_i, s_{-i})$ and $s' = (s'_i, s_{-i})$.

We have $\Phi(s) - \Phi(s') =$

$$\left.= \sum_{r \in R_3} c_r(\#(r, s)) + \sum_{r \in R_2} c_r(\#(r, s)) \right\} \text{cost}_i(s)$$

$$\left.- \sum_{r \in R_4} c_r(\#(r, s')) - \sum_{r \in R_2} c_r(\#(r, s')) \right\} -\text{cost}_i(s')$$

Best Response Dynamics in Congestion Games

Theorem (Rosenthal 1973): In any congestion game, the best response dynamics always converges to a pure Nash equilibrium.

In particular, this implies that every congestion game has a pure Nash equilibrium.

The theorem also gives us an *algorithm* to find a PNE:

- start from any arbitrary strategy profile,
- run the best response dynamics until we reach a PNE.

Best Response Dynamics in Congestion Games

Theorem (Rosenthal 1973): In any congestion game, the best response dynamics always converges to a pure Nash equilibrium.

In particular, this implies that every congestion game has a pure Nash equilibrium.

The theorem also gives us an *algorithm* to find a PNE:

- start from any arbitrary strategy profile,
- run the best response dynamics until we reach a PNE.

How do we prove that the algorithm will terminate?

Termination

Termination

Whenever a player best responds, the player's utility is increased by $u_i(s_i', s_{-i}) - u_i(s_i, s_{-i}) = \alpha$.

Termination

Whenever a player best responds, the player's utility is increased by $u_i(s'_i, s_{-i}) - u_i(s_i, s_{-i}) = \alpha$.

Since this is a potential game, the potential function Φ is also increased by α.

Termination

Whenever a player best responds, the player's utility is increased by $u_i(s_i', s_{-i}) - u_i(s_i, s_{-i}) = \alpha$.

Since this is a potential game, the potential function Φ is also increased by α.

Recall: $\Phi(s) = \sum_{r \in R} \sum_{j=1}^{\#(r,s)} c_r(j)$

Termination

Whenever a player best responds, the player's utility is increased by $u_i(s_i', s_{-i}) - u_i(s_i, s_{-i}) = \alpha$.

Since this is a potential game, the potential function Φ is also increased by α.

Recall: $\Phi(s) = \sum_{r \in R} \sum_{j=1}^{\#(r,s)} c_r(j)$

The potential function is at least 0 be definition.

Termination

Whenever a player best responds, the player's utility is increased by $u_i(s_i', s_{-i}) - u_i(s_i, s_{-i}) = \alpha$.

Since this is a potential game, the potential function Φ is also increased by α.

Recall: $\Phi(s) = \sum_{r \in R} \sum_{j=1}^{\#(r,s)} c_r(j)$

The potential function is at least 0 be definition.

What is the maximum possible value of the potential function?

Termination

Whenever a player best responds, the player's utility is increased by
$u_i(s_i', s_{-i}) - u_i(s_i, s_{-i}) = \alpha$.

Since this is a potential game, the potential function Φ is also increased by α.

Recall: $\Phi(s) = \sum_{r \in R} \sum_{j=1}^{\#(r,s)} c_r(j)$

The potential function is at least 0 be definition.

What is the maximum possible value of the potential function?

$$m \cdot n \cdot \max_j c_j(n)$$

Termination

Whenever a player best responds, the player's utility is increased by
$u_i(s_i', s_{-i}) - u_i(s_i, s_{-i}) = \alpha$.

Since this is a potential game, the potential function Φ is also increased by α.

Recall: $\Phi(s) = \sum_{r \in R} \sum_{j=1}^{\#(r,s)} c_r(j)$

The potential function is at least 0 be definition.

What is the maximum possible value of the potential function?

$$m \cdot n \cdot \max_j c_j(n)$$

How much does the potential increase in each step?

Termination

Whenever a player best responds, the player's utility is increased by $u_i(s_i', s_{-i}) - u_i(s_i, s_{-i}) = \alpha$.

Since this is a potential game, the potential function Φ is also increased by α.

Recall: $\Phi(s) = \sum_{r \in R} \sum_{j=1}^{\#(r,s)} c_r(j)$

The potential function is at least 0 be definition.

What is the maximum possible value of the potential function?

$m \cdot n \cdot \max_j c_j(n)$

How much does the potential increase in each step?

Assuming that the costs are integers, by at least 1.

An algorithm for finding PNE of congestion games

The theorem also gives us an *algorithm* to find a PNE:

- start from any arbitrary strategy profile,
- run the best response dynamics until we reach a PNE.

An algorithm for finding PNE of congestion games

The theorem also gives us an *algorithm* to find a PNE:

- start from any arbitrary strategy profile,
- run the best response dynamics until we reach a PNE.

By the potential argument, the algorithm will terminate. But will it terminate in polynomial time?

An algorithm for finding PNE of congestion games

The theorem also gives us an *algorithm* to find a PNE:

- start from any arbitrary strategy profile,
- run the best response dynamics until we reach a PNE.

By the potential argument, the algorithm will terminate. But will it terminate in polynomial time?

<u>We would have to show two things:</u>

An algorithm for finding PNE of congestion games

The theorem also gives us an *algorithm* to find a PNE:

- start from any arbitrary strategy profile,
- run the best response dynamics until we reach a PNE.

By the potential argument, the algorithm will terminate. But will it terminate in polynomial time?

<u>We would have to show two things:</u>

- The best response of a player can be computed in polynomial time.

An algorithm for finding PNE of congestion games

The theorem also gives us an *algorithm* to find a PNE:

- start from any arbitrary strategy profile,
- run the best response dynamics until we reach a PNE.

By the potential argument, the algorithm will terminate. But will it terminate in polynomial time?

<u>We would have to show two things:</u>

- The best response of a player can be computed in polynomial time.
- The equilibrium will be found within a polynomial number of best response steps.

Congestion Games

Definition: An (atomic) congestion game is a tuple $G = (N, R, S, c)$ where

1. N is a set of n players.

2. R is a set of m resources.

3. $S = S_1 \times \ldots \times S_n$, where $S_i \subseteq 2^R \setminus \{0\}$ is the set of (pure) strategies of Player i.

4. $c = (c_1, \ldots, c_r)$, where $c_k : \mathbb{N} \to \mathbb{R}$ is the cost function for resource $k \in R$.

In a congestion game, every player has a cost (or *disutility*) defined as

$$\text{cost}_i(s) = \sum_{r \in R | r \in s_i} c_r(\#(r, s))$$

where $\#(r, s)$ is the number of players that took any strategy that involves resource r under strategy profile profile s.

Congestion Games

Definition: An (atomic) congestion game is a tuple $G = (N, R, S, c)$ where

1. N is a set of n players.

We can enumerate over these.

2. R is a set of m resources.

3. $S = S_1 \times \ldots \times S_n$, where $S_i \subseteq 2^R \backslash \{0\}$ is the set of (pure) strategies of Player i.

4. $c = (c_1, \ldots, c_r)$, where $c_k : \mathbb{N} \to \mathbb{R}$ is the cost function for resource $k \in R$.

In a congestion game, every player has a cost (or *disutility*) defined as

$$\text{cost}_i(s) = \sum_{r \in R | r \in s_i} c_r(\#(r, s))$$

where $\#(r, s)$ is the number of players that took any strategy that involves resource r under strategy profile profile s.

An algorithm for finding PNE of congestion games

The theorem also gives us an *algorithm* to find a PNE:

- start from any arbitrary strategy profile,
- run the best response dynamics until we reach a PNE.

By the potential argument, the algorithm will terminate. But will it terminate in polynomial time?

<u>We would have to show two things:</u>

- The best response of a player can be computed in polynomial time.
- The equilibrium will be found within a polynomial number of best response steps.

An algorithm for finding PNE of congestion games

The theorem also gives us an *algorithm* to find a PNE:

- start from any arbitrary strategy profile,
- run the best response dynamics until we reach a PNE.

By the potential argument, the algorithm will terminate. But will it terminate in polynomial time?

<u>We would have to show two things:</u>

- The best response of a player can be computed in polynomial time. ✓
- The equilibrium will be found within a polynomial number of best response steps.

An algorithm for finding PNE of congestion games

The theorem also gives us an *algorithm* to find a PNE:

- start from any arbitrary strategy profile,
- run the best response dynamics until we reach a PNE.

By the potential argument, the algorithm will terminate. But will it terminate in polynomial time?

<u>We would have to show two things:</u>

- The best response of a player can be computed in polynomial time. ✓
- The equilibrium will be found within a polynomial number of best response steps. ?

Termination

Whenever a player best responds, the player's utility is increased by
$u_i(s'_i, s_{-i}) - u_i(s_i, s_{-i}) = \alpha$.

Since this is a potential game, the potential function Φ is also increased by α.

Recall: $\Phi(s) = \sum_{r \in R} \sum_{j=1}^{\#(r,s)} c_r(j)$

The potential function is at least 0 be definition.

What is the maximum possible value of the potential function?

$m \cdot n \cdot \max_j c_j(n)$

How much does the potential increase in each step?

Assuming that the costs are integers, by at least 1.

Termination

The best response dynamics converges in at most $m \cdot n \cdot \max_j c_j(n)$ steps.

Termination

The best response dynamics converges in at most $m \cdot n \cdot \max_j c_j(n)$ steps.

Is this a polynomial time algorithm?

Representation

Representation

Let's say that $\beta = 5$.

Representation

Let's say that $\beta = 5$.

How do we "save" 5 in a computer, using only 0 and 1?

Representation

Let's say that $\beta = 5$.

How do we "save" 5 in a computer, using only 0 and 1?

Binary representation: $5_{10} = 101_2 \rightarrow 101$

Representation

Let's say that $\beta = 5$.

How do we "save" 5 in a computer, using only 0 and 1?

Binary representation: $5_{10} = 101_2 \rightarrow 101$

Unary representation: $5_{10} \rightarrow 11111$

Description of the game

Description of the game

How do we represent the input to the algorithm?

Description of the game

How do we represent the input to the algorithm?

- The number of players n and number of resources m are numbers that are given in binary.

Description of the game

How do we represent the input to the algorithm?

- The number of players n and number of resources m are numbers that are given in binary.
- The cost functions for each agent can be represented in space $O(m \cdot n \cdot \log \max_j r_j(n))$, where we represent the function $r_j(\cdot)$ using a binary representation.

Termination

The best response dynamics converges in at most $m \cdot n \cdot \max_j c_j(n)$ steps.

Is this a polynomial time algorithm?

Termination

The best response dynamics converges in at most $m \cdot n \cdot \max_j c_j(n)$ steps.

Is this a polynomial time algorithm?

It is not, as $\max_j c_j(n)$ is *exponential* in the size of the input.

Termination

The best response dynamics converges in at most
$m \cdot n \cdot \max_j c_j(n)$ steps.

Is this a polynomial time algorithm?

It is not, as $\max_j c_j(n)$ is *exponential* in the size of the input.

It is what we call a pseudopolynomial time algorithm, i.e., it runs in time which is polynomial in the unary representation of the input.

Termination

The best response dynamics converges in at most $m \cdot n \cdot \max_j c_j(n)$ steps.

Is this a polynomial time algorithm?

It is not, as $\max_j c_j(n)$ is *exponential* in the size of the input.

It is what we call a pseudopolynomial time algorithm, i.e., it runs in time which is polynomial in the unary representation of the input.

Intuition: If the cost functions are represented with fairly small numbers, then it is a fast algorithm.

An algorithm for finding PNE of congestion games

The theorem also gives us an *algorithm* to find a PNE:

- start from any arbitrary strategy profile,
- run the best response dynamics until we reach a PNE.

By the potential argument, the algorithm will terminate. But will it terminate in polynomial time?

We would have to show two things:

- The best response of a player can be computed in polynomial time. ✓
- The equilibrium will be found within a polynomial number of best response steps. ?

An algorithm for finding PNE of congestion games

The theorem also gives us an *algorithm* to find a PNE:

- start from any arbitrary strategy profile,
- run the best response dynamics until we reach a PNE.

By the potential argument, the algorithm will terminate. But will it terminate in polynomial time?

<u>We would have to show two things:</u>

- The best response of a player can be computed in polynomial time. ✔
- The equilibrium will be found within a polynomial number of best response steps. ✘

A different approach perhaps?

Could we design a polynomial time algorithm for finding PNE in congestion games, perhaps using a different approach?

A different approach perhaps?

Could we design a polynomial time algorithm for finding PNE in congestion games, perhaps using a different approach?

Not likely.

A different approach perhaps?

Could we design a polynomial time algorithm for finding PNE in congestion games, perhaps using a different approach?

Not likely.

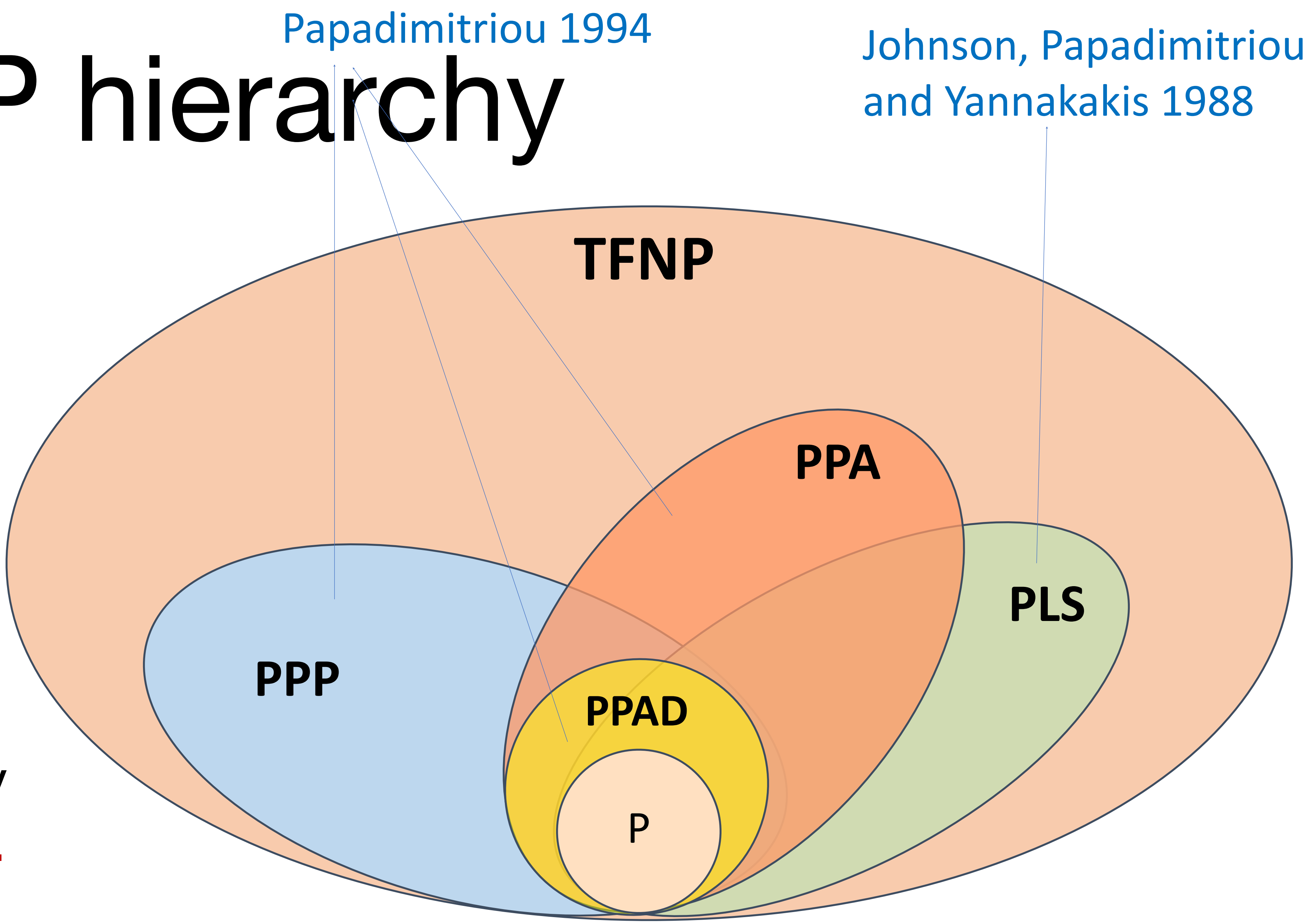
Theorem (Johnson et al. 1988, Fabrikant et al. 2004): Computing a PNE of a congestion game is PLS-complete.

The TFNP hierarchy

Define several subclasses of TFNP.

Show completeness results for those classes.

Approach initiated by (Papadimitriou 1994).



How about mixed equilibria?

Theorem (Johnson et al. 1988, Fabrikant et al. 2004): Computing a PNE of a congestion game is PLS-complete.

How about mixed equilibria?

Theorem (Johnson et al. 1988, Fabrikant et al. 2004): Computing a PNE of a congestion game is PLS-complete.

How about mixed equilibria of congestion games?

How about mixed equilibria?

Theorem (Johnson et al. 1988, Fabrikant et al. 2004): Computing a PNE of a congestion game is PLS-complete.

How about mixed equilibria of congestion games?

- They exist by Rosenthal (1971), since PNE are MNE.

How about mixed equilibria?

Theorem (Johnson et al. 1988, Fabrikant et al. 2004): Computing a PNE of a congestion game is PLS-complete.

How about mixed equilibria of congestion games?

- They exist by Rosenthal (1971), since PNE are MNE.
- They exist by Nash (1951), since congestion games are finite normal form games.

How about mixed equilibria?

Theorem (Johnson et al. 1988, Fabrikant et al. 2004): Computing a PNE of a congestion game is PLS-complete.

How about mixed equilibria of congestion games?

- They exist by Rosenthal (1971), since PNE are MNE.
- They exist by Nash (1951), since congestion games are finite normal form games.

Could they be easier to compute?

How about mixed equilibria?

Theorem (Johnson et al. 1988, Fabrikant et al. 2004): Computing a PNE of a congestion game is PLS-complete.

How about mixed equilibria of congestion games?

- They exist by Rosenthal (1971), since PNE are MNE.
- They exist by Nash (1951), since congestion games are finite normal form games.

Could they be easier to compute?

Theorem (Babichenko and Rubinstein 2021): Computing a MNE of a congestion game is PPAD $\cap$ PLS - complete.

The TFNP hierarchy

Define several subclasses of TFNP.

Show completeness results for those classes.

Approach initiated by (Papadimitriou 1994).