Algorithmic Game Theory

Applied Algorithms
PKU summer 2019

Tami Tamir

Game Theory - Introduction.

When we play, we have to make decisions.

The other players must respond to our decisions in their own decisions.

Game theory: A field of mathematics that deals with the construction and analysis of mathematical models to describe situations in which decision-making is required.

Not only games for entertainment.

Two-player binary-choice games (identical players)

The players need to select one out of two strategies. The game payoff matrix has the following form:

Alice \ Bob	Strategy 1	Strategy 2
Strategy 1	a, a	b, c
Strategy 2	c, b	d, d

In this profile, Alice will be paid c and Bob will be paid b.

Rock paper scissors

- A game for two players.
- Each player has three strategies.
- 2-player games can be describe by a 2-dim table (a payoff matrix):

			
	0 , 0	-1, 1	1 , -1
	1, -1	0 , 0	-1 , 1
	-1, 1	1 , -1	0 , 0

Zero-sum game with identical players

Nash Equilibrium

A set of strategic choices is a Nash equilibrium (NE) if each player is doing the best possible given what the other is doing.

- Rock-paper-scissors has no NE in pure strategies.
  - "If I know you play Rock, I'll play paper"
  - "If I know you play Paper, I'll paly Scissors"
  - "If I know you play Scissors, I'll paly Rock"
- The players will keep changing strategies.
- The game has a NE in mixed-strategies (both players play R,P,S with probability 1/3).

Example 2: The Prisoner's Dilemma

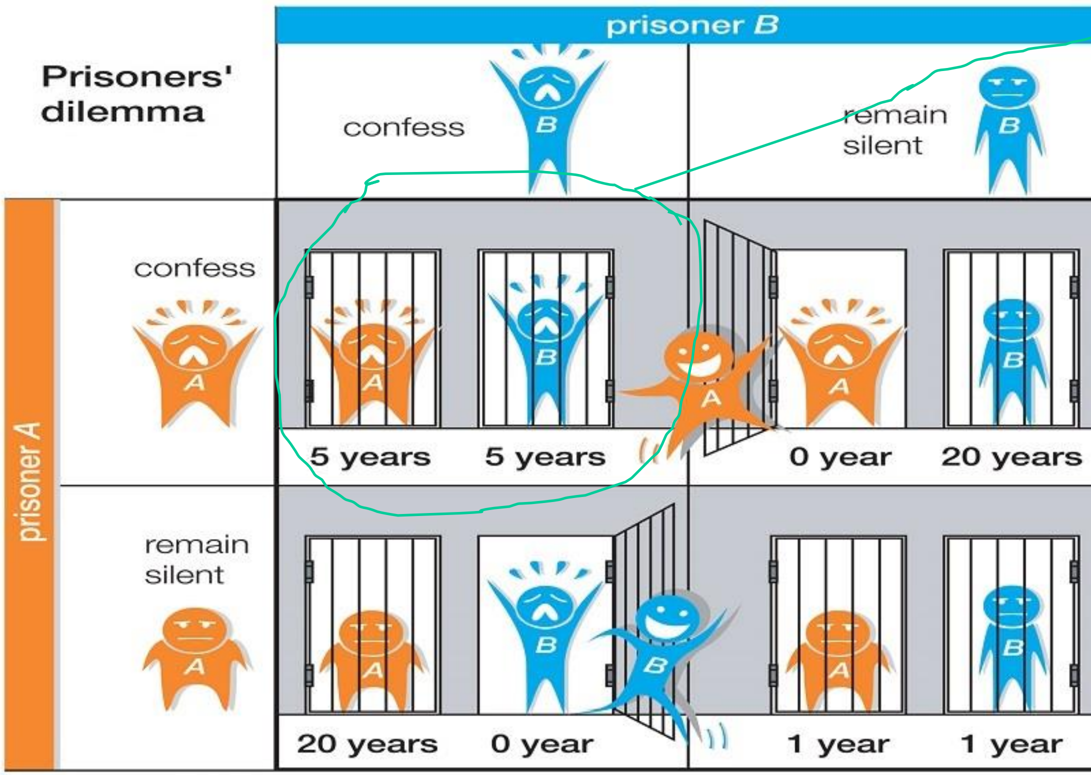
A symmetric two-player game in which the payoff matrix fulfills $c < a < d < b$

Alice \ Bob	Strategy 1	Strategy 2
Strategy 1	$a > c$ a, a	$a > c$ b, c $b > d$
Strategy 2	c, b	$b > d$ d, d

Strategy 2 is optimal if selected by both players. Unfortunately, strategy 1 is dominant for both players.

Example 2: The Prisoner's Dilemma

Two prisoners are on trial for a crime.
Each one faces a choice of confessing to the crime or remaining silent.



The only NE in this game

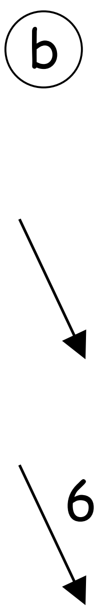
Resource Allocation Games

- A set of resources.
- A set of players.
- Each player needs a subset of resources that fulfills her needs.
- There may be several such subsets – defining the strategy space of the players.
- Players' costs depend on the congestion on the resources they use.

- Traditional Algorithms: A centralized utility assigns the players to resources.
- Algorithmic Game Theory: Players are selfish agents who select their assignment.

Example: Network Formation Games

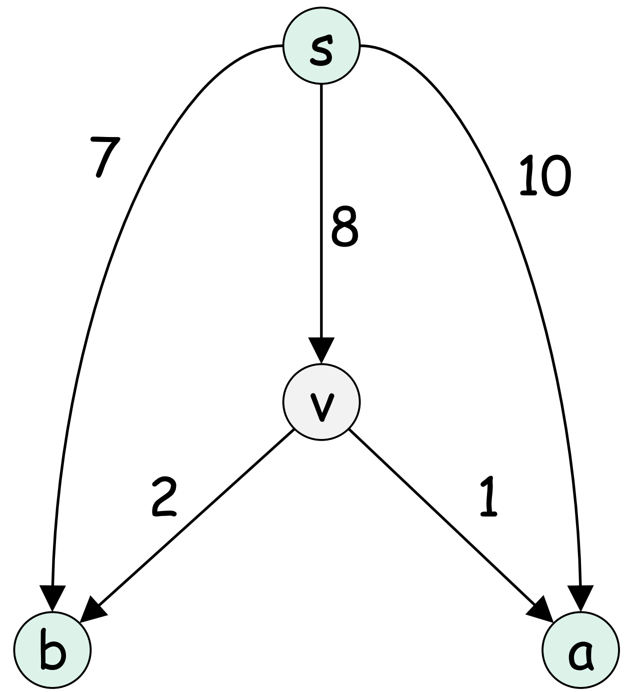
[Anshelevich, Dasgupta, Kleinberg, Tardos, Wexler, Roughgarden 2004]

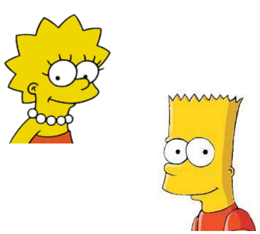


locations.

communication channels.

cost of creating the channel.





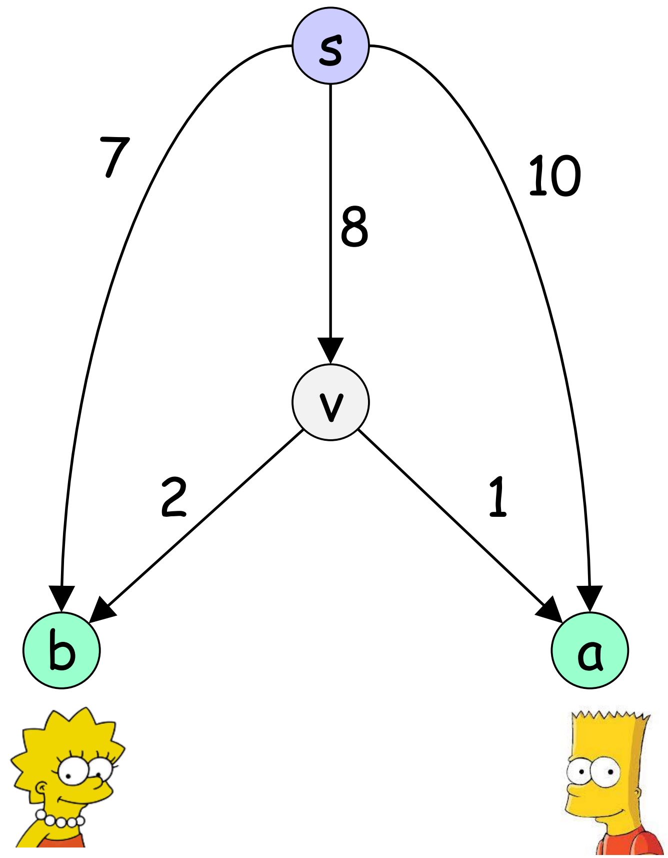
Players that need to transmit messages between locations in the network.

A network formation game

Example: Two players need to transmit messages from s

Player 1  needs to reach a

Player 2  needs to reach b



A network formation game

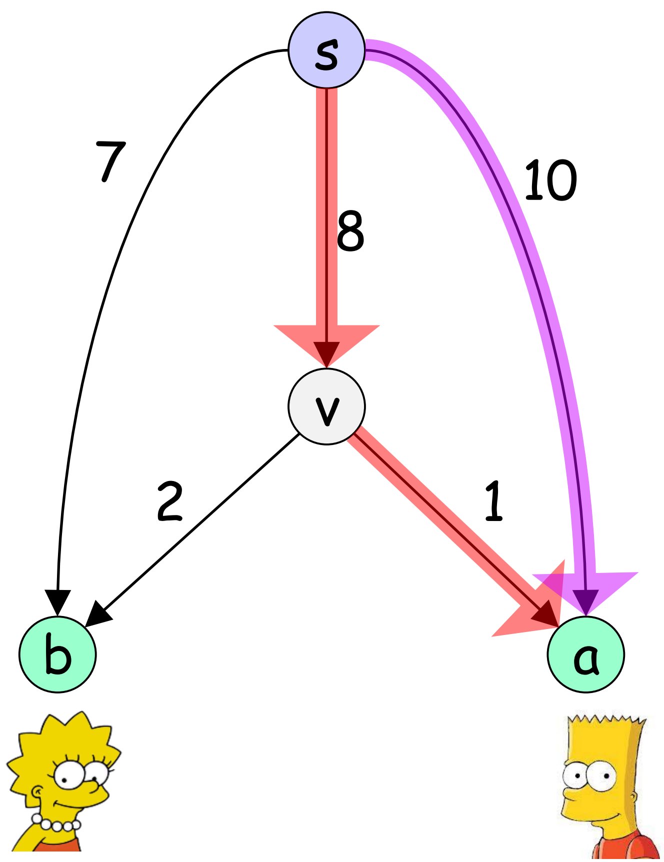
Example: Two players need to transmit messages from s

Player 1  needs to reach a

Player 2  needs to reach b

The strategy space of  :

{ {⟨s,v⟩,⟨v,a⟩} , {⟨s,a⟩} }

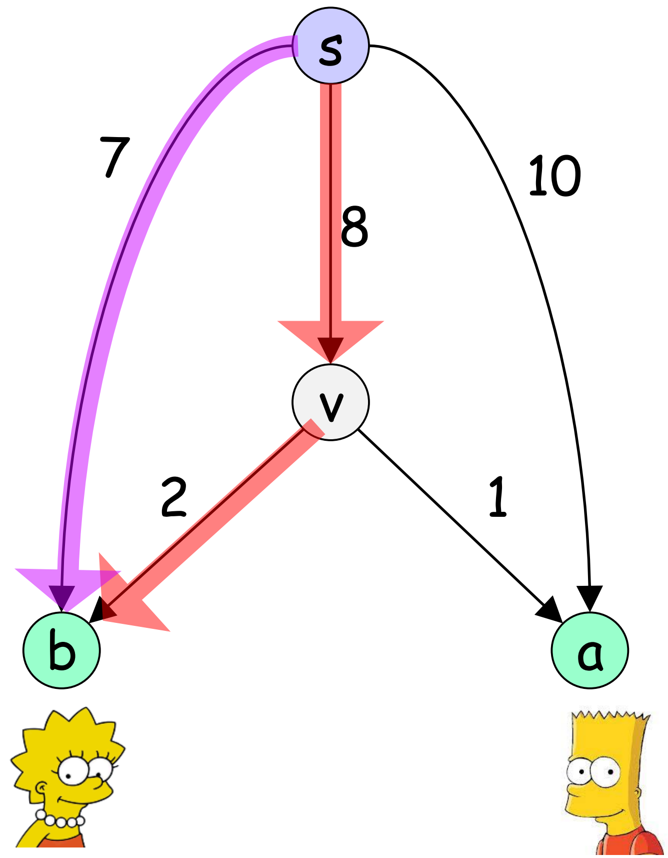


A network formation game

Example: Two players need to transmit messages from s

Player 1  needs to reach a

Player 2  needs to reach b

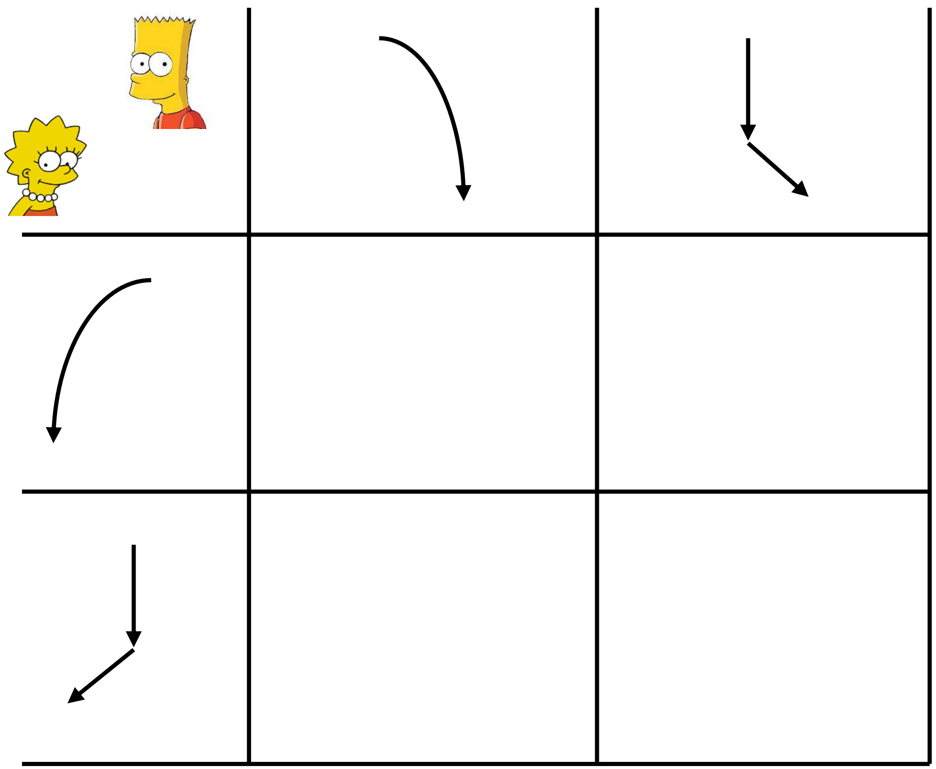


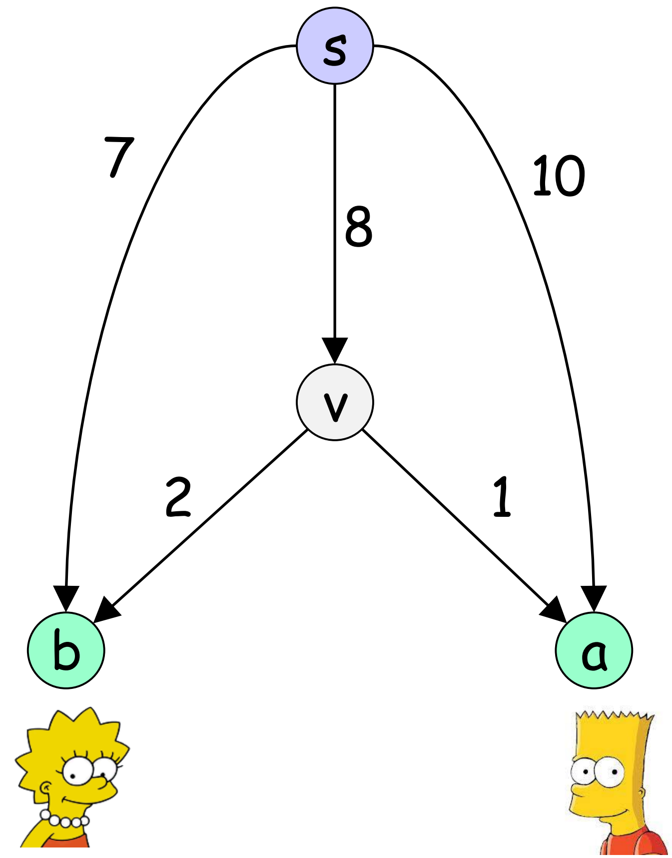
The strategy space of  :
{ {⟨s,v⟩,⟨v,a⟩} , {⟨s,a⟩} }

The strategy space of  :
{ {⟨s,b⟩} , {⟨s,v⟩, ⟨v,b⟩} }

A profile is a choice of strategy for each player.

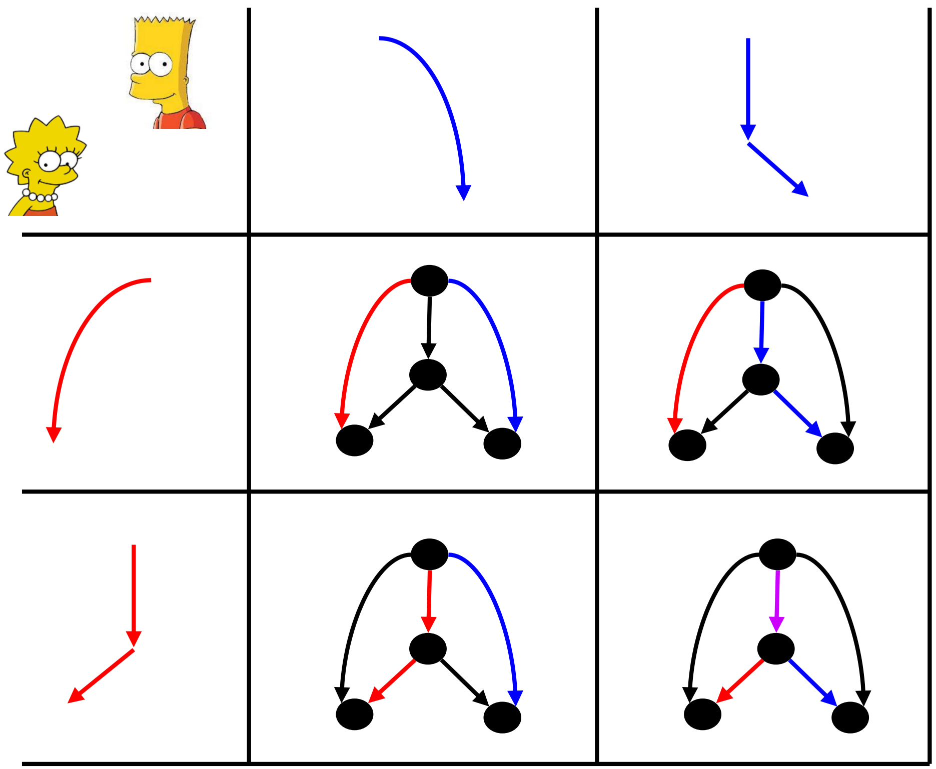
Four possible profiles in our example:

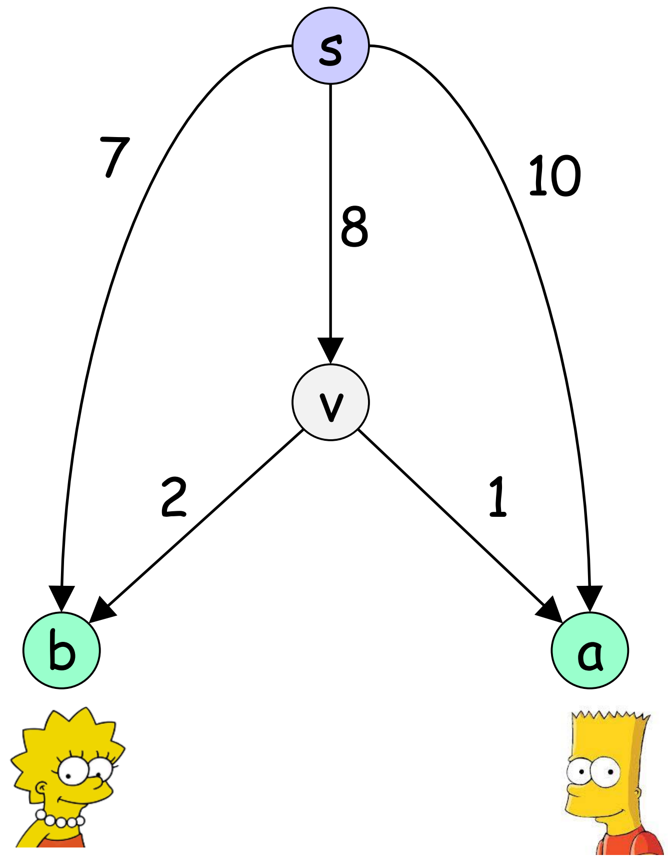




A **profile** is a choice of strategy for each player.

Four possible **profiles** in our example:

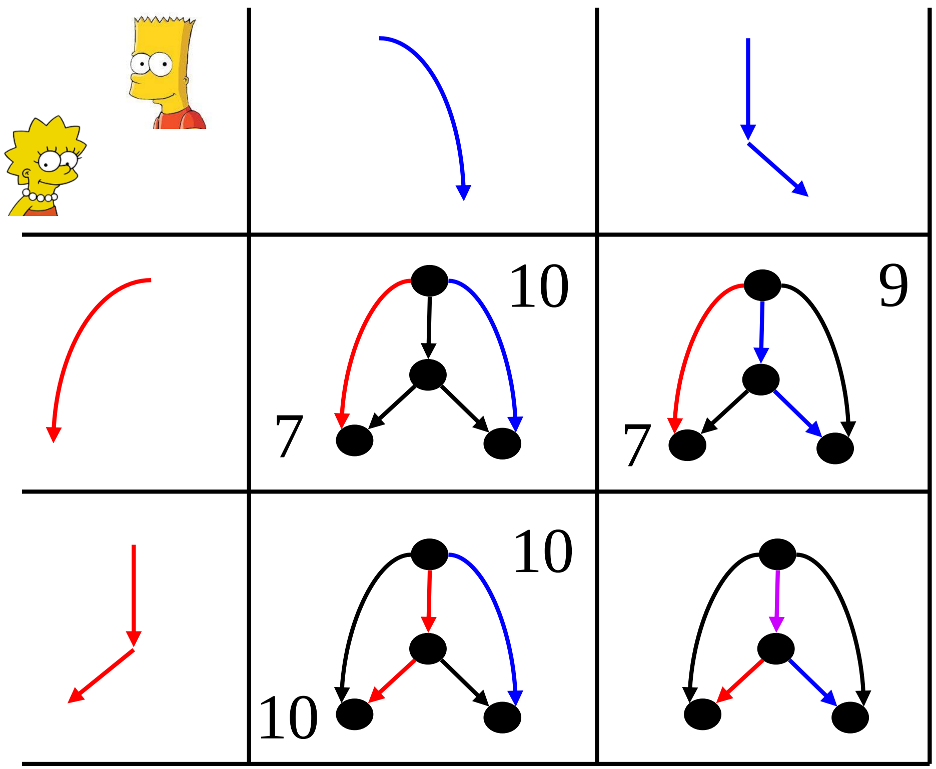


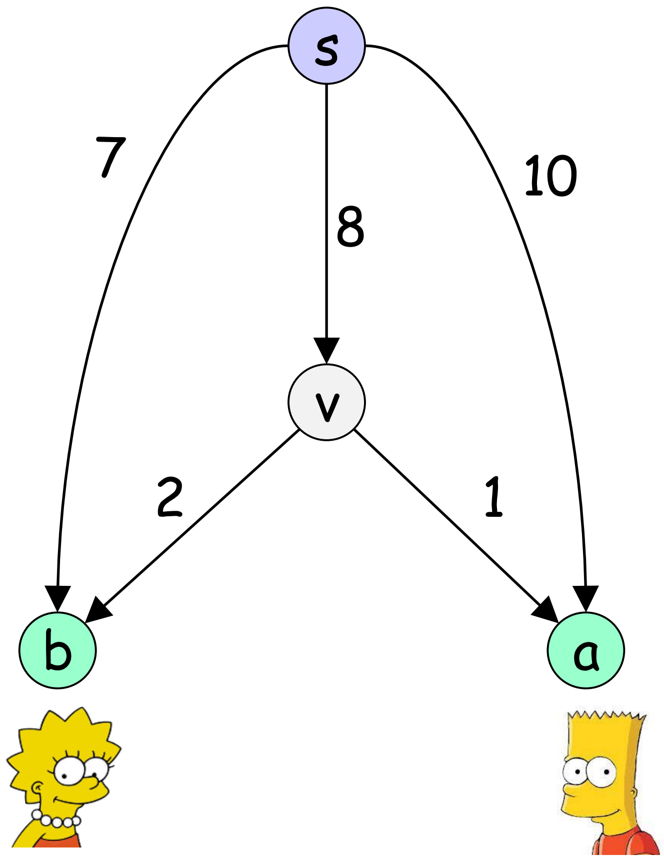


What are the costs?

A profile is a choice of strategy for each player.

Four possible profiles in our example:

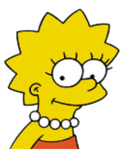




What are the costs?

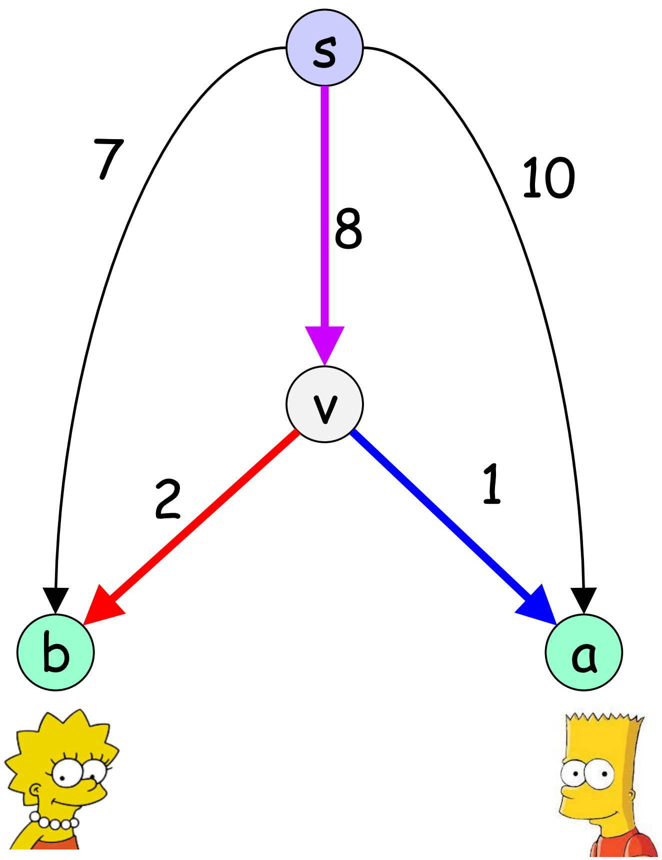
How is a cost shared?

Players that use the same channel share its cost:



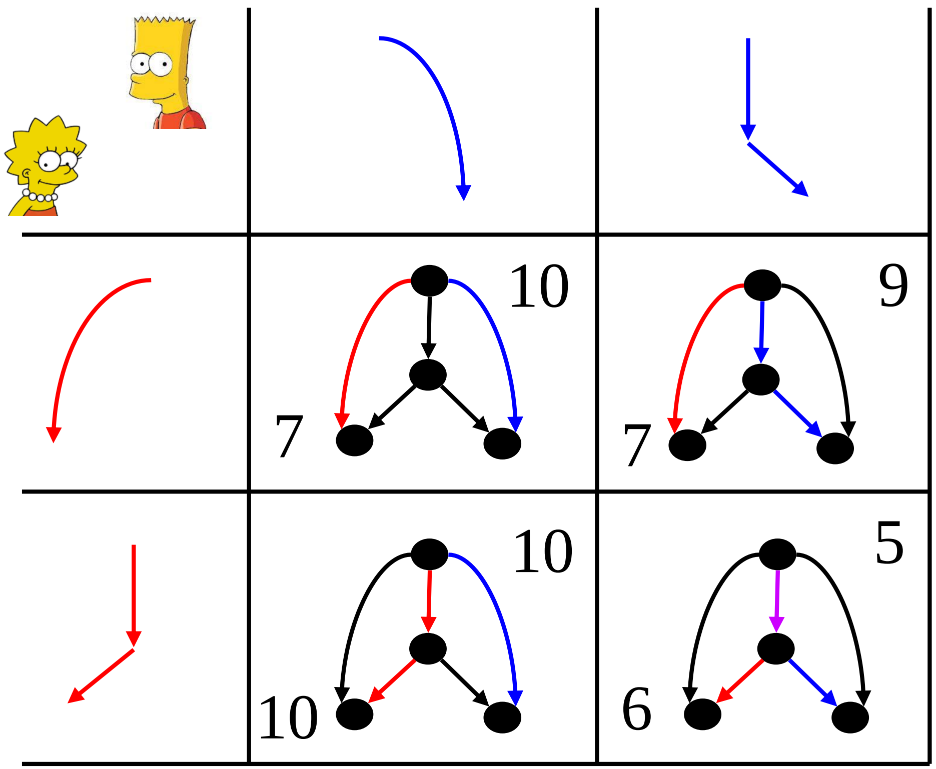
$$\frac{8}{2}+2=6$$

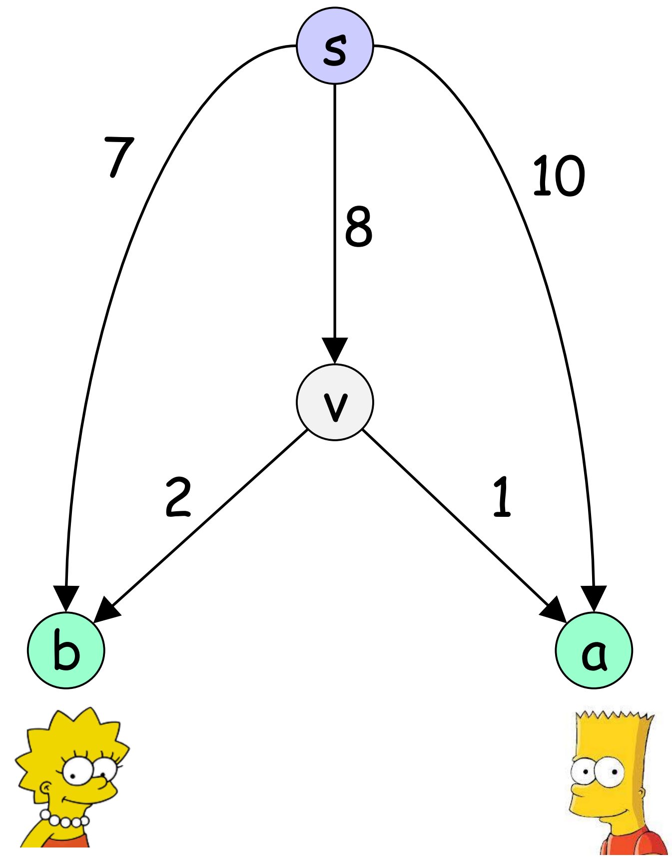
$$\frac{8}{2}+1=5$$



A profile is a choice of strategy for each player.

Four possible profiles in our example:



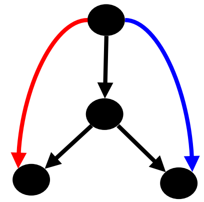


Best Response Dynamics (BRD)

- A local search method.
- Players proceed in turns, each performing a selfish improving step.
- In many scenarios, BRD lead to a pure Nash equilibrium.

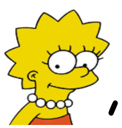
A stable profile in which no one has an improving step.

Best response dynamics.

Example: starting from 

Cost for  :10

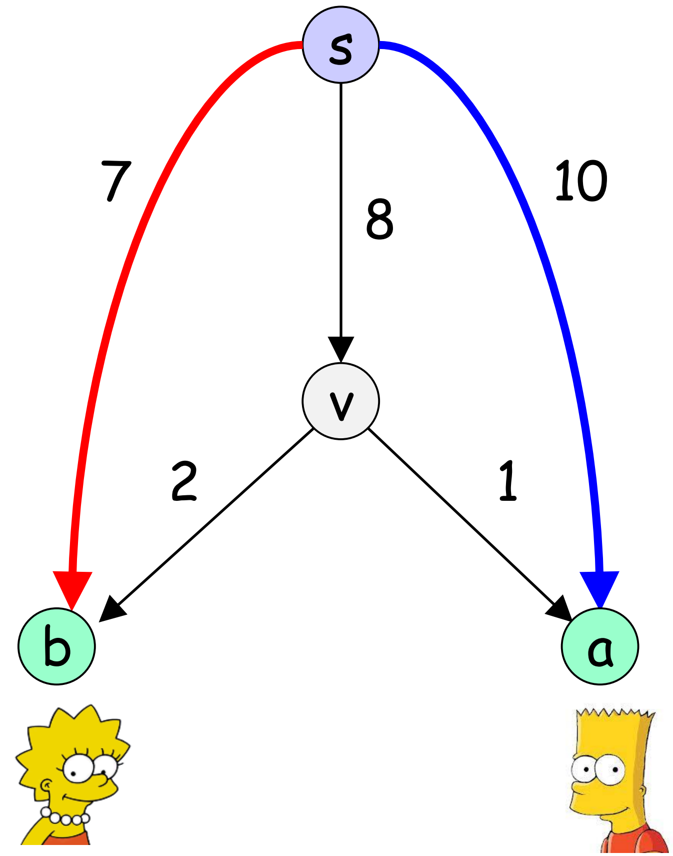
Cost for  :7

, want to change strategy?

No, 7 < 10

, want to change strategy?

Sure ,9 < 10 



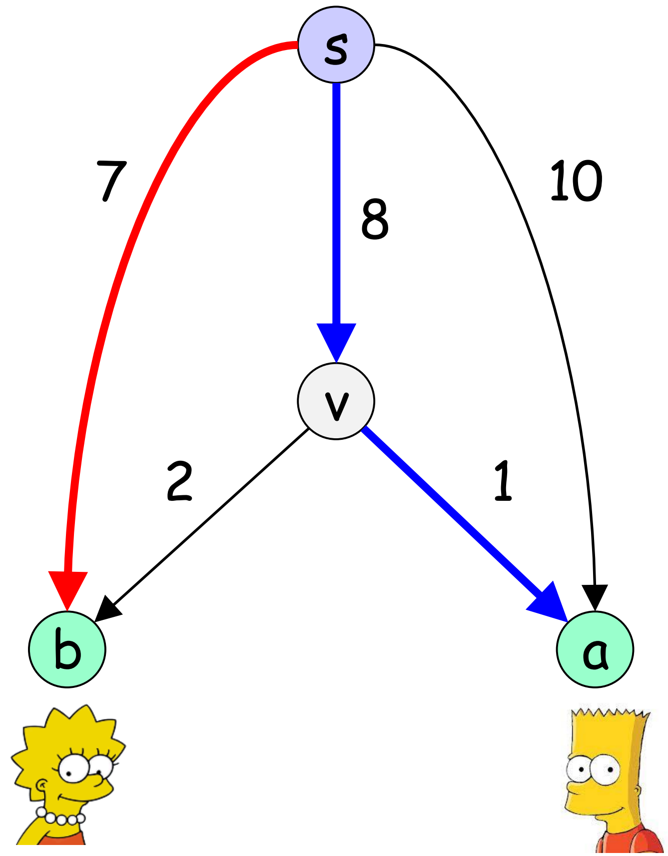
Best response dynamics.

Cost for  :9

Cost for  :7

, want to change strategy?

Yes, 6 < 7



Best response dynamics.

Cost for  :5

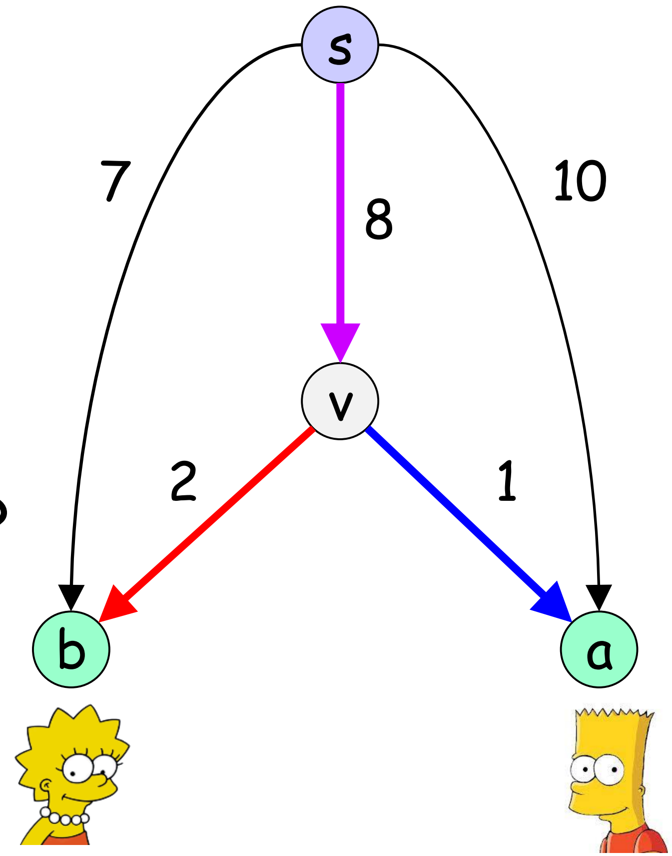
Cost for  :6

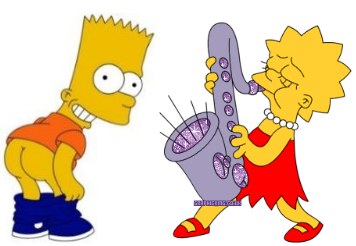
, want to change strategy?

No, 5 < 10

, want to change strategy?

No, 6 < 7





BRD halts, we've reached a NE.

Network Formation Games – Formal definition

- Given a directed graph $G = (V, E)$ with edge costs $c_e \geq 0$, a source node s, and k agents located at terminal nodes $t_1, \ldots, t_k$. Agent j must construct a path P_j from node s to its terminal t_j.

- Fair share: If x agents use edge e, they each pay c_e / x.
- The agents are selfish – each agent wants to minimize its cost.
- Agents might modify their selection as a response to actions of other agents.

Slides on multicast routing based on slides by Kevin Wayne.

Nash Equilibrium

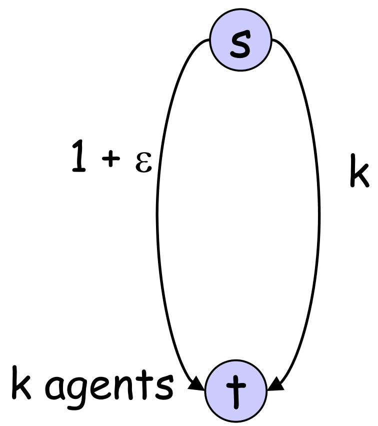
- Best response dynamics (BRD): Each agent is continually prepared to improve its solution in response to changes made by other agents.

- Nash equilibrium: Solution where no agent has an incentive to switch.

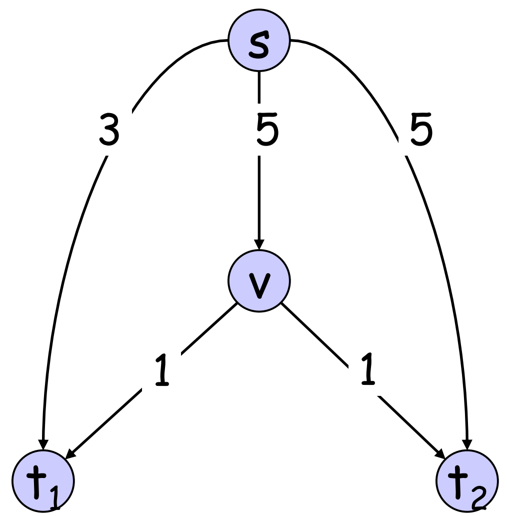
- Fundamental question: When do Nash equilibria exist? Does BRD terminate?

Social Optimum

- Social optimum: Minimizes total cost to all agents.
- Observation: In general, there can be many Nash equilibria. Even when it's unique, it does not necessarily equal the social optimum.



Social optimum = $1 + \varepsilon$
Nash equilibrium A = $1 + \varepsilon$
Nash equilibrium B = k



Social optimum = 7
Unique Nash equilibrium = 8

Price of Anarchy

- Price of anarchy: Ratio of worst Nash equilibrium to social optimum.

Theorem: In network formation games with k agents, PoA=k.

Proof:

- 1. PoA $\leq$ k. Assume by contradiction that in some NE, the PoA is more than k. This implies that some agent pays more than the social optimum. This agent can switch to his path in the social optimum and reduce its cost. Contradiction to NE.
- 2. For every $\varepsilon > 0$, there exists an instance with k agents for which PoA > k- ε.

  See left instance in previous slide.

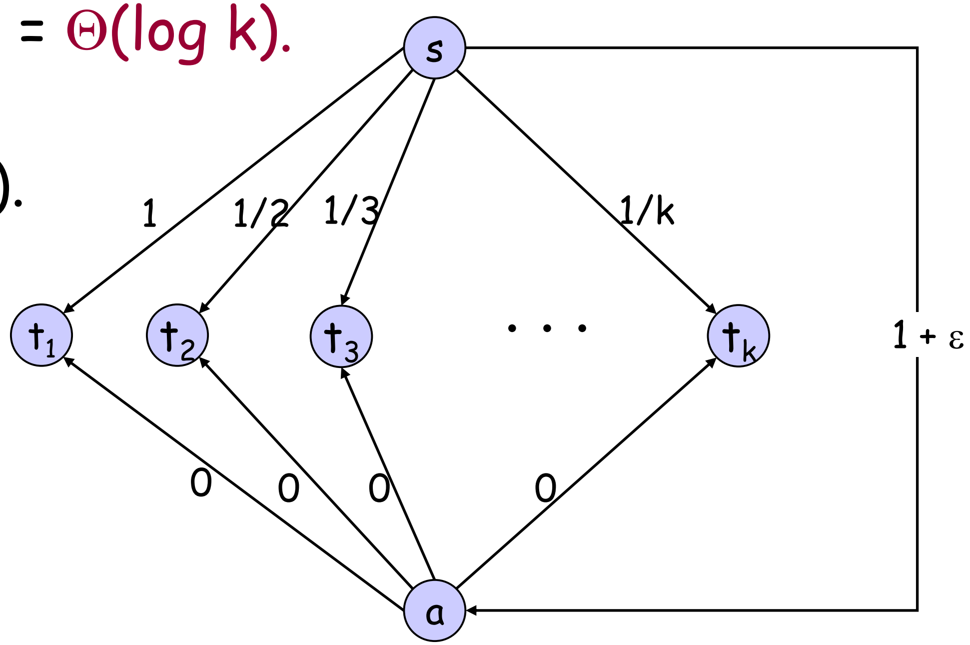
Price of Stability

- Price of stability: Ratio of best Nash equilibrium to social optimum.
- Fundamental question: What is the price of stability?

Example: Price of stability = $\Theta(\log k)$.
Social optimum: Everyone Takes bottom paths (via 'a').
Unique Nash equilibrium: Everyone takes top paths.
Price of stability:

$H(k) / (1 + \varepsilon)$.

$1 + 1/2 + \ldots + 1/k$



Finding a Nash Equilibrium

Theorem: The following algorithm terminates with a Nash equilibrium.

```
Best-Response-Dynamics(G,c) {
   Pick a path for each agent

   while (not a Nash equilibrium) {
      Pick an agent i who can improve by
       switching paths
      Switch path of agent i
   }
}
```

Finding a Nash Equilibrium

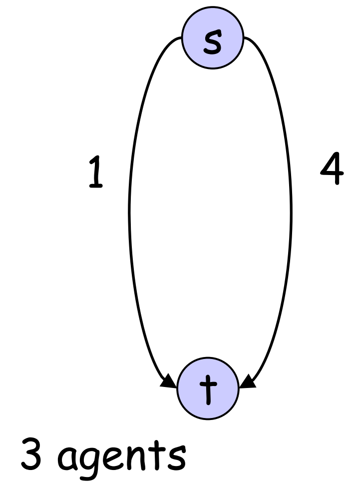
Proof Idea: Define a potential function over the possible solution sets. Show that the potential decrease whenever some agent improve.

Attempt 1:

Let $\Phi(s) = \Sigma_{j=1}^{k} cost(t_i)$ be the potential function.

A problem: The potential might increase when some agent improve.

Example: When all 3 agent use the right path, each pays 4/3 and the potential (total cost) is 4.
After one agent moves to the left path the potential increases to 5.



Finding a Nash Equilibrium

Attempt 2:

Consider a set of paths $P_1, \ldots, P_k$.

- Let x_e denote the number of paths that use edge e.
- Let $\Phi(P_1, \ldots, P_k) = \sum_{e \in E} c_e \cdot H(x_e)$ be a potential function.

  H(0) = 0, $H(k) = \sum_{i=1}^{k} \frac{1}{i}$
- Consider agent j switching from path P_j to path P_j'.
- Agent j switches because

$$\sum_{f \in P_j' - P_j} \frac{c_f}{x_f + 1} < \sum_{e \in P_j - P_j'} \frac{c_e}{x_e}$$

newly incurred cost saved cost

Finding a Nash Equilibrium

- Φ increases by

$$\sum_{f \in P_j' - P_j} c_f \left[H(x_f + 1) - H(x_f)\right] = \sum_{f \in P_j' - P_j} \frac{c_f}{x_f + 1}$$

- Φ decreases by

$$\sum_{e \in P_j - P_j'} c_e \left[H(x_e) - H(x_e - 1)\right] = \sum_{e \in P_j - P_j'} \frac{c_e}{x_e}$$

- Thus, net change in Φ is negative.
- Since there are only finitely many sets of paths, it implies that the algorithm terminates with a NE.

Bounding the Price of Stability

Claim: Let $C(P_1, \ldots, P_k)$ denote the total cost of selecting paths $P_1, \ldots, P_k$.
For any set of paths $P_1, \ldots, P_k$, we have

$$C(P_1,\ldots,P_k) \le \Phi(P_1,\ldots,P_k) \le H(k) \cdot C(P_1,\ldots,P_k)$$

Proof: Let x_e denote the number of paths containing edge e.

- Let E^+ denote set of edges that belong to at least one of the paths.

$$C(P_1,\ldots,P_k) = \sum_{e \in E^+} c_e \le \underbrace{\sum_{e \in E^+} c_e H(x_e)}_{\Phi(P_1,\ldots,P_k)} \le \sum_{e \in E^+} c_e H(k) = H(k)\ C(P_1,\ldots,P_k)$$

Bounding the Price of Stability

Theorem: There is a Nash equilibrium for which the total cost to all agents exceeds that of the social optimum by at most a factor of $H(k)$.

Proof:

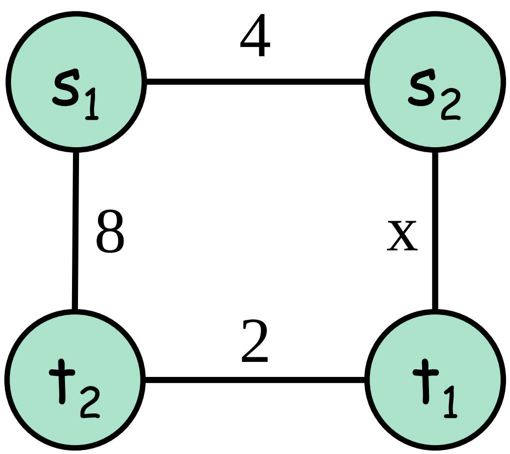
- Let $(P_1^*, \ldots, P_k^*)$ denote set of socially optimal paths.
- Perform BRD starting from P^*.
- Since Φ is monotone decreasing $\Phi(P_1, \ldots, P_k) \leq \Phi(P_1^*, \ldots, P_k^*)$.

$$C(P_1, \ldots, P_k) \underset{\substack{\uparrow \\ \text{previous claim} \\ \text{applied to } P}}{\leq} \Phi(P_1, \ldots, P_k) \leq \Phi(P_1^*, \ldots, P_k^*) \underset{\substack{\uparrow \\ \text{previous claim} \\ \text{applied to } P^*}}{\leq} H(k) \cdot C(P_1^*, \ldots, P_k^*)$$

A Network Formation Game.
In class Sample problem

Consider the following undirected network.
For every value of $x \geq 0$ determine

1. The social optimum.
2. The routings that form a NE
3. The PoA and the PoS.



Min-cost stable profile

Theorem: Given an instance of network formation game, it is NP-hard to determine if the game has a Nash equilibrium of cost at most C.

Proof: A reduction from 3-dimentional matching.

Input: a set of triplets $T \subseteq X \times Y \times Z$, where $|X| = |Y| = |Z| = n$. The number of triplets is $|T| \geq n$.

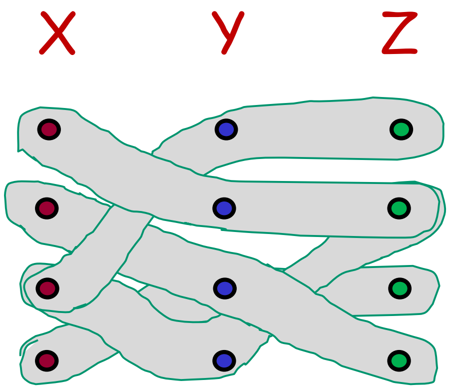
Output: a 3-dimensional matching in T of maximal cardinality: A subset $T' \subseteq T$, such that every element in $X \cup Y \cup Z$ appears at most once in T', and $|T'|$ is maximal.

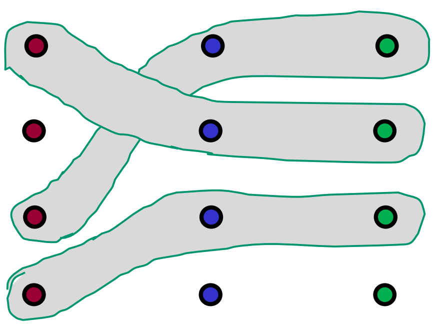
Maximum 3-dim matching

Example: n=4

$X=\{x_1,x_2,x_3,x_4\}$, $Y=\{y_1,y_2,y_3,y_4\}$, $Z=\{z_1,z_2,z_3,z_4\}$.

$T=\{ (x_1,y_2,z_2), (x_2,y_3,z_4), (x_3,y_1,z_1), (x_3,y_4,z_2), (x_4,y_3,z_3) \}$





A maximum 3-matching

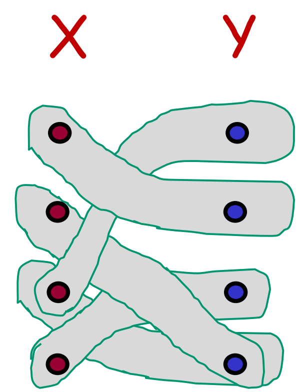
Optimum matching has size 3

$M= \{ (x_1,y_2,z_2), (x_3,y_1,z_1), (x_4,y_3,z_3) \}$

Maximum 3-dim matching

The Maximum 3-dim matching problem is known to be NP-hard (and hard to approximate), even if every element is restricted to belong to at most 3 triplets.

Remark: 2-dim matching (in which $T \subseteq X \times Y$) is a special case of bipartite matching. It is solvable in poly-time. algorithm.



Back to our reduction: Given an instance of 3D-matching. We will construct a network formation game.

Reduction to 3-dim matching

The network:

$$V = \underbrace{X \cup Y \cup Z}_{\text{element-nodes}} \cup \underbrace{\{v_{ijk} \text{ for each triplet } (x_i, y_j, z_k)\}}_{\text{triplet nodes}} \cup \underbrace{\{t\}}_{\text{target}}$$

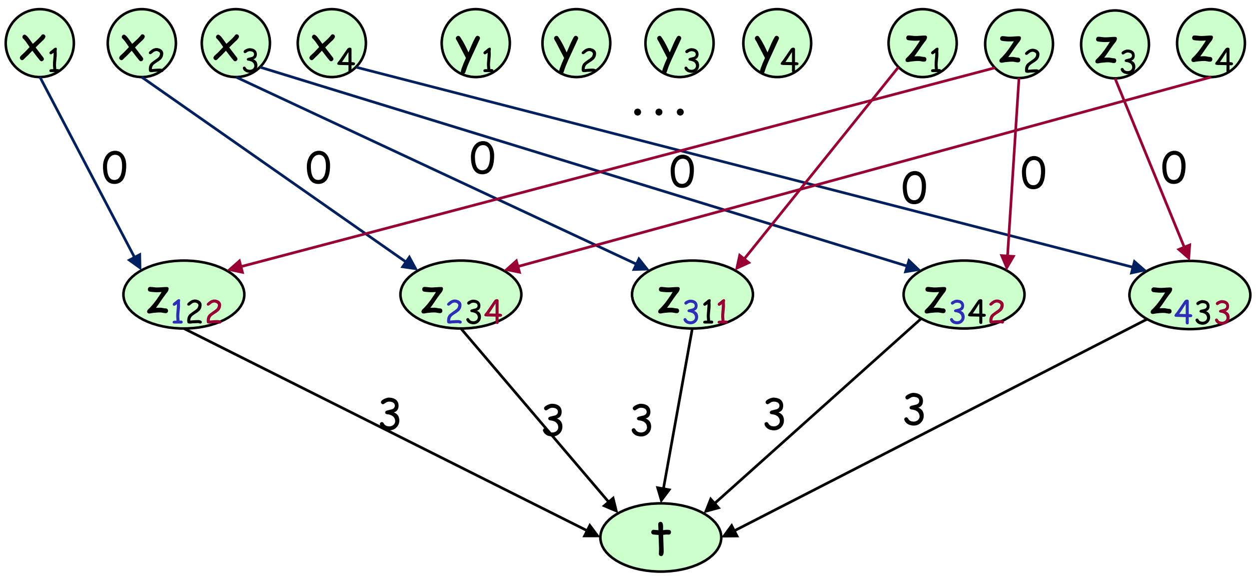
E= $\{(v_{ijk}, t)\}$ - *for* every triplet-node, $c(e)=3$.
$\cup \{(x_i,v_{ijk})\} \cup \{(y_j,v_{ijk})\} \cup \{(z_k,v_{ijk})\}$ - for every element-node, and its corresponding triplet-nodes, $c(e)=0$.

The game: There are $3n$ players, one for each element. The objective of every player is a path from its element-node to the target t.

Reduction to 3-dim matching

Example: $X=\{x_1,x_2,x_3,x_4\}$, $Y=\{y_1,y_2,y_3,y_4\}$, $Z=\{z_1,z_2,z_3,z_4\}$.

$T=\{ (x_1,y_2,z_2), (x_2,y_3,z_4), (x_3,y_1,z_1), (x_3,y_4,z_2), (x_4,y_3,z_3) \}$



Claim: there exists a 3-dim matching of size n if and only if the NFG has a Nash equilibrium of cost 3n.

Proof: In Class.

Summary: Network formation games

- Existence: Nash equilibria always exist for k-agent multicast routing with fair sharing.

- Price of stability: Best Nash equilibrium is never more than a factor of H(k) worse than the social optimum. For some networks this is tight.
- Price of anarchy: Any Nash equilibrium is never more than a factor of k worse than the social optimum. For some networks this is tight.

- Fundamental open problem: Find any Nash equilibria in poly-time.

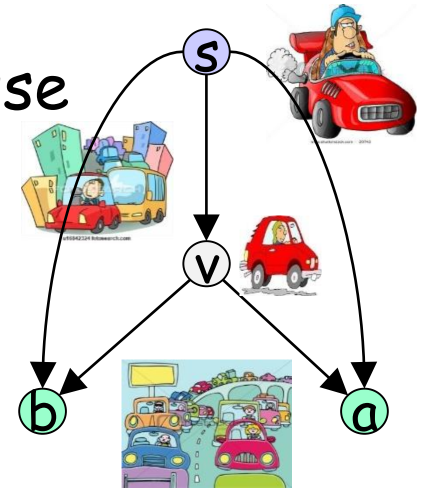
Congestion Games

Cost Sharing games: Games with **positive** congestion effect – players want to share resources with other players.

Example: Network formation.

Congestion Games: Games with **negative** congestion effect – players want to use resources with no (few) partners.

Both are justified by real applications.



Example: Network Congestion Game

- A network congestion game is defined by a tuple $\{N, G, \{s_i, t_i\}$ for all $i \in N$, $\{c_j\}$ for all $j \in M\}$
- $N = \{1..n\}$. the set of players. Each player is associated with a source $s_i \in V$ and a target $t_i \in V$.
- a graph $G=(V,E)$, $E = \{1..m\}$ denotes the graph edges.
- For $i \in N$, the objective of Player i is a path from s_i to t_i. Let A_i denote the strategy space of Player i (= set of $(s_i$-$t_i)$-paths in G).
- For $j \in E$, $c_j \in R^n$ denotes the vector of costs, where $c_j(k)$ is the delay travelling on edge j when used by k players.

Example: Network Congestion Game

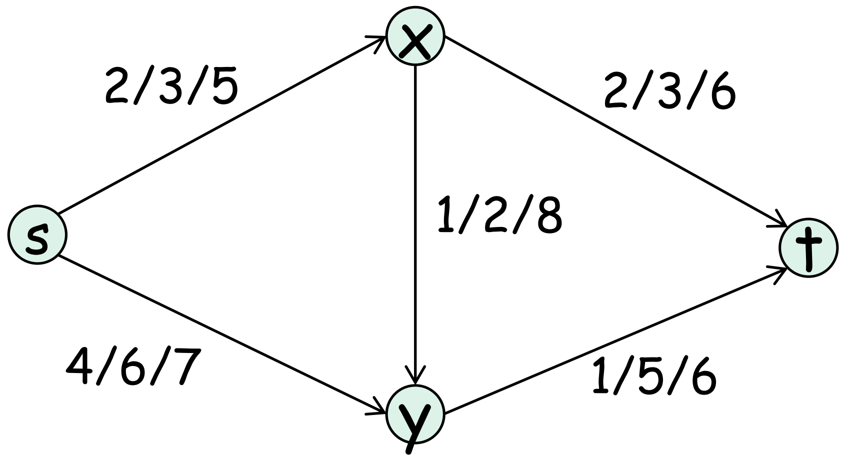
- A profile a is a set of strategies selected by the players.
- $a=(a_1, a_2,\ldots, a_n)\in(A_1 \times A_2 \times\ldots \times A_n)$
- Let $n_j(a)$ denote the number of players for which resource j is used in profile a. ($j \in a_i$)
- The cost function for player i in the profile a is:

  $u_i(a) = \sum_{j\in a_i} c_j\left(n_j(a)\right)$.

Remark: (For now) all players are equal in a sense that they have the same 'weight' (it doesn't matter which players are using a facility, only how many players are using it).

Example: Network Congestion Game

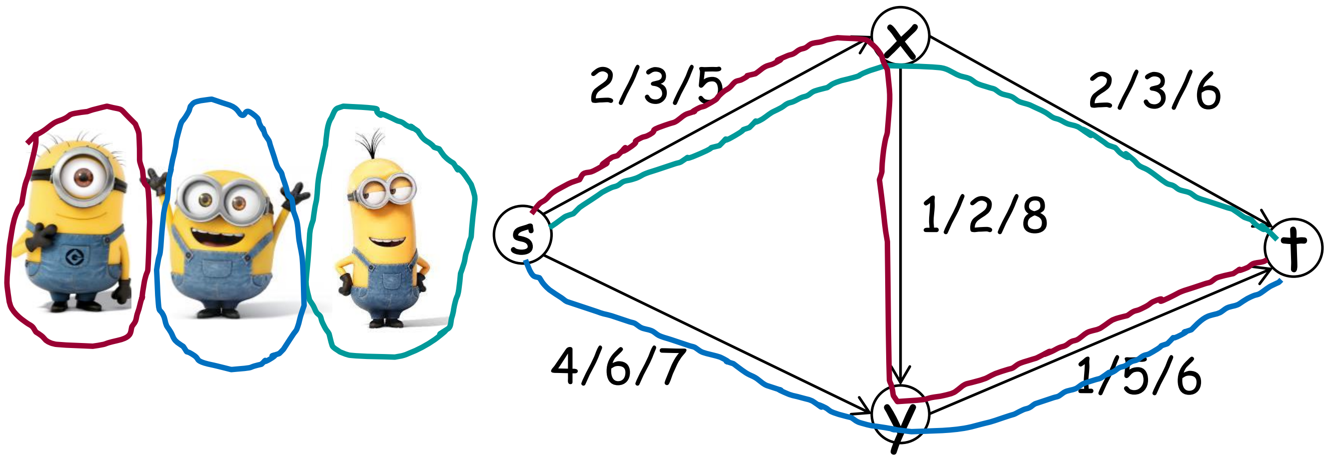
Stuart Bob Kevin



- Assume that players A,B,C have to go from s to t
- Edges are labeled by the function c_j.

For example, if two players are using the edge (yt) then each player experiences a delay of 5 on this edge. The strategy set of all three players includes three paths: for all i=A,B,C A_i={{sx,xt},{sy,yt},{sx,xy,yt}}

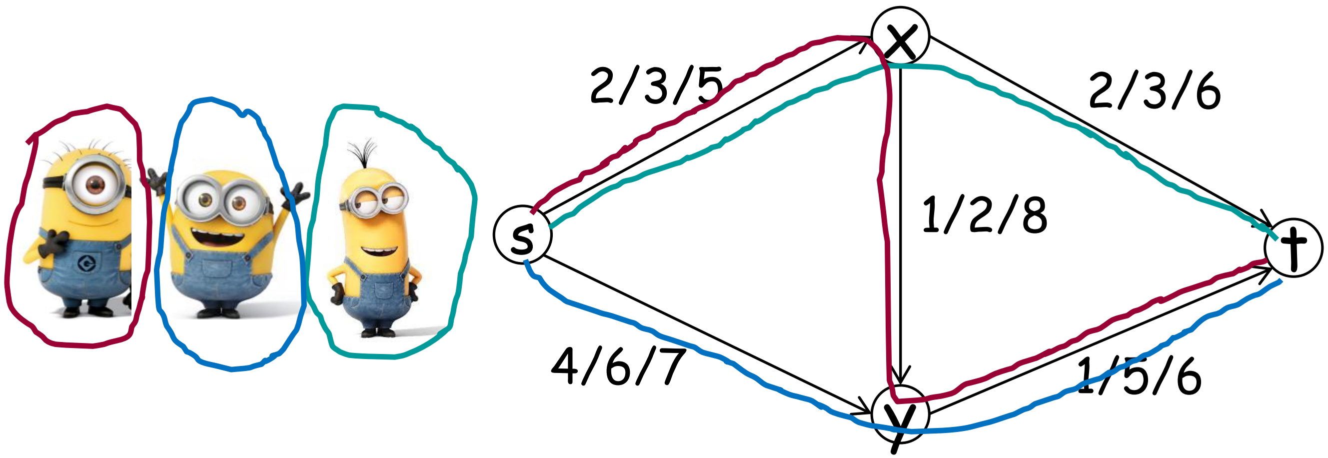
Network Congestion Game



Example:

-  selects the path s-x-t. cost (delay) = 3+2=5
-  selects the path s-x-y-t. cost = 3+1+5=9
-  selects the path s-y-t. cost = 4+5=9

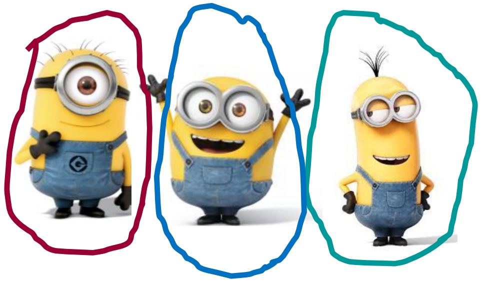
Network Congestion Game

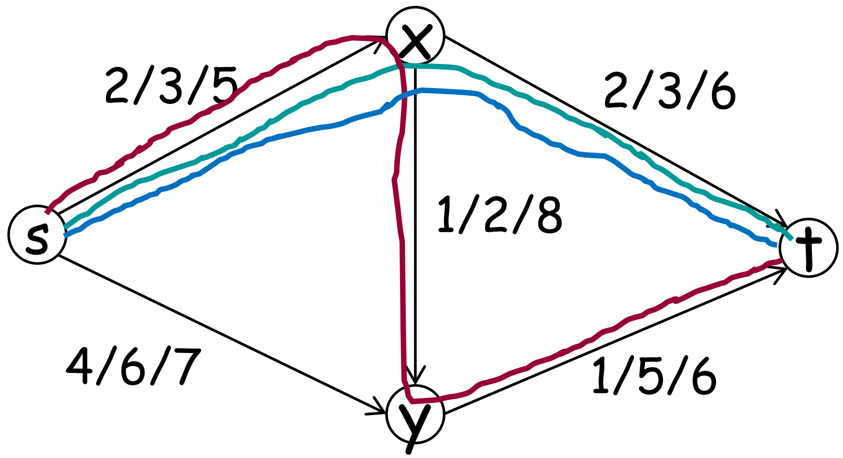


Is it a Nash Equilibrium profile?

- Should  switch from s-y-t to s-x-t?
- Currently his cost is 9. By migrating…

Network Congestion Game

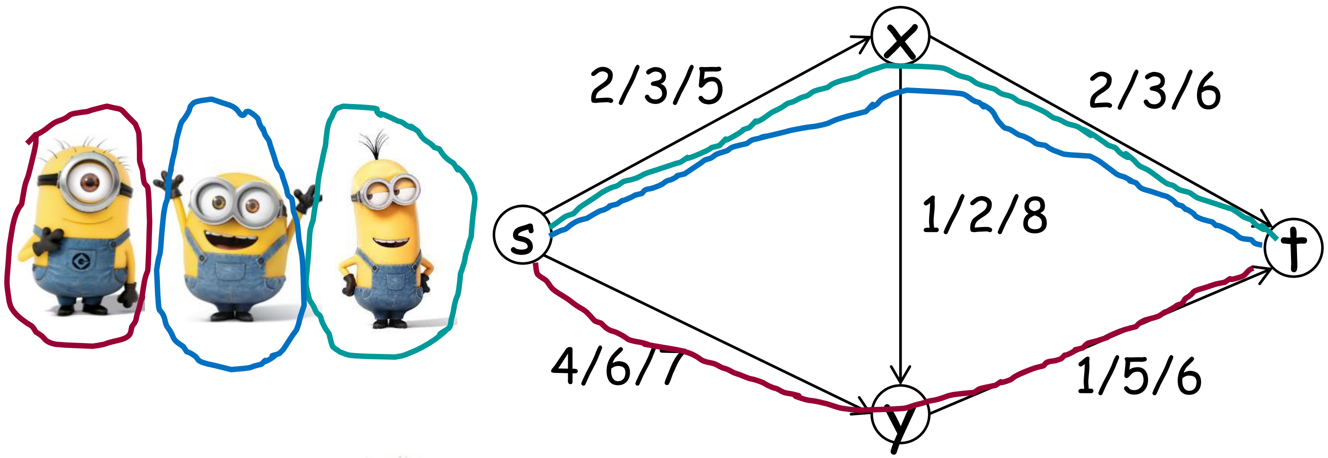




Is it a Nash Equilibrium profile?

- Should  switch from s-y-t to s-x-t?
- Currently his cost is 9. By migrating his cost would reduce to 5+3=8.

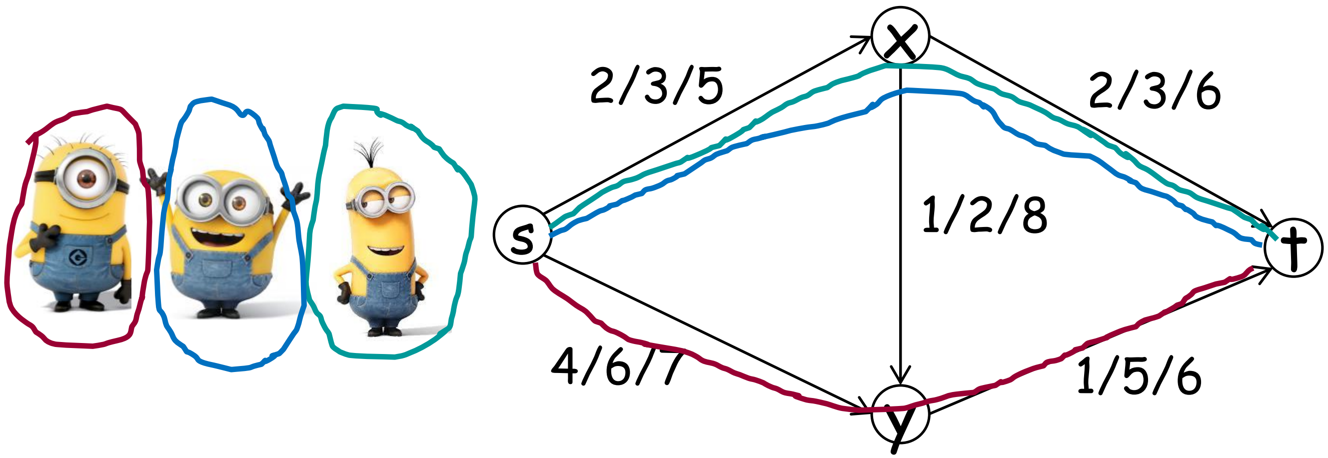
Network Congestion Game



- Next,  is migrating and reduces his cost from 8 to 5.
- Each of   pays 6.

We have reached a NE.

Network Congestion Game



Note: Even though the game is symmetric (all players have the same objective), they do not share the same strategy in the NE.

General Congestion Game

- A congestion game is defined by a tuple

$\{N, M, \{A_i\}$ for all $i \in N$, $\{c_j\}$ for all $j \in M\}$

- $N = \{1..n\}$ denotes the set of players.
- $M = \{1..m\}$ denotes the set of resources.
- For $i \in N$, A_i denotes the set of strategies of player i, where each $a_i \in A_i$ is a non empty subset of the resources.
- For $j \in M$, $c_j \in R^n$ denotes the vector of costs, where $c_j(k)$ is the cost related to each user of resource j, if there are exactly k players using it

Nash Equilibrium Existence.

Theorem: Every finite congestion game has a pure strategy Nash equilibrium.

Proof: Let a be a deterministic strategy vector as defined above, let Φ:A$\rightarrow$R be a potential function defined as follows:

$$\Phi(a) = \sum_{j=1}^{m} \sum_{k=1}^{n_j(a)} c_j(k)$$

Claim: For every improvement step of player i, it holds that $\Delta\Phi=\Delta u_i$.

Proof: in class.

Nash Equilibrium Existence.

Corollary: Since Φ can accept a finite number of values, BRD converges to a NE.

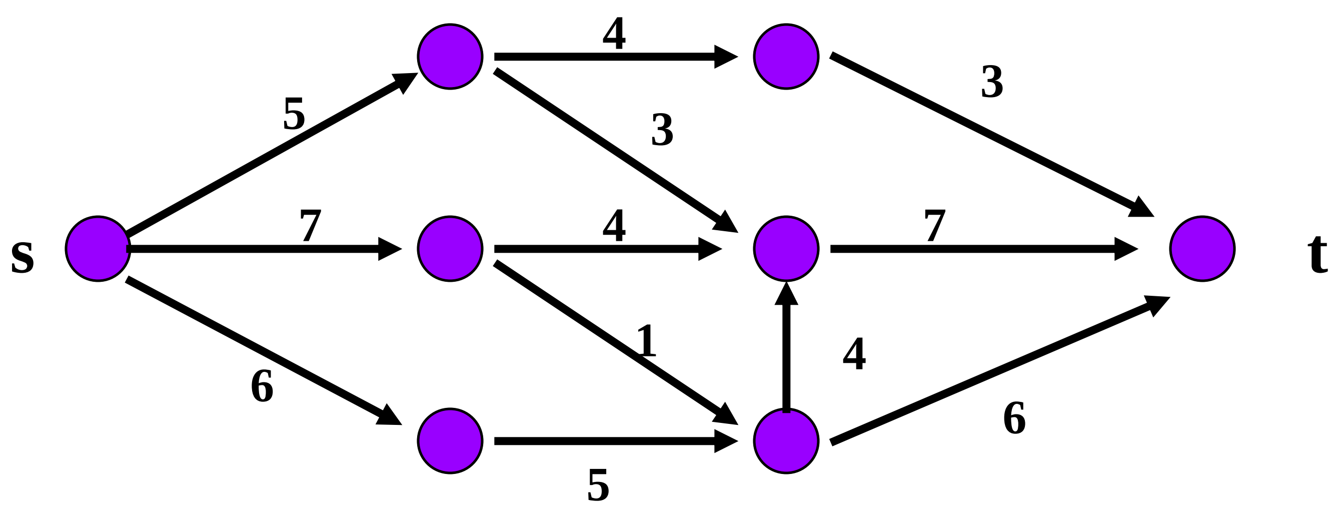
```
Best-Response-Dynamics(G,c) {
   Pick a strategy for each agent

   while (not a Nash equilibrium) {
      Pick an agent i who can improve by
switching strategy.
      Let i switch to a lowest-cost strategy}
     }
```

Computing a NE in congestion games

- For general congestion games, the problem of finding a NE is PLS-complete (PLS = Polynomial-time Local Search). Probably can't be solved in polynomial time.
- We will see a polynomial time algorithm for symmetric network congestion games.
- A network congestion game is symmetric if all the players have the same set of strategies (common source and target vertices).
- The algorithm is based on a reduction to a max-flow min-cost problem.

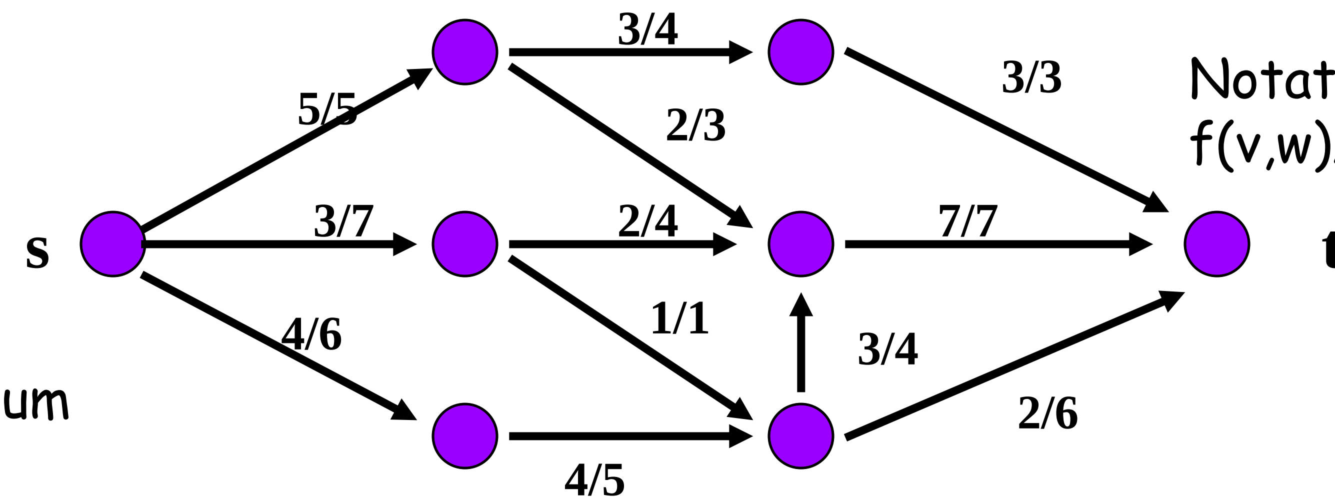
Maximum Flow (no costs)

- Input: a directed graph (network) G
  - each edge (v,w) has associated capacity c(v,w)
  - a specified source node s and target node t
- Problem: What is the maximum flow you can route from s to t while respecting the capacity constraint of each edge?



Properties of Flow: $f(v,w)$ - flow on edge (v,w)

- **Edge condition:** $0 \le f(v,w) \le c(v,w)$: the flow through an edge cannot exceed the capacity of an edge.
- **Vertex condition:** for all v except s,t : $\Sigma_u f(u,v) = \Sigma_w f(v,w)$: the total flow entering a vertex is equal to total flow exiting this vertex.
- total flow leaving s = total flow entering t.

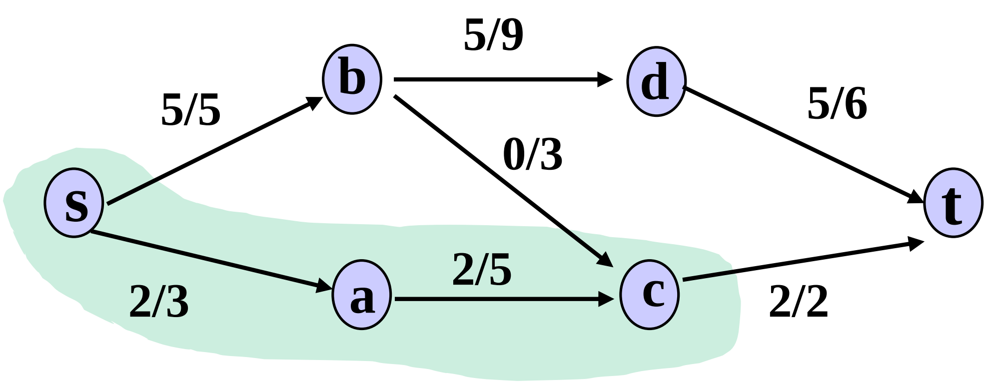


Notation on edges $f(v,w)/c(v,w)$

Not a maximum flow!

Max-flow Min-Cut Theorem

The value of a maximum flow in a network is equal to the minimum capacity of a cut.



Example: Flow value = max-cut capacity = 7

Max-flow Min-cost problem

Given a flow-network G where each edge (v,w) has associated capacity $c(v,w)$, and **a cost** $cost(v,w)$.

The goal is to find a maximum flow of minimum cost.

The cost of a flow f : $\Sigma_{f(v,w)>0}\, cost(v,w)f(v,w)$

Out of all the maximum flows, which has minimal cost?

The max-flow min-cost problem has a polynomial time algorithm.

Computing a NE in a symmetric network congestion game

Input:

- A graph $G=(V,E)$
- A source $s \in V$ and a target $t \in V$.
- For each edge $j=1..m$, the cost function $c_j(k)$
- The number of players, n.

Output: A NE profile.

Computing a NE in a symmetric network congestion game

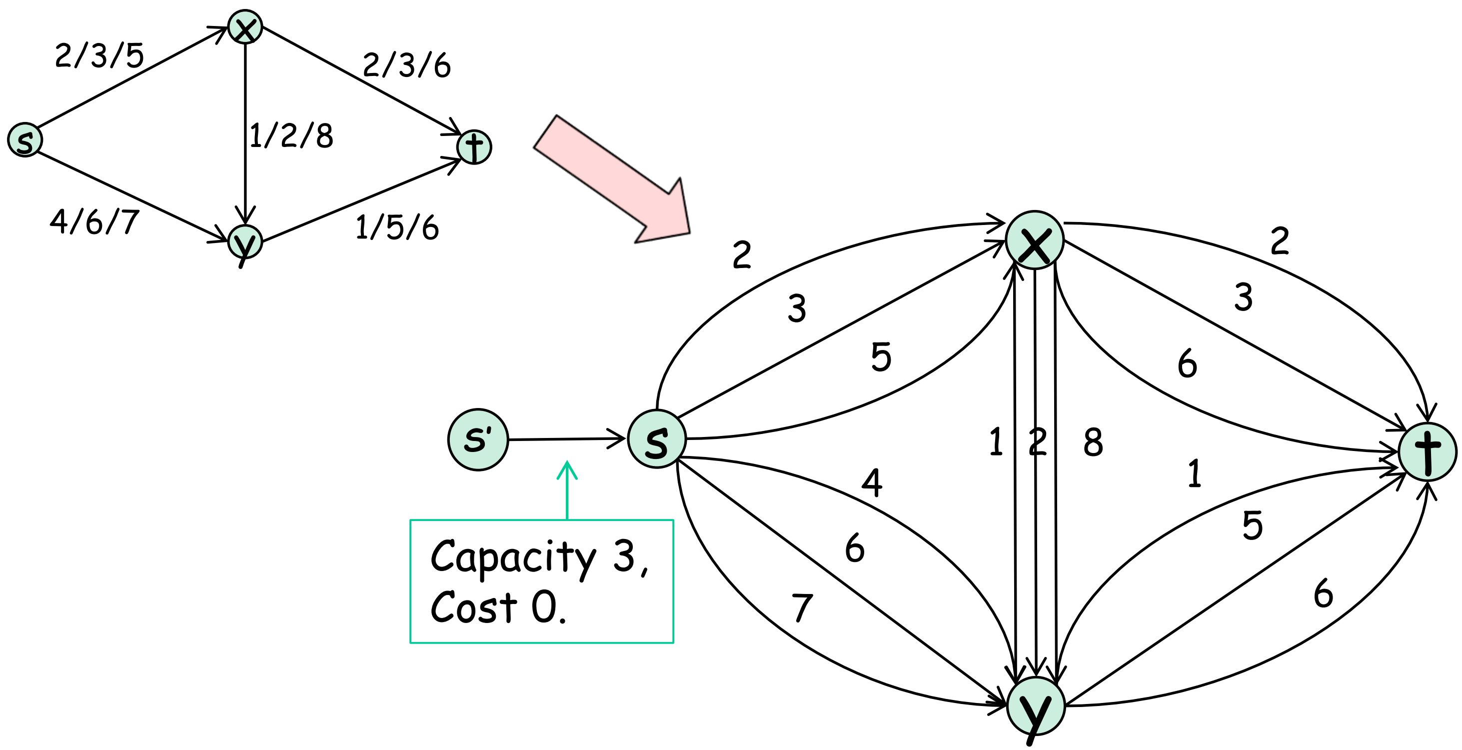
Algorithm:

- Build a flow network with costs: replace in G every edge e by n parallel edges between the same nodes, each with capacity 1, and with costs $c_e(1),\ldots, c_e(n)$.
- Find a min-cost flow of value n (how?)
- The flow induces n disjoint paths from s to t. These paths define a NE profile.

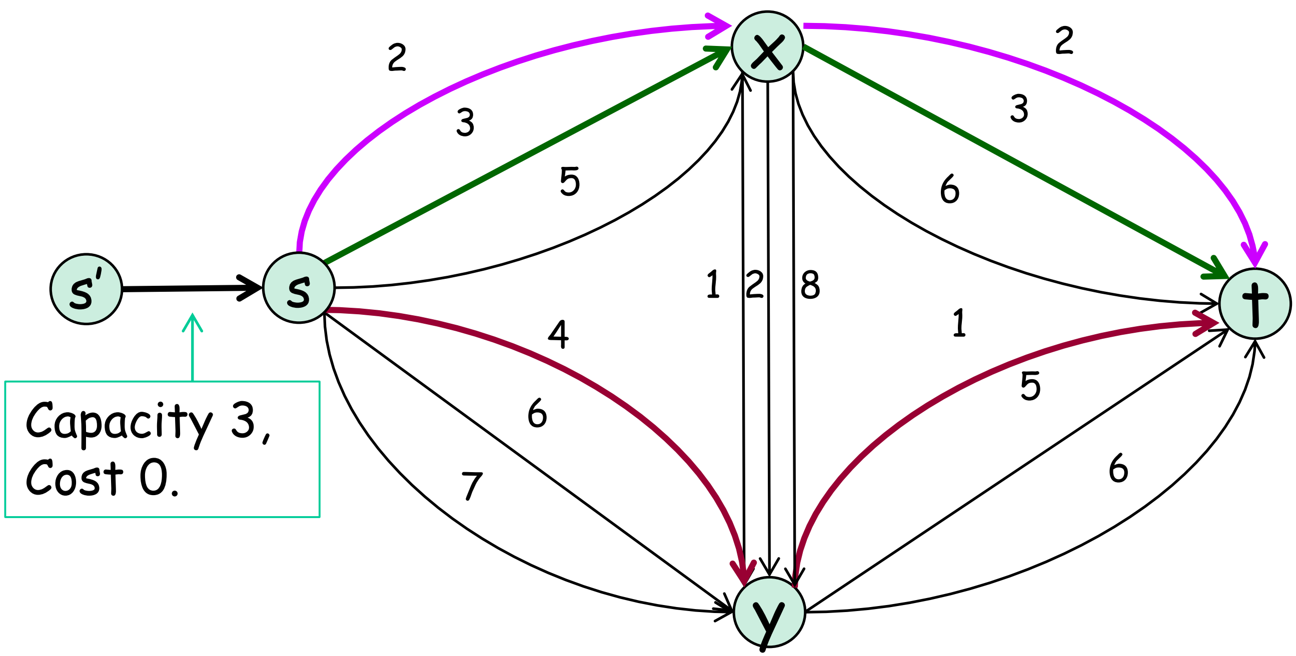
Proof and Analysis: In class

Computing a NE in a symmetric network congestion game

Construction example: edges are labeled by their costs, all edges except (s',s) have capacity 1.



Computing a NE in a symmetric network congestion game



Min-cost flow of value 3.

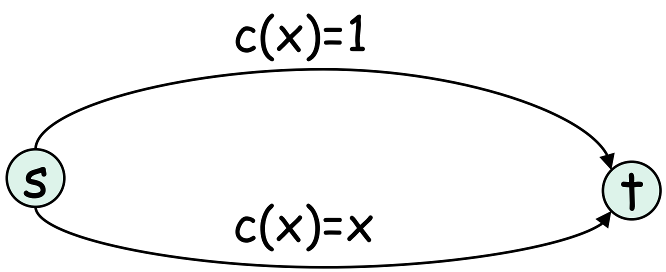
Computing a NE in a symmetric network congestion game

Note: The algorithm does not find a social optimum profile.

Minimizing the potential is not equivalent to minimizing the total players' cost!

How bad is selfish routing?

- Pigou's Network: For routing with splittable flow, the price of anarchy as well as the price of stability can be 4/3.

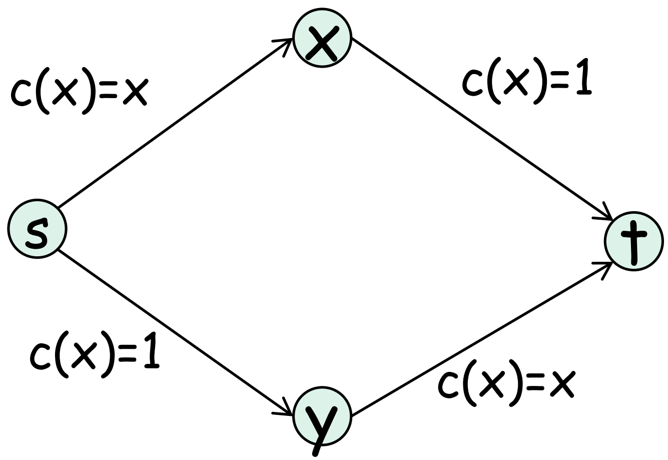


- For simplicity we assume the total load is 1 and measure the load on each edge by fractions

Objective function: Minimize the average delay.
Social Optimum: split the flow, $\frac{1}{2}$ on top path, $\frac{1}{2}$ on lower path. Average delay is $(1 + \frac{1}{2})/2 = \frac{3}{4}$.
Unique NE: All players in lower path. Average delay is 1.

How bad is selfish routing?

- Braess's Paradox: The addition of an intuitively helpful link can negatively impact all of the users of a congested network.

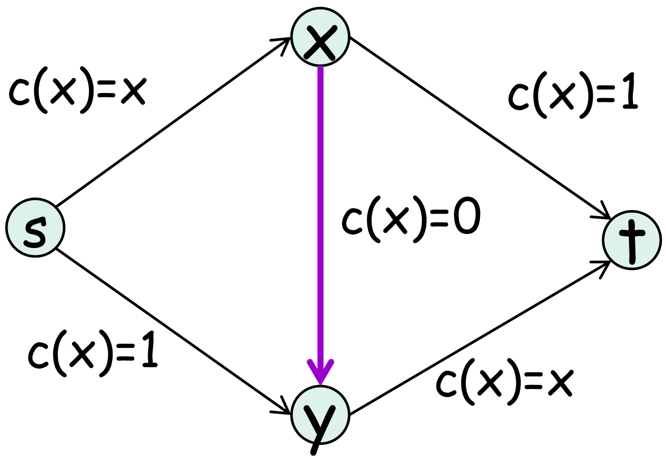


- Again, for simplicity we measure the load in fractions.
- What is the unique NE for routing of one unit of flow from s to t?

Answer: split the flow, $\frac{1}{2}$ on top path, $\frac{1}{2}$ on lower path.
The delay for all players is 3/2.
This NE is also the optimal social cost.

How bad is selfish routing?

- Suppose that a new edge with delay 0 (independent of the load) is added.



The optimal flow remains the same. The new edge is not used.

- However, it is not a NE!
- The new unique NE is when all the flow route in the path s-x-y-t. The corresponding delay is 2.

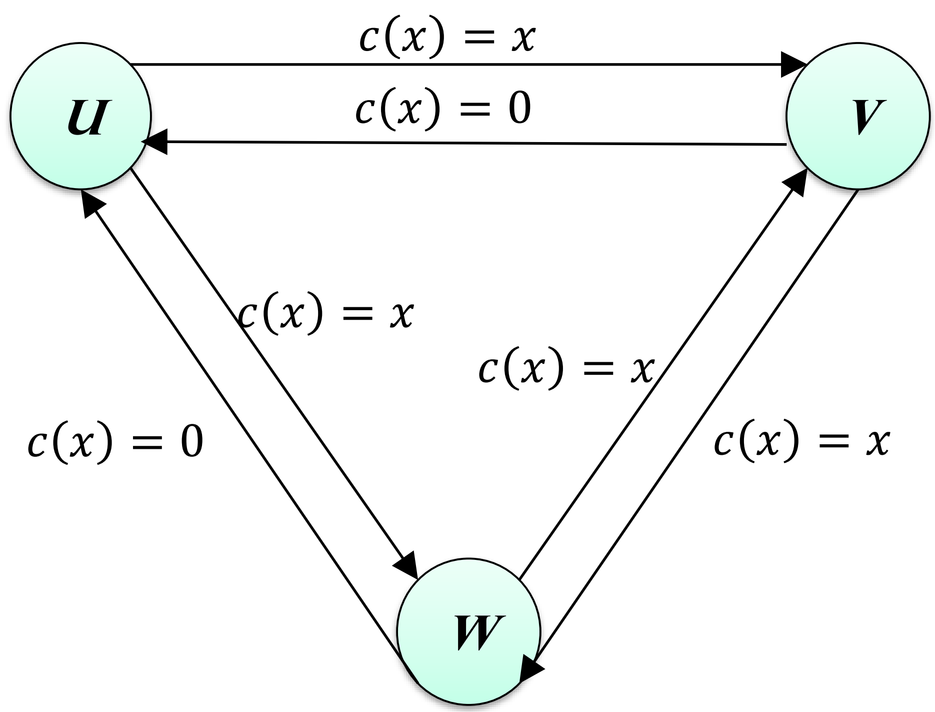
How bad is selfish routing?

More Known Results for Splittable flow:

- If the delay function of each edge is a linear function of the edge congestion then the price of anarchy is at most 4/3, and this is tight.
- If the delay functions assumed only to be non-decreasing, then the price of anarchy is unbounded.
- Many results for specific cost functions or specific network structure.

Unsplittable Flow - higher PoA

Consider the following network:

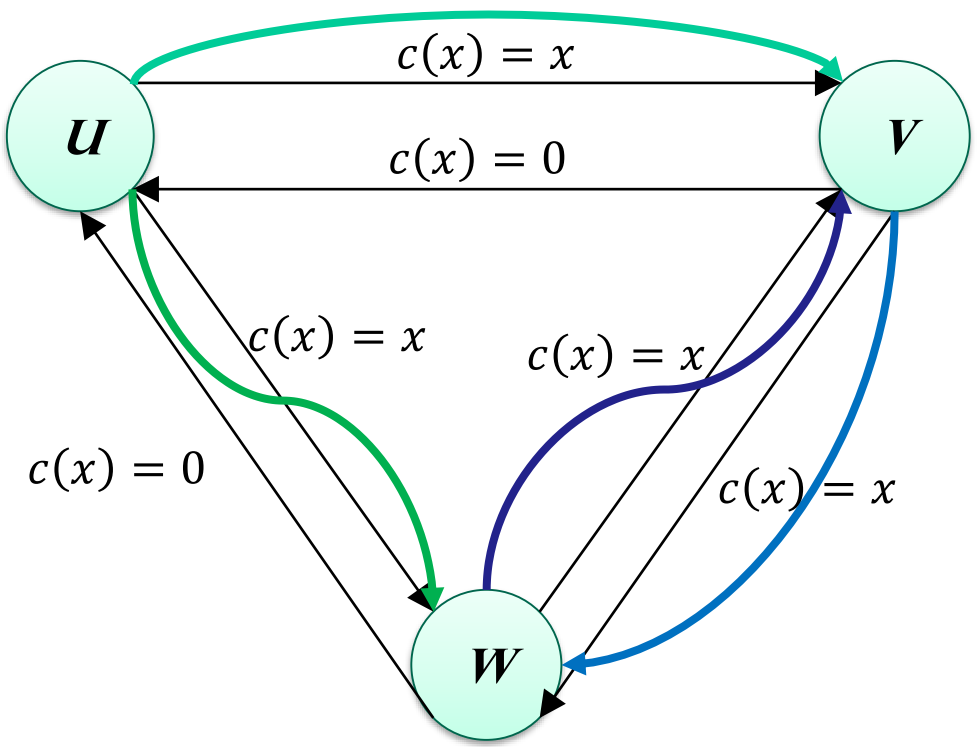


- There are 4 players with the following requests:

  (s_i, t_i): $\{(U,V); (U,W); (V,W); (W,V)\}$

Calculating OPT's Cost

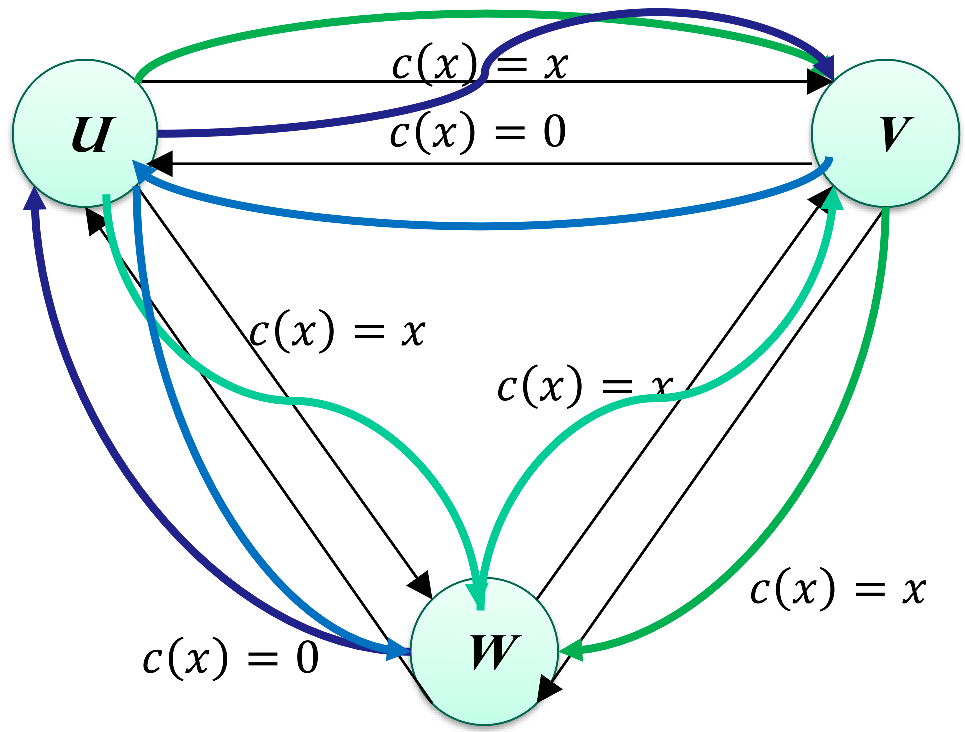
- OPT routes:
  - $(s_1, t_1) = (U, V)$
  - $(s_2, t_2) = (U, W)$
  - $(s_3, t_3) = (V, W)$
  - $(s_4, t_4) = (W, V)$



- Total Cost $= 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 = 4$

Calculating NE's Cost and PoA

- A possible NE
  - $(s_1, t_1) = (U, V)$
  - $(s_2, t_2) = (U, W)$
  - $(s_3, t_3) = (V, W)$
  - $(s_4, t_4) = (W, V)$

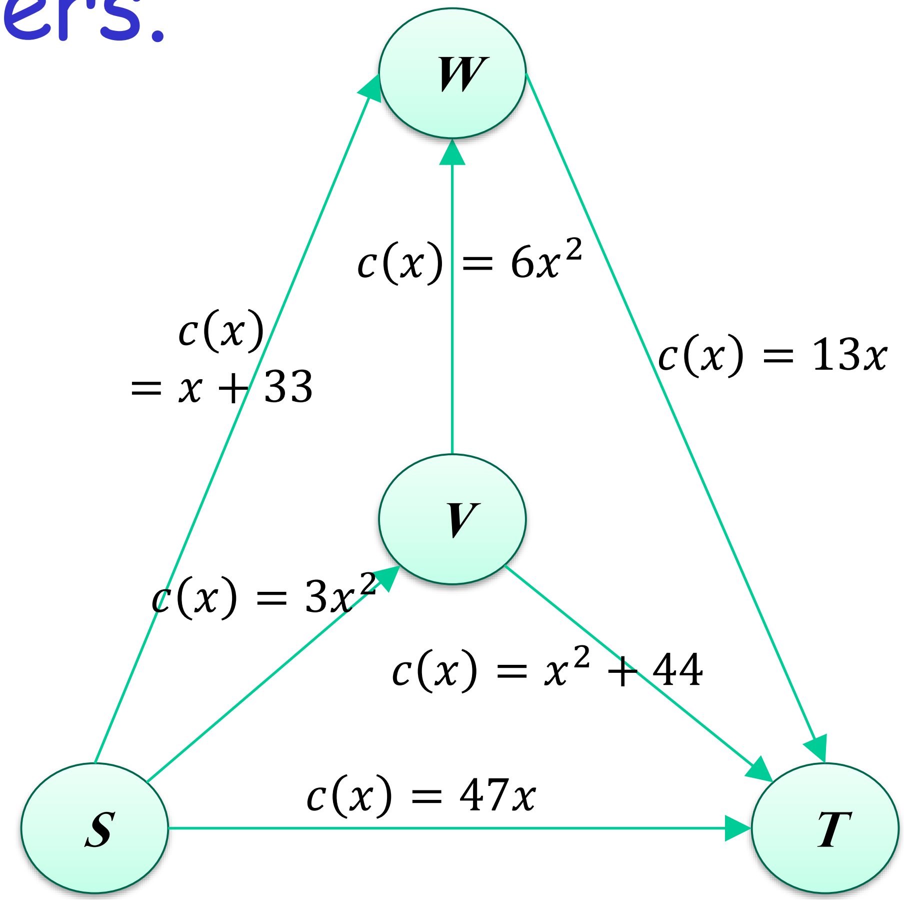


- $Total\ Cost = 2 \cdot 2 + 2 \cdot 2 + 1 \cdot 1 + 1 \cdot 1 = 10$

$$\Rightarrow PoA = \frac{10}{4} = 2.5$$

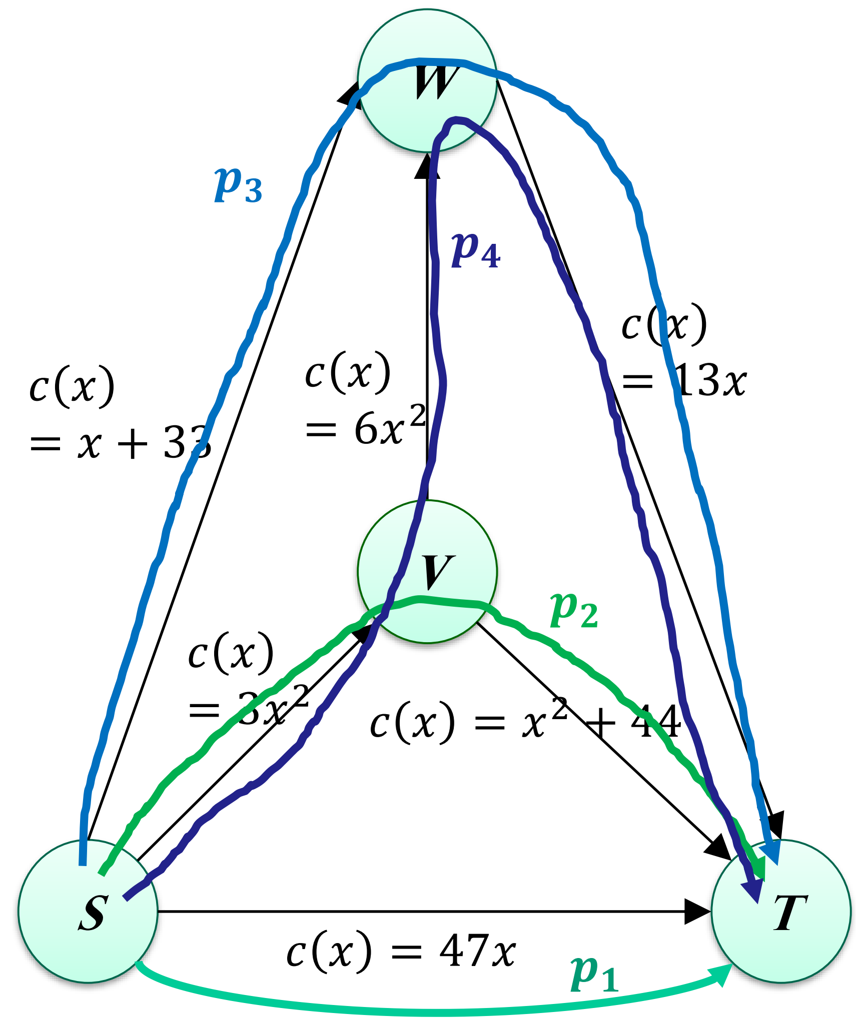
Unsplittable routing with weighted players.

- Let w_i denote the weight of player i.
- The loads and the payments are proportional to the weights.
  - Two players. Both need an (S,T)-path.
  - $w_1 = 1, w_2 = 2$



Unsplittable Routing with weighted players.

- There are four paths from S to T:
  - p_1: $S \rightarrow T$
  - p_2: $S \rightarrow V \rightarrow T$
  - p_3: $S \rightarrow W \rightarrow T$
  - p_4: $S \rightarrow V \rightarrow W \rightarrow T$



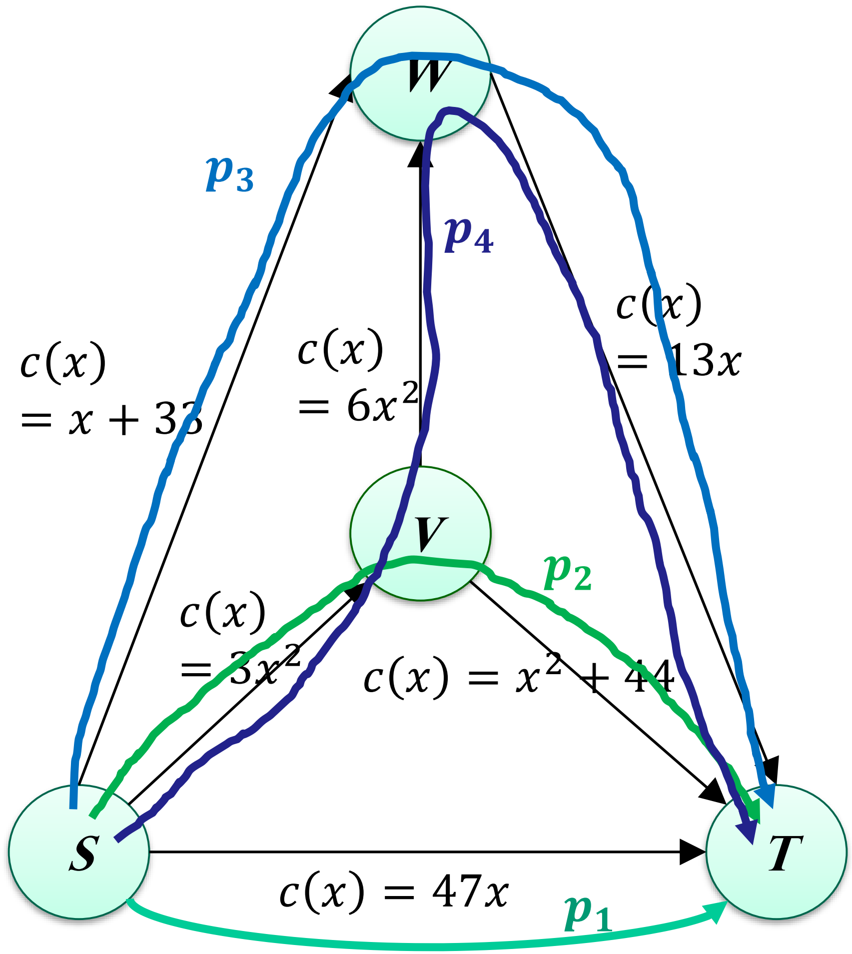
Weighted players. Cont.

p_1

p_2

p_3

p_4



$w_1 = 1, w_2 = 2$

Claim 1:

If Player 2 chooses either p_1 or p_2 then Player 1 will choose p_4

Weighted players. Cont.

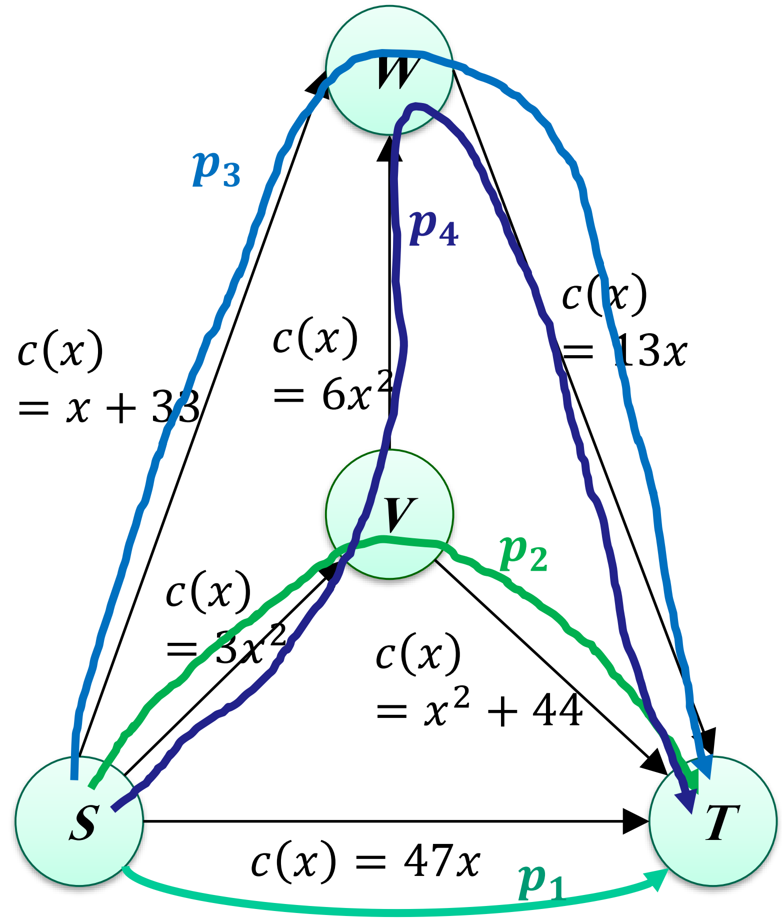
Proof of Claim 1:

Assuming player 2 chose path p_2, the cost of player 1 will be:

Path	Cost
p_1	47
p_2	$27 + 53 = 80$
p_3	$34 + 13 = 47$
p_4	$27 + 6 + 13 = 46$

Similarly, if player 2 choose path p_1, the cost of player 1 will be:

Path	Cost
p_1	141
p_2	$3 + 45 = 48$
p_3	$34 + 13 = 47$
p_4	$3 + 6 + 13 =22$



$w_1 = 1, w_2 = 2$

Weighted players. Cont.

Similarly, it is possible to show the following claims:

1. Player 2 chooses either p_1 or p_2 → player 1 will choose p_4
2. Player 1 chooses path p_4 → player 2 will choose p_3
3. Player 2 chooses either p_3 or p_4 → player 1 will choose p_1
4. Player 1 chooses path p_1 → player 2 will choose p_2

Corollary: There is no pure NE in this game.

In general: A pure NE may not exist in weighted congestion games.

Job Scheduling Games

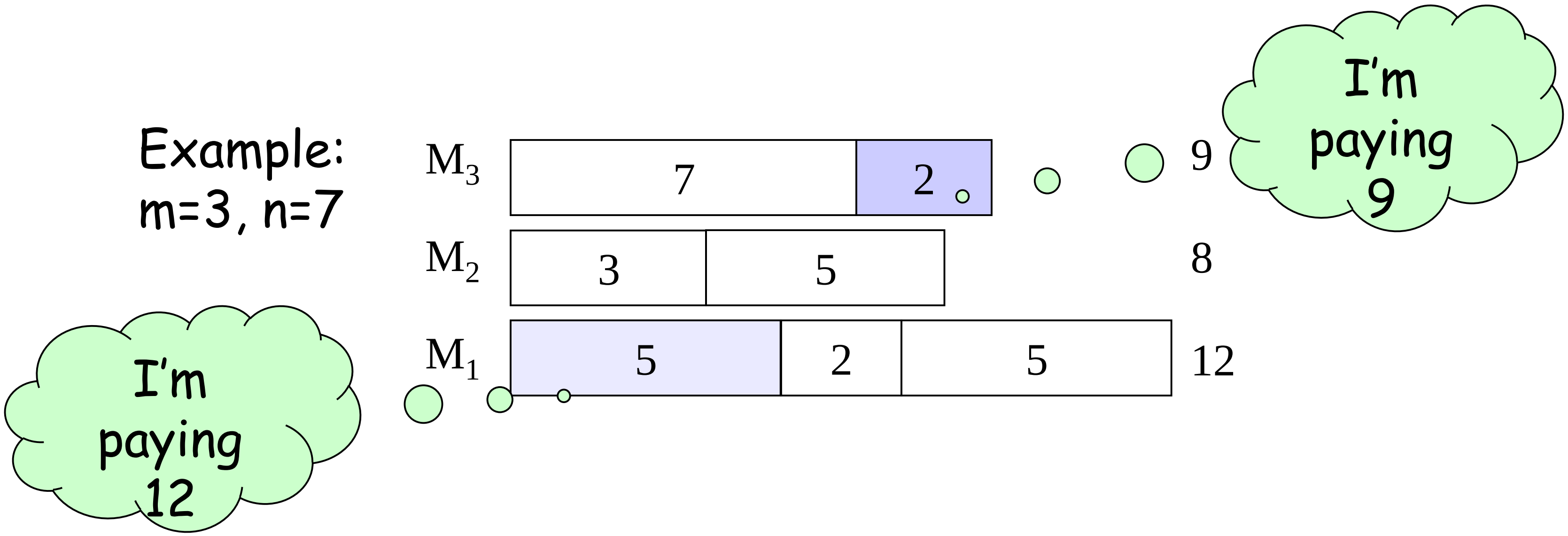
A weighted congestion game with singleton strategies.

- n jobs (the players).
- m machines (the resources).
- Each job has a length (will also be denoted weight or load)
- Each job represents a selfish agent who attempts to optimize its own objective.
- $\{A_i\}$ = set of machines.
- A profile is an assignment of the jobs to the machines.

Job Scheduling Games

- The cost of each job is the total load on the machine it is assigned to.

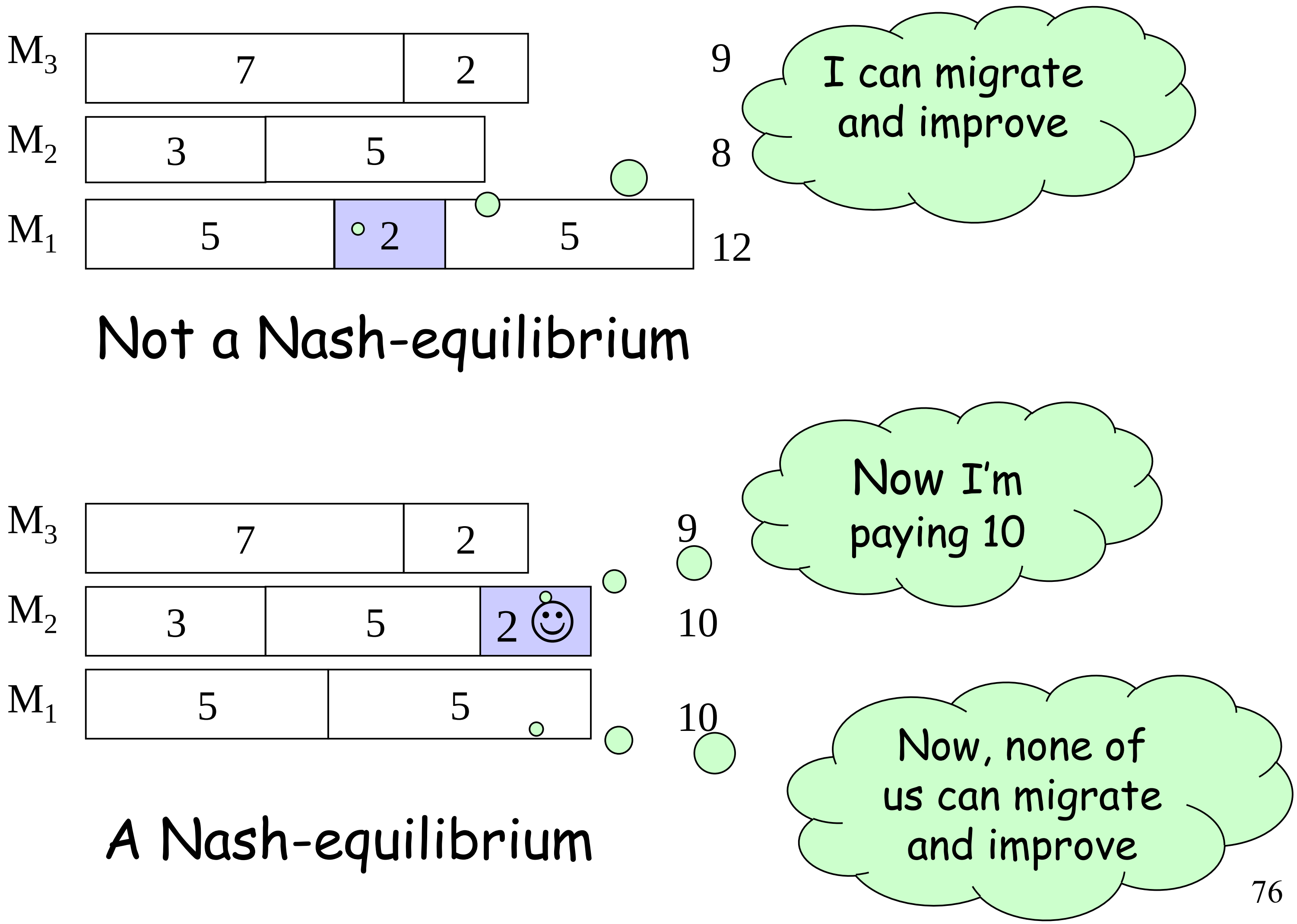
Example:
m=3, n=7



Note: In this payment scheme, the internal job order on each machine has no effect on the individual cost.

Motivation: routing on parallel links, round-robin, more

NE: no single-job migrations



NE: no single-job migrations

Notation:

L_i: Load on machine i (this is also the cost of all jobs assigned to i).

cost(j): cost of job j.

Theorem: NE always exist in job scheduling games, and can be found efficiently.

Proof: In class.

Equilibrium inefficiency

Social Optimum: A schedule in which the maximal cost of a job is minimized.
For a schedule s, $cost(s)=max_j\ cost(j)$.
This is equivalent to minimum makespan.

$$PoS = \frac{\text{Minimumum makespan in the best NE}}{\text{Minimumum makespan}}$$

Theorem: The Price of Stability in the job scheduling games is 1.
Proof: Note that a beneficial move does not increase the makespan. Therefore, by preforming best-response starting from any optimal assignment, we reach a NE whose makespan is equal to the optimum.

Equilibrium inefficiency

Theorem: The Price of Anarchy in the job scheduling game is $2 - \frac{2}{m+1}$.

Example (m=2): Consider the following assignment s on m=2 machines. s is a NE. its cost is 4.

M_2	2	2
M_1	1	1

This instance has an assignment with cost 3

M_2	2	1
M_1	2	1

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

$$\text{PoA} = \frac{\text{Minimumum makespan in the worst NE}}{\text{Minimumum makespan}}$$

Bounding the PoA

Theorem: The Price of Anarchy in the job scheduling game is $2 - \frac{2}{m+1}$.

Proof: Let s be any NE. Let i be the machine with the highest load (that is cost(s)=L_i) and let j be the shortest job on machine i.

If j is the only job on machine i then PoA=1 (why?).

Otherwise, $p_j \leq \frac{1}{2}$ cost(s).

Observation: For every machine i', $L_{i'} \geq L_i - p_j$

Therefore: $L_{i'} \geq L_i - p_j \geq L_i - \frac{1}{2} cost(s) = \frac{1}{2} cost(s)$.

Bounding the PoA

- So for every machine $i' \neq i$, $L_{i'} \geq \frac{1}{2}\,cost(s)$.

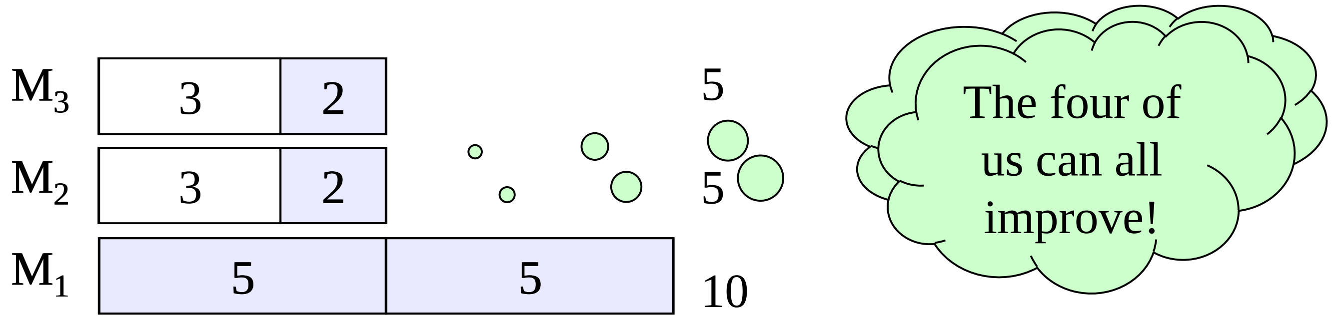
$$cost(OPT) \geq \frac{\sum_k pk}{m} = \frac{\sum_i L_i}{m} \geq \frac{cost(s) + (m-1)\frac{1}{2}cost(s)}{m} =$$

$$\frac{(m+1)cost(s)}{2m}.$$

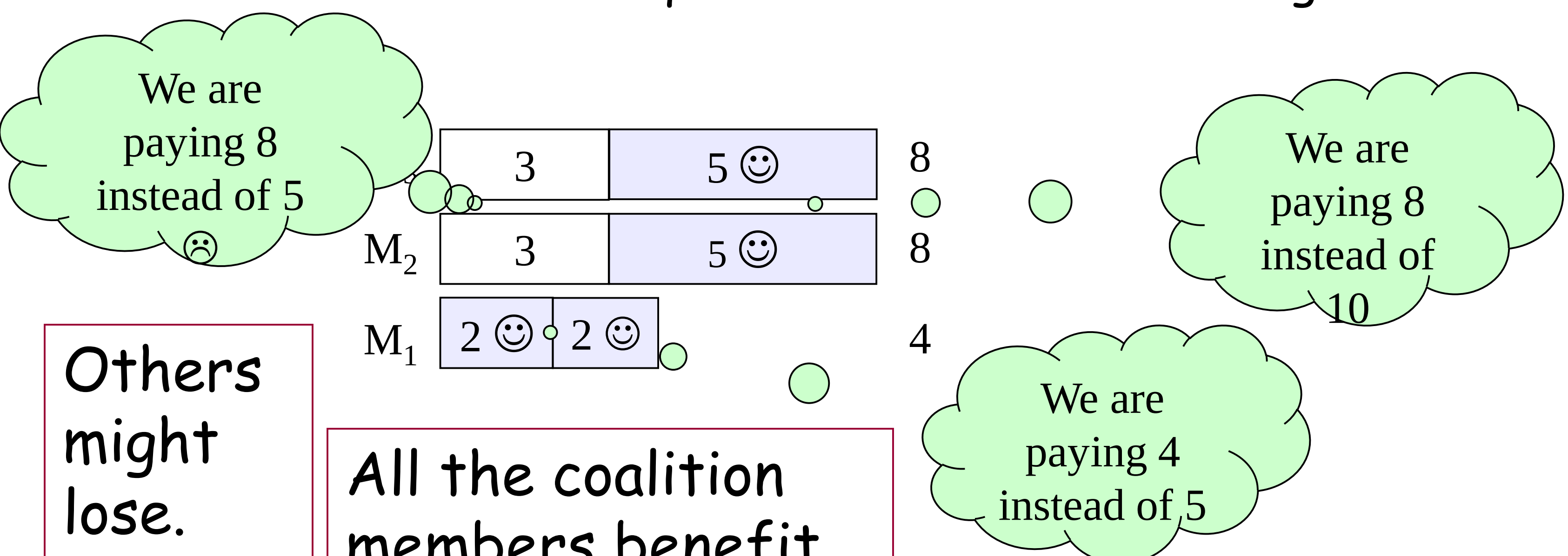
Therefore, $\frac{cost(s)}{cost(OPT)} \leq 2 - \frac{2}{m+1}$

- The analysis is tight (example of m=2 can be generalized).

SE: no coalition migrations



A Nash-equilibrium - But not a Strong NE.



Others might lose.

All the coalition members benefit.

NE vs. SE

Nash Equilibrium: no single player can deviate and improve its utility.

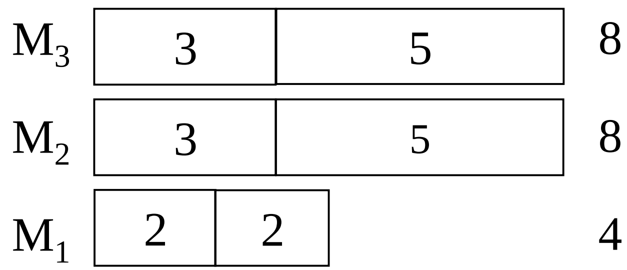
The Global Social Cost might not be achieved due to:

- Players' selfishness
- Lack of coordination

Strong Equilibrium [Aumann'59]: No coalition can deviate and **strictly** improve the utility of **all** of its members

- Separates the effect of selfishness from lack of coordination
- May be a better prediction of rational behavior
- Most games do not admit Strong Eq.

SE: no coalition migrations



Theorem: SE always exists in job scheduling (even for unrelated machines) [AFM07]

Proof: In class.

Does a NE approximate SE?

In the example, each of the coalition jobs improves by a factor of 5/4 (10→8, 5→4)

- Is there a bound on the improvement ratio exhibited by all coalition member?
- Is there a bound on the improvement ratio exhibited by a single coalition member?

In the example, the cost of some jobs is increased by a factor of 8/5 (5→8)

- Is there a bound on the damage ratio exhibited by some non coalition member job?

Evaluating approximate SE

Given a configuration s, and a coalition Γ, we consider 3 measurements:

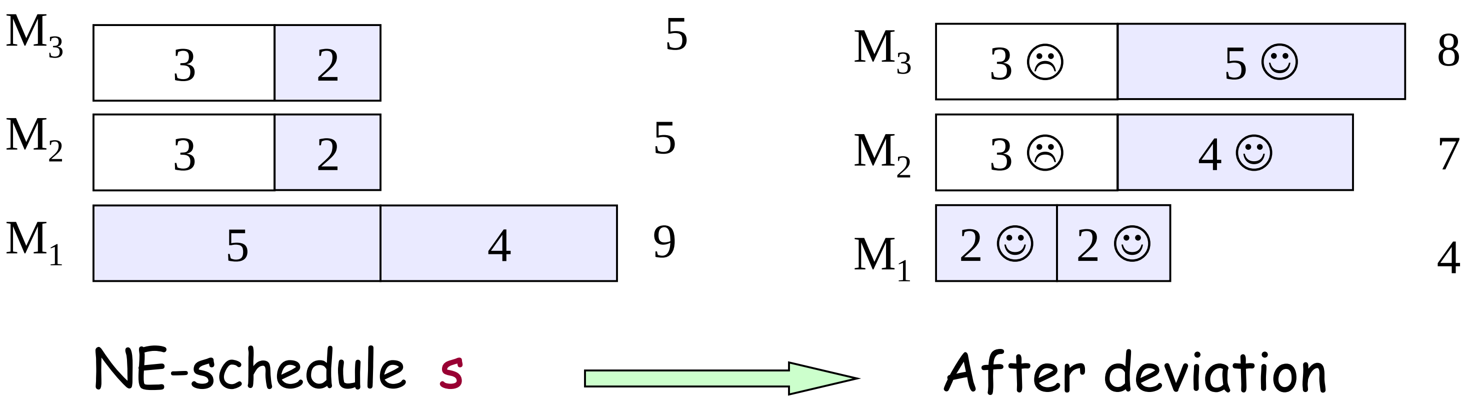
1. Minimum Improvement Ratio: $IR_{min}(s,\Gamma)$ is the minimal improvement ratio of some job in Γ.

s is α-SE if there is no coalition Γ for which $IR_{min}(s,\Gamma) > \alpha$.

2. Maximum Improvement Ratio: $IR_{max}(s,\Gamma)$ is the maximal improvement ratio of some job in Γ.

3. Maximum Damage Ratio: $DR_{max}(s,\Gamma)$ is the maximal damage ratio of some job not in Γ.

Evaluating approximate SE



Consider the coalition Γ={5,4,2,2,}

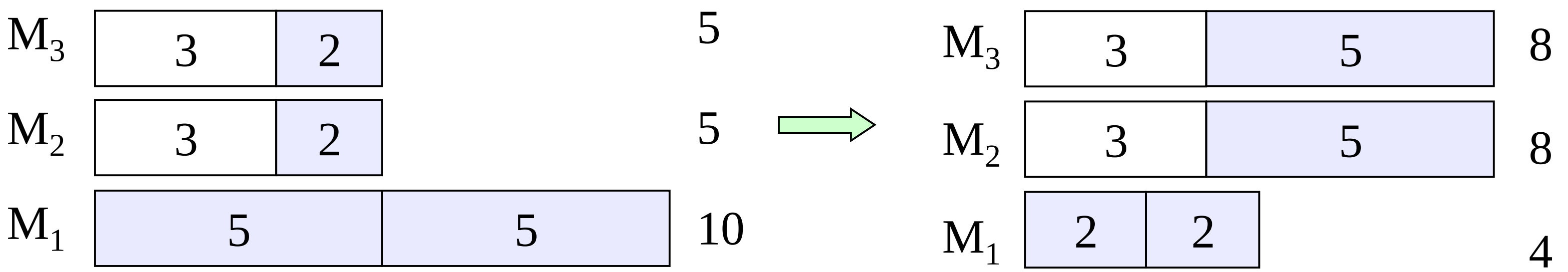
$IR_{min}(s,\Gamma) = 9/8=1.125$

$IR_{max}(s,\Gamma) = 9/7\approx1.28$

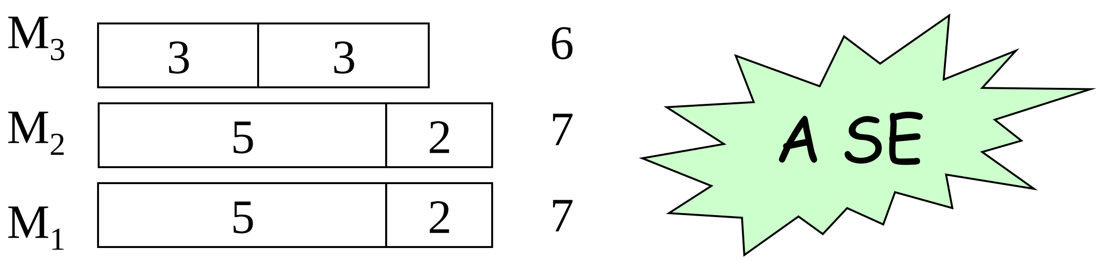
$DR_{max}(s,\Gamma) = 8/5 = 1.6$

LPT vs. any-NE

In our example, the initial configuration is a NE.



The same set of jobs under LPT:



Known Results

	IR_{min}			IR_{max}		DR_{max}	
	Upper bound		Lower bound	Upper bound	Lower bound	Upper bound	Lower bound
	m=3	m≥3					
NE	5/4	$2-\frac{2}{m+1}$	5/4	unbounded		2	2
LPT	$\frac{1}{2}+\frac{\sqrt{6}}{4}$	$\frac{4}{3}-\frac{1}{3m}$	$\frac{1}{2}+\frac{\sqrt{6}}{4}$	5/3 (m=3)	$2-\frac{1}{m}$	3/2	3/2

~1.12

In any schedule produced by LPT, no coalition can improve the cost of all its members by ratio > $\frac{1}{2}+\frac{\sqrt{6}}{4}$ and this is tight.

In any NE schedule, no coalition-move can increase the cost of some job by a factor > 2, and this is tight.