Price of Anarchy and Coordination Mechanisms

Selfishness and how to cope with it

Elias Koutsoupias

Department of Informatics and Telecommunications
University of Athens

k-Implementation

Monterer, Tennenholtz

EC 03

A simple example

A game with 2 players and 2 strategies where the cost of each player is

2	1
1	2

A simple example

A game with 2 players and 2 strategies where the cost of each player is

2	1
1	2

- Suppose that we want the first player to play the first strategy and the second player to play the second strategy. How can we induce the players to do it?

A simple example

A game with 2 players and 2 strategies where the cost of each player is

2	1
1	2

- Suppose that we want the first player to play the first strategy and the second player to play the second strategy. How can we induce the players to do it?
- Idea: We, a reliable third party, promise to pay each player some amounts that depend on the selected strategy profile. How to select the amounts to achieve the desired outcome and end up paying the smallest possible amount?

Example (cont.)

In the example, we promise to give to the first player 2 euro if both players select the first strategy. Similarly, we promise to give to the other player 2 euro if both players select the second strategy.
The new costs are

0	1
1	2

2	1
1	0

Example (cont.)

In the example, we promise to give to the first player 2 euro if both players select the first strategy. Similarly, we promise to give to the other player 2 euro if both players select the second strategy.
The new costs are

0	1
1	2

2	1
1	0

- The desired strategies are now dominant.

Example (cont.)

In the example, we promise to give to the first player 2 euro if both players select the first strategy. Similarly, we promise to give to the other player 2 euro if both players select the second strategy.
The new costs are

0	1
1	2

2	1
1	0

- The desired strategies are now dominant.
- We don't have to pay anything!

k-Implementation

Given a game and a desired set of strategy profiles, we ask whether we can pay the players k units to make a desired strategy profile dominant.

k-Implementation

Given a game and a desired set of strategy profiles, we ask whether we can pay the players k units to make a desired strategy profile dominant.

Theorem: In finite games with cost function c*, every strategy profile* z *has an optimal implementation when we pay*

$$k(z) = \sum_{i=1}^{n} \max_{x_i} \left[c_i(z_i, z_{-i}) - c_i(x_i, z_{-i}) \right]$$

k-Implementation

Given a game and a desired set of strategy profiles, we ask whether we can pay the players k units to make a desired strategy profile dominant.

Theorem: In finite games with cost function c, every strategy profile z has an optimal implementation when we pay

$$k(z) = \sum_{i=1}^{n} \max_{x_i} \left[c_i(z_i, z_{-i}) - c_i(x_i, z_{-i}) \right]$$

Theorem: A strategy profile has a 0-implementation if and only if it is a Nash equilibrium.

k-Implementation

As we saw, implementing a single strategy profile is easy. When we want to implement a set of desired strategy profiles, the problem becomes NP-complete:

Theorem: Given a game and a set of desired strategy profiles, finding a k-implementation which implements one of the strategies and has minimum payments is NP-hard.

Taxes

R. Cole, Y. Dodis, and T. Roughgarden

STOC 03, EC 03

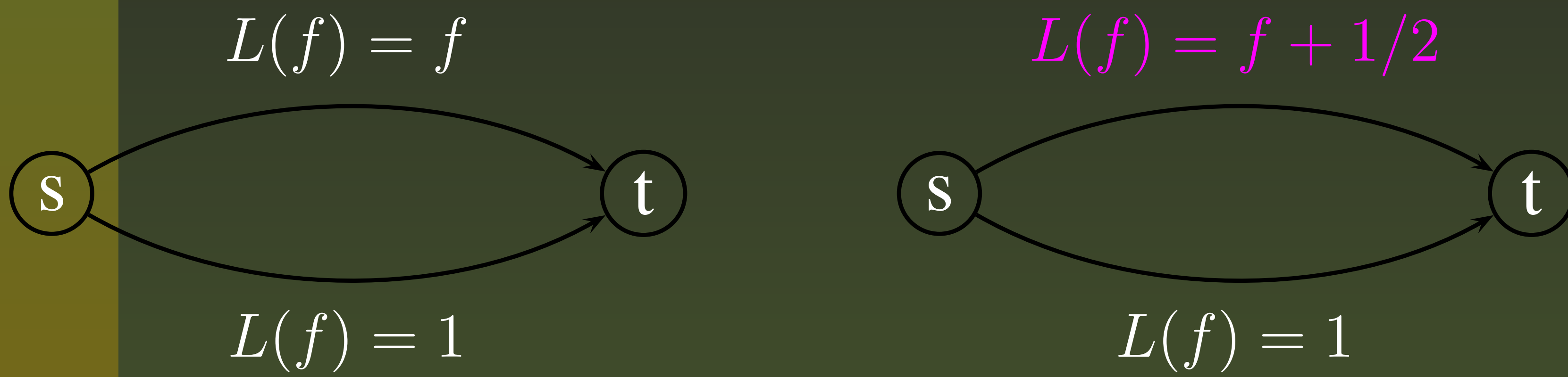
Marginal cost taxes

- Marginal cost tax: Each player pays tax equal to the additional delay other players experience because of his presence. [Familiar? VCG mechanism]
- More precisely, for an edge e with delay function L_e, the tax is

  $$\tau = f_e \cdot L'(f_e),$$

  where L' is the derivative of L and f_e is the optimal flow. This makes the new delay on edge e: $L(f) + f_e \cdot L'(f_e)$.

Marginal cost taxes - Example



- On the left, a network before taxes. On the right, the same network with marginal cost tax (when the rate is 1).
- Left Nash equilibrium, a flow of 1 uses the upper edge. $Cost = 1$.
- Right Nash equilibrium, a flow of 1 uses both edges (1/2 and 1/2). $Cost = 1/2 + 1/2 = 1$.

Marginal cost taxes

Theorem: [*Latency only*] *For every network with continuous differentiable delay functions, the marginal cost tax achieves minimum (optimal) latency.*

Marginal cost taxes

Theorem: [Latency only] For every network with continuous differentiable delay functions, the marginal cost tax achieves minimum (optimal) latency.

Theorem: [Latency + Tax] For networks with linear delay functions, the marginal cost tax does not improve the cost, i.e. latency+tax $\geq$ latency before taxes.

Tax - Braess' Network

- For flow 1, the network on the left has cost (latency+taxes) 2 and all the flow goes through the 0 latency edge.
- For the same flow the network on the right has cost 3/2 (the flow splits 1/2 and 1/2).
- The Price of Anarchy improves from 4/3 to 1.
- This is not marginal cost tax.

Taxes - Other results

- Taxes sometimes improve the Price of Anarchy (when cost=latency+tax).

Taxes - Other results

- Taxes sometimes improve the Price of Anarchy (when cost=latency+tax).
- For every linear network, the optimal tax on each edge is either 0 or infinity (equivalent to the removal of the edge).

Taxes - Other results

- Taxes sometimes improve the Price of Anarchy (when cost=latency+tax).
- For every linear network, the optimal tax on each edge is either 0 or infinity (equivalent to the removal of the edge).
- Computing the optimal taxes is NP-hard (because computing the optimal removals is also NP-hard [Roughgarden]).