Algorithmic Game Theory

Quick notes

Lorenzo Laneve

Contents

1 Strategic Games and Nash Equilibria **3**
- 1.1 Introduction . . . 3
- 1.2 Best response . . . 3
- 1.3 Potential games . . . 3
- 1.4 Congestion games . . . 5
- 1.5 Convergence time of best response . . . 5

2 Price of Anarchy **7**
- 2.1 Smoothness . . . 7
- 2.2 Affine delay functions . . . 8

3 Hardness of Nash Equilibria **9**
- 3.1 Polynomial-time algorithm for symmetric network games . . . 9
- 3.2 The complexity class PLS . . . 9
- 3.3 PLS-hardness and max cut . . . 10

4 Mixed and Correlated Equilibria **12**

5 Regret Minimization **14**
- 5.1 Online learning from experts . . . 14
- 5.2 External regret . . . 15
- 5.3 Multiplicative Weights Update . . . 15
- 5.4 Computing a coarse correlated equilibrium . . . 16

6 Price of Stability **18**

7 Mechanisms with Money **20**
- 7.1 Setting (cost version) . . . 20
- 7.2 Setting (utility version) . . . 20
- 7.3 Truthfulness and VCG mechanism . . . 21
- 7.4 One-parameter mechanisms . . . 22
- 7.5 Selfish related machines . . . 24
- 7.6 Selfish unrelated machines . . . 24
- 7.7 Combinatorial auctions . . . 26
- 7.8 Greedy approximation: greedy-by-value . . . 27
- 7.9 Greedy approximation: greedy-by-value/density . . . 28

8 Voting Systems **29**
- 8.1 Examples of voting systems . . . 29
- 8.2 Arrow's theorem . . . 30
- 8.3 Single-peaked preferences . . . 31

9 Mechanisms without Money **32**
9.1 House allocation 32
9.2 Kidney exchange 33
9.3 Stable matching 33

10 Best Response Mechanisms **36**
10.1 Asynchronous best-response dynamics 36
10.2 Never best response and convergence 36
10.3 Clear outcome and incentive compatibility 37
10.4 Gao-Rexford model in BGP 38
10.5 TCP games 39
10.6 Stable matching 40
10.7 Single item auction 41

11 Sponsored Search **42**
11.1 VCG vs GSP 42
11.2 Nash equilibria in GSP 43
11.3 Bounds on revenues for GSP 45

1 Strategic Games and Nash Equilibria

1.1 Introduction

Definition 1.1 (Strategic game)**.** *A strategic game is a tuple $G = (n, S, u)$ where:*

- *n is the number of players;*
- *$S = S_1 \times \cdots \times S_n$ is the set of states of the game, composed of strategies $s_i \in S_i$ of each player i;*
- *$u = (u_1, \dots, u_n)^T : S \to \mathbb{R}^n$ contains the utility functions of the players.*

The goal of player i is to maximize the utility u_i by choosing only a strategy $s_i \in S_i$ (given the choice of the other players). Equivalently we can define a game $G = (n, S, c)$ where $c = -u$ are players' costs (that should be minimized).

Definition 1.2. *A state $s^* \in S$ is said to be a pure Nash equilibrium for $G = (n, S, u)$ if, for any player i and any alternative strategy $s_i \in S_i$:*

$$u_i(s^*) \geq u_i(s_i, s^*_{-i})$$

i.e. no player can increase their utility by changing strategy.

1.2 Best response

Definition 1.3 (Best response)**.** *In a game $G = (n, S, u)$, $\bar{s}_i \in S_i$ is said to be a best response for i to a state s if this maximizes their utility:*

$$u_i(\bar{s}_i, s_{-i}) = \max_{s \in S_i} u_i(s, s_{-i})$$

A best response might not be necessarily unique.

Algorithm 1.4 (Best response dynamics)**.** *Starting with an arbitrary state $s \in S$:*

- *While a player can strictly improve its utility, pick i as such player, and set*

$$s \leftarrow (\bar{s}_i, s_{-i})$$

where $\bar{s}_i$ is one of i's best response to s, and repeat.

Theorem 1.5. *The best response dynamics always converges to a pure Nash equilibrium. In particular, if a game does not admit a pure Nash equilibrium, the game will not converge.*

Proof. Let s^* the output of the algorithm. One can see that the termination condition of the algorithm yields exactly the definition of pure Nash equilibrium on s^*. □

1.3 Potential games

Definition 1.6. *A game $G = (n, S, u)$ is said to be a potential game if there exists a function $f : S \to \mathbb{R}$ such that:*

$$f(s) - f(s'_i, s_{-i}) = u_i(s) - u_i(s'_i, s_{-i})$$

for every i. Such f is said to be a potential for G.

The definition also applies for cost games: if f is a potential function for the cost game, then $-f$ is a potential function for the utility game.

Example 1.7. *The following game is a potential game, where Φ as defined on the right is a possible potential function. Notice that also $\Phi + C$ is a valid potential function, for any $C \in \mathbb{R}$.*

	A	*B*
A	*1*, *2*	*0*, *0*
B	*0*, *0*	*2*, *1*

$\Longrightarrow$

Φ	*A*	*B*
A	*0*	*-1*
B	*-2*	*0*

Lemma 1.8. *In a potential game under best response dynamics, any potential function f strictly increases at each step.*

Proof. Switching to a best response for a player i at step k means that player i is strictly increasing its utility u_i, i.e.:

$$f(s_{k+1}) - f(s_k) = u_i(s_{k+1}) - u_i(s_k) > 0$$

which means that f strictly increases. □

Theorem 1.9. *Any potential game admits a pure Nash equilibrium.*

Proof. Suppose for a contradiction that the best response dynamics cycles, i.e. we have states $s_1, \dots, s_\ell \in S$ such that the algorithm passes through:

$$s_1 \to \cdots \to s_\ell \to s_1$$

which contradicts Lemma 1.8. □

Theorem 1.10. *A state $s \in S$ locally optimizes a potential function if and only if it is a pure Nash equilibrium.*

Proof. Consider without loss of generality utility games, and suppose for a contradiction we have a pure Nash equilibrium s such that $f(s) < f(s'_i, s_{-i})$ for some player i (i.e. s is not a local optimum for the potential function f). By definition of potential:

$$f(s) < f(s'_i, s_{-i}) \Longrightarrow u_i(s) < u_i(s'_i, s_{-i})$$

implying that s is not a pure Nash equilibrium. Conversely, if s is a local optimum, best response dynamics starting in s immediately terminates (in order to go ahead with the algorithm, one should strictly increase the potential function, by Lemma 1.8), and we already know that best response dynamics terminates only on pure Nash equilibria by Theorem 1.5. □

Definition 1.11 (Social welfare)**.** *The social welfare $SW(s)$ is defined as the sum of all players' utilities:*

$$SW(s) = \sum_i u_i(s)$$

Example 1.12. *It is not always true that a best response increases the social welfare, not even in a potential game.*

	A	*B*
A	*5*, *-2*	*2*, *-5*
B	*-1*, *-1*	*0*, *0*

$\Longrightarrow$

Φ	*A*	*B*
A	*5*	*2*
B	*-1*	*0*

Switching from BB to AB (best response of player 1 *to BB), changes the social welfare from* 0 *to* -3.

1.4 Congestion games

Definition 1.13 (Congestion game). *A congestion game is a game $G = (n, S, c)$ considering m resources, where:*

$$S_i \subseteq \mathbf{2}^{[m]}$$

i.e. a player chooses a subset of resources (not all subsets may be allowed to every player). Players' costs are defined as:

$$c_i(s) = \sum_{r \in s_i} d_r(n_r(s))$$

where $n_r(s)$ is the number of players using resource r in state s, and $d_r(n)$ is the delay function of resource r, telling the cost charged by r when used by n players.

Example 1.14 (Network congestion game). *A congestion game with a graph on m edges, containing two vertices s, t. A strategy is an s-t path.*

Theorem 1.15. *Congestion games are potential games. In particular they admit a pure Nash equilibrium, and best response dynamics always converges.*

Proof. Consider the Rosenthal function:

$$\Phi(s) = \sum_r \sum_{k=1}^{n_r(s)} d_r(k)$$

We claim that this is indeed a potential function for congestion games. Suppose player i changes strategy, and this causes a change of cost Δc_i. One can see that the Rosenthal function can be rewritten as:

$$\begin{aligned}
\Phi(s) &= \sum_{r \notin s_i} \sum_{k=1}^{n_r(s)} d_r(k) + \sum_{r \in s_i} \sum_{k=1}^{n_r(s)} d_r(k) \\
&= \sum_{r \notin s_i} \sum_{k=1}^{n_r(s)} d_r(k) + \sum_{r \in s_i} \sum_{k=1}^{n_r(s)-1} d_r(k) + \sum_{r \in s_i} d_r(n_r(s)) \\
&= \sum_{r \notin s_i} \sum_{k=1}^{n_r(s)} d_r(k) + \sum_{r \in s_i} \sum_{k=1}^{n_r(s)-1} d_r(k) + c_i(s)
\end{aligned}$$

Since changing only i's strategy does not change the first two terms, the function changes only in the last term, and it changes exactly by Δc_i, proving the Rosenthal's function is a potential function for congestion games. □

1.5 Convergence time of best response

Definition 1.16 (Singleton congestion game). *A congestion game is called singleton if $S_i \subseteq [m]$, i.e. each player can choose only a single resource.*

Theorem 1.17. *Best response dynamics in a singleton congestion game converges in $\mathcal{O}(n^2 m)$ steps.*

Proof. We sort the $d_r(k)$'s in increasing order, for $k \in [n], r \in [m]$, and we define $\bar{d}_r(k)$ as the rank of $d_r(k)$ in the order. Notice that, in particular, $0 \leq d_r(k) \leq nm$.

Replacing $d_r(k)$ with $\bar{d}_r(k)$ in the game preserves best responses: if, in state s, resource r is a best response for player i, then $d_r(n_r(s))$ is the lowest among the possible choices, and so it will appear before other resources in the order defined above, i.e. $\bar{d}_r(n_r(s))$ is also the lowest.

This means that the best response dynamics will be the same in both games. We conclude the proof by bounding the number of steps in the altered game: let $\bar{\Phi}$ the potential function in the altered game:

$$\bar{\Phi}(s) = \sum_r \sum_{k=1}^{n_r(s)} \bar{d}_r(k) \leq \sum_r \sum_{k=1}^{n_r(s)} nm = nm \sum_r n_r(s) = n^2 m$$

since $\sum_r n_r(s) = n$, i.e. each player takes exactly one resource. Since each step decreases $\bar{\Phi}$ by at least 1, and $\bar{\Phi}$ itself is non-negative, we converge in at most n^2m steps. $\square$

2 Price of Anarchy

Definition 2.1 (Price of Anarchy)**.** *Given a game $G = (n, G, u)$, the price of anarchy is defined as:*

$$PoA(G) = \frac{\max_{s \in S} SW(s)}{\min_{s \in PNE(G)} SW(s)}$$

or, equivalently, for cost games:

$$PoA(G) = \frac{\max_{s \in PNE(G)} cost(s)}{\min_{s \in S} cost(s)}$$

where $PNE(G) \subseteq S$ is the set of pure Nash equilibria of G and $cost(s) = \sum_i c_i(s)$.

The price of anarchy gives the approximation guarantee of an equilibrium.

2.1 Smoothness

Definition 2.2. *A game $G = (n, S, u)$ is (λ, μ)-smooth if the following holds for any pair of states $s, s^* \in S$:*

$$\sum_i u_i(s_i^*, s_{-i}) \geq \lambda \cdot SW(s^*) - \mu \cdot SW(s)$$

or equivalently, for cost games:

$$\sum_i c_i(s_i^*, s_{-i}) \leq \lambda \cdot cost(s^*) + \mu \cdot cost(s)$$

Theorem 2.3. *If G is (λ, μ)-smooth, then*

- *$PoA(G) \leq \frac{\lambda}{1-\mu}$ for cost games;*
- *$PoA(G) \leq \frac{1+\mu}{\lambda}$ for utility games.*

Proof. Let s be an equilibrium and s^* a global optimum. For cost games:

$$\begin{aligned} cost(s) &= \sum_i c_i(s) \\ &\leq \sum_i c_i(s_i^*, s_{-i}) && \text{by definition of equilibrium} \\ &\leq \lambda \cdot cost(s^*) + \mu \cdot cost(s) \end{aligned}$$

implying $\frac{cost(s)}{cost(s^*)} \leq \frac{\lambda}{1-\mu}$. For utility games:

$$\begin{aligned} SW(s) &= \sum_i u_i(s) \\ &\geq \sum_i u_i(s_i^*, s_{-i}) && \text{by definition of equilibrium} \\ &\geq \lambda \cdot SW(s^*) - \mu \cdot SW(s) \end{aligned}$$

implying $\frac{SW(s^*)}{SW(s)} \leq \frac{1+\mu}{\lambda}$. □

2.2 Affine delay functions

Definition 2.4. *A congestion game is said to have affine delay functions if:*

$$d_r(n) = a_r n + b_r$$

for every resource $r \in [m]$.

Lemma 2.5. *$y(z+1) \le \frac{5}{3}y^2 + \frac{1}{3}z^2$ for all $y, z \in \mathbb{N}$.*

Proof. If $y = 1$, then we have:

$$z + 1 \le \frac{5}{3} + \frac{1}{3}z^2 \iff z^2 - 3z + 2 \ge 0$$

For $z = 0, 1$ one can see that it holds. for $z = 2$ we have $z^2 + 2 = 3z$, then z^2 will grow faster. If $y \ge 2$ then one can see that:

$$\frac{3}{4}y^2 + \frac{1}{3}z^2 - yz = \left(y\sqrt{\frac{3}{4}} - z\sqrt{\frac{1}{3}}\right)^2 \ge 0 \Longrightarrow yz \le \frac{3}{4}y^2 + \frac{1}{3}z^2$$

Thus, along with the fact that $y \le y^2/2$ for $y \ge 2$

$$y(z+1) = yz + y \le \frac{3}{4}y^2 + \frac{1}{3}z^2 + \frac{1}{2}y^2 = \frac{5}{4}y^2 + \frac{1}{3}z^2 \le \frac{5}{3}y^2 + \frac{1}{3}z^2$$

□

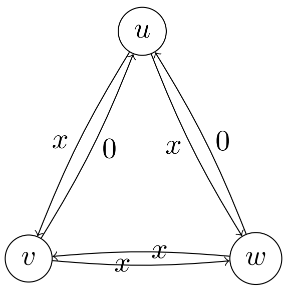
Theorem 2.6. *Congestion games with affine delay functions are $(\frac{5}{3}, \frac{1}{3})$-smooth.*

Proof. For any two states $s, s^* \in S$:

$$\begin{aligned}
\sum_i c_i(s_i^*, s_{-i}) &= \sum_i \sum_{r \in s_i} d_r(n_r(s_i^*, s_{-i})) \\
&\le \sum_i \sum_{r \in s_i} d_r(n_r(s) + 1) && n_r \text{ is 1-Lipschitz} \\
&= \sum_r \sum_{i: r \in s_i^*} d_r(n_r(s) + 1) \\
&= \sum_r n_r(s^*) \cdot d_r(n_r(s) + 1) \\
&= \sum_r a_r(n_r(s) + 1)n_r(s^*) + b_r n_r(s^*) \\
&\le \sum_r a_r \left(\frac{1}{3}n_r^2(s) + \frac{5}{3}n_r^2(s^*)\right) + b_r n_r(s^*) && \text{by Lemma 2.5} \\
&\le \sum_r a_r \left(\frac{1}{3}n_r^2(s) + \frac{5}{3}n_r^2(s^*)\right) + \frac{5}{3}b_r n_r(s^*) + \frac{1}{3}b_r n_r(s) \\
&\le \sum_r \frac{1}{3}n_r(s)(a_r n_r(s) + b_r) + \frac{5}{3}n_r(s^*)(a_r n_r(s^*) + b_r) \\
&= \frac{1}{3}\sum_r n_r(s)(a_r n_r(s) + b_r) + \frac{5}{3}\sum_r n_r(s^*)(a_r n_r(s^*) + b_r) \\
&= \frac{1}{3}\sum_r \sum_{i: r \in s_i^*} (a_r n_r(s) + b_r) + \frac{5}{3}\sum_r \sum_{i: r \in s_i^*} (a_r n_r(s^*) + b_r) \\
&= \frac{1}{3}\sum_i \sum_{r \in s_i^*} (a_r n_r(s) + b_r) + \frac{5}{3}\sum_i \sum_{r \in s_i^*} (a_r n_r(s^*) + b_r) \\
&= \frac{1}{3}cost(s) + \frac{5}{3}cost(s^*)
\end{aligned}$$

□

Consider an asymmetric network congestion game (source and target are not the same for every one).



where $d_r(x) = x$ for the edges with label x. We define four players:

- Player 1 goes $u \to v$;
- Player 2 goes $u \to w$;
- Player 3 goes $v \to w$;
- Player 4 goes $w \to v$.

If everyone chose the direct edge, then we would have an optimal total cost of 4. However, there is a pure Nash equilibrium where everyone chooses a path of length 2, yielding a total cost of 10. Thus the price of anarchy here is $\geq \frac{10}{4} = \frac{5}{2}$. This proves that the bound is tight and the price of anarchy for congestion games with affine delay functions is exactly 5/2.

3 Hardness of Nash Equilibria

3.1 Polynomial-time algorithm for symmetric network games

Theorem 3.1. *Pure Nash equilibria can be computed in polynomial time for symmetric network congestion games with non-decreasing delays.*

Proof. Given the graph, copy each edge n times, where the k-th copy of the edge r has capacity 1 and cost $d_r(k)$. We claim that a min-cost max flow (we bound the flow with n using an edge connecting the real source to s) here yields a minimum on the Rosenthal potential function (which we know to be surely a pure Nash equilibrium). Notice that each state $s \in S$ corresponds to a flow, and for a min cost flow the cost is exactly the value of the Rosenthal function: in fact, since delays are non-decreasing, if k players pass through an edge, then the min cost flow will take the k lightest edges (otherwise we could find a flow with strictly smaller cost by switching to this choice). Then, letting s^* be a min cost flow, suppose for a contradiction that we have $s \in S$ with $\Phi(s) < \Phi(s^*)$: since s corresponds to a flow, this would have strictly smaller cost, contradicting the optimality of s^*. Notice that the value of the flow is always n for any state. □

3.2 The complexity class PLS

Definition 3.2. *A local search problem* Π *consists of a set of instances* $\mathcal{I}_\Pi$*, where each instance* $I \in \mathcal{I}_\Pi$ *has:*

- *A finite set of feasible solutions* $\mathcal{F}(I)$*;*
- *An objective function* $c : \mathcal{F} \to \mathbb{R}$ *we want to minimize;*

- *A neighbourhood* $N_I : \mathcal{F}(I) \to 2^{\mathcal{F}(I)}$

A solution $s \in \mathcal{F}(I)$ *is said to be a local optimum for* I *if:*

$$c(s) \leq c(s') \quad \forall s' \in N_I(s)$$

We can analogously define the problem for maximization with $u = -c$.

Definition 3.3 (PLS)**.** *A local search problem* Π *is said to be in PLS if* c *is computable in polynomial time, and also the following two functions are computable in polynomial time for any instance* $I \in \mathcal{I}_\Pi$*:*

- $start_I : \emptyset \to \mathcal{F}(I)$ *returns a feasible solution* $s \in \mathcal{F}(I)$*;*
- $next_I : \mathcal{F}(I) \to \mathcal{F}(I)$ *returns a feasible solution* $s' \in N_I(s)$ *with* $c(s') > c(s)$ *given a solution* s*, of* $s' = s$ *if* s *is a local optimum.*

Algorithm 3.4 (Local Search)**.** *Start with a feasible solution returned by* $s \leftarrow start_I()$*, then iteratively set* $s \leftarrow next_I(s)$ *until* $c(s)$ *stops decreasing.*

Theorem 3.5. *The Local Search algorithm always converges.*

Proof. At each step the algorithm strictly decreases the cost of the solution, and there are finitely many solutions. □

Note that this does not imply $P \subseteq PLS$ (unless $P = NP$), since the local search algorithm need not converge in polynomial time: there may be exponentially many steps.

3.3 PLS-hardness and max cut

Definition 3.6 (Max Cut)**.** *The max cut problem can be formalized as a local search:*

- *Instances are graphs* $G = (V, E, w)$ *with edge weights* $w : E \to \mathbb{N}$*;*
- *Feasible solutions are all the cuts of* G*;*
- *The neighbors of a cut* s *are the cuts obtained by moving exactly one of the vertices to the other side of the partition.*

Theorem 3.7. *Local search for unweighted maximum cut* ($w \equiv 1$) *converges in* $\mathcal{O}(|E|)$ *steps.*

Proof. The starting feasible solution has, in the worst case, 0 edges, while each step increases the size of the cut by at least 1, and a cut size cannot exceed $|E|$. □

Note: this does not mean the unweighted max cut is in P. We only find a local optimum.

Definition 3.8 (PLS-reduction)**.** *Given two local search problems* Π_1, Π_2*, we say that* Π_1 *PLS-reduces to* Π_2 ($\Pi_1 \leq_{PLS} \Pi_2$) *if the following two functions are computable in polynomial time:*

- $f : \mathcal{I}_{\Pi_1} \to \mathcal{I}_{\Pi_2}$ *maps every instance of* Π_1 *to an instance of* Π_2*;*
- $g : \mathcal{I}_{\Pi_2} \to \mathcal{I}_{\Pi_1}$ *maps every local optimum of* $f(I)$ *to a local optimum of* I*.*

Hardness and completeness for PLS are defined with respect to this reduction.

Theorem 3.9. *Max-Cut is PLS-complete.*

Clearly, finding a pure Nash equilibrium in congestion games is in PLS:

- *start* can be defined by taking any state $s \in S$ of the game;

- c is the Rosenthal potential function;
- The definition of local optimum is exactly the definition of pure Nash equilibrium.

Theorem 3.10. *Finding a pure Nash equilibrium is PLS-complete.*

Proof. We prove this by reducing the max cut problem to it. Given a graph $G = (V, E)$, each vertex is a player, and each edge e becomes a pair of resources r_e^L, r_e^R with delay functions:

$$d_{r_e}(k) = \begin{cases} 0 & k \leq 1 \\ w_e & k = 2 \end{cases}$$

Each player has only two strategies: either choosing all the L resources associated to its incident edges (i.e. the vertex goes on the left partition) or all the R resources (i.e. the vertex goes on the right partition). Notice that, with this, each resource is actually used by at most 2 players. We have two cases:

- If the edge e is in the current cut, then $n_{r_e}^L(s) = n_{r_e}^R(s)) = 1$ since the players of its endpoints chosen different resources, and the total contribution of e to the Rosenthal function is 0;
- If the edge e is not in the cut, then one of $n_{r_e}^L(s), n_{r_e}^R(s)$ is 2 (and the other is zero) implying that the total contribution of e to the Rosenthal function is w_e.

By rewriting the Rosenthal function, denoting with $C(s)$ the cut of the solution s:

$$\begin{aligned}
\Phi(s) &= \sum_r \sum_{k=1}^{n_r(s)} d_r(k) \\
&= \sum_e \left(\sum_{k=1}^{n_{r_e}^L(s)} d_{r_e}^L(k) + \sum_{k=1}^{n_{r_e}^R(s)} d_{r_e}^R(k) \right) \\
&= \sum_{e \in C(s)} \left(\sum_{k=1}^{n_{r_e}^L(s)} d_{r_e}^L(k) + \sum_{k=1}^{n_{r_e}^R(s)} d_{r_e}^R(k) \right) + \sum_{e \notin C(s)} \left(\sum_{k=1}^{n_{r_e}^L(s)} d_{r_e}^L(k) + \sum_{k=1}^{n_{r_e}^R(s)} d_{r_e}^R(k) \right) \\
&= \sum_{e \in C(s)} 0 + \sum_{e \notin C(s)} w_e \\
&= \sum_e w_e - \sum_{e \in C(s)} w_e
\end{aligned}$$

i.e. the Rosenthal function is locally minimum when the cut is locally maximum, which concludes the proof. □

4 Mixed and Correlated Equilibria

Definition 4.1 (Mixed Nash equilibrium). *Let $G = (n, S, u)$ be a game. Each player i chooses a probability distribution p_i over S_i. The product distribution $\sigma = p_1 \cdots p_n$ is a mixed Nash equilibrium if, for any player i and any other alternative strategy s_i':*

$$\mathbb{E}_{s\sim\sigma}\left[u_i(s)\right] \geq \mathbb{E}_{s\sim\sigma}\left[u_i(s_i', s_{-i})\right]$$

Definition 4.2 (Correlated equilibrium). *Let $G = (n, S, u)$ be a game. A probability distribution σ over S is said to be a correlated equilibrium if, for any player i and alternative strategy s_i':*

$$\mathbb{E}_{s\sim\sigma}\left[u_i(s) \mid s_i\right] \geq \mathbb{E}_{s\sim\sigma}\left[u_i(s_i', s_{-i}) \mid s_i\right]$$

Definition 4.3 (Coarse Correlated equilibrium). *Let $G = (n, S, u)$ be a game. A probability distribution σ over S is said to be a correlated equilibrium if, for any player i and alternative strategy s_i':*

$$\mathbb{E}_{s\sim\sigma}\left[u_i(s)\right] \geq \mathbb{E}_{s\sim\sigma}\left[u_i(s_i', s_{-i})\right]$$

These two definitions can be rewritten for cost games. Moreover, we define ε-mixed Nash equilibria, ε-correlated equilibria, and ε-coarse correlated equilibria, when the bounds are loosened by an additive factor ε.

Theorem 4.4. *For any game G, PNE(G) $\subseteq$ MNE(G) $\subseteq$ CE(G) $\subseteq$ CCE(G).*

Proof. We prove each inclusion separately:

- Any pure Nash equilibrium is trivially a mixed Nash equilibrium (it suffices to take the trivial distributions where the chosen strategy has probability 1).
- In a mixed Nash equilibrium σ the strategies are independent, thus:

$$\mathbb{E}_{s\sim\sigma}\left[u_i(s_i', s_{-i}) \mid s_i\right] = \mathbb{E}_{s\sim\sigma}\left[u_i(s_i', s_{-i})\right] \leq \mathbb{E}_{s\sim\sigma}\left[u_i(s)\right] = \mathbb{E}_{s\sim\sigma}\left[u_i(s) \mid s_i\right]$$

  i.e. σ is a correlated equilibrium.
- If σ is a correlated equilibrium, then:

$$\mathbb{E}_{s\sim\sigma}\left[u_i(s_i', s_{-i}) \mid s_i\right] \leq \mathbb{E}_{s\sim\sigma}\left[u_i(s) \mid s_i\right]$$

  by the law of total expectation:

$$\begin{aligned}\mathbb{E}_{s\sim\sigma}\left[u_i(s_i', s_{-i})\right] &= \sum_{s_i\in S_i} \mathbb{E}_{s\sim\sigma}\left[u_i(s_i', s_{-i}) \mid s_i\right] \mathbf{P}\left(s_i\right)\\ &\leq \sum_{s_i\in S_i} \mathbb{E}_{s\sim\sigma}\left[u_i(s) \mid s_i\right] \mathbf{P}\left(s_i\right)\\ &= \mathbb{E}_{s\sim\sigma}\left[u_i(s)\right]\end{aligned}$$

  i.e. σ is also a coarse correlated equilibrium.

□

We can extend the definitions of price of anarchy and (λ, μ)-smoothness immediately to these three new types of equilibria.

Corollary 4.5. *$PoA_{PNE} \leq PoA_{MNE} \leq PoA_{CE} \leq PoA_{CCE}$.*

Theorem 4.6 (Nash). *Every finite game admits a mixed Nash equilibrium.*

Theorem 4.7. *Bounds of Theorem 2.3 hold also for PoA_{CCE}.*

Proof. Take a coarse correlated equilibrium $s \sim \sigma$ and an optimal state s^*:

$$\begin{aligned}
\mathbb{E}_{s\sim\sigma}\left[SW(s)\right] &= \mathbb{E}_{s\sim\sigma}\left[\sum_i u_i(s)\right] \\
&= \sum_i \mathbb{E}_{s\sim\sigma}\left[u_i(s)\right] && \text{linearity of expectation} \\
&\geq \sum_i \mathbb{E}_{s\sim\sigma}\left[u_i(s_i^*, s_{-i})\right] && \text{definition of CCE} \\
&\geq \lambda \cdot SW(s^*) - \mu \cdot \mathbb{E}_{s\sim\sigma}\left[SW(s)\right] && \text{smoothness + LOE}
\end{aligned}$$

This implies $\frac{SW(s^*)}{\mathbb{E}[SW(s)]} \leq \frac{1+\mu}{\lambda}$ for any coarse correlated equilibria s. For cost games:

$$\begin{aligned}
\mathbb{E}_{s\sim\sigma}\left[cost(s)\right] &= \mathbb{E}_{s\sim\sigma}\left[\sum_i c_i(s)\right] \\
&= \sum_i \mathbb{E}_{s\sim\sigma}\left[c_i(s)\right] && \text{linearity of expectation} \\
&\leq \sum_i \mathbb{E}_{s\sim\sigma}\left[c_i(s_i^*, s_{-i})\right] && \text{definition of CCE} \\
&\leq \lambda \cdot cost(s^*) + \mu \cdot \mathbb{E}_{s\sim\sigma}\left[cost(s)\right] && \text{smoothness + LOE}
\end{aligned}$$

implying $\frac{\mathbb{E}[cost(s)]}{cost(s^*)} \leq \frac{\lambda}{1-\mu}$ for any coarse correlated equilibria s. □

Example 4.8. *The price of anarchy of different types of equilibria can be arbitrarily distant from each other. Consider a symmetric network congestion game with two players and two s-t edges with affine delay functions $d_r(n) = 1 - M + Mn$: a pure Nash equilibrium consists of the two players taking different edges and incurring a cost of 1 each, with a total cost of 2 (which is also optimal, i.e. $PoA_{PNE} = 1$). On the other hand, a mixed state where each player takes one of the two edges uniformly at random gives a mixed Nash equilibrium: if one player fixes the strategy, the expected cost will be $\frac{1+M}{2} > \frac{M}{2}$, and the expected total cost will be $1 + M$, by linearity of expectation.*

5 Regret Minimization

5.1 Online learning from experts

Definition 5.1 (Learning from experts problem)**.** *We have to answer a yes/no question in each round, and we have m experts which give us an answer. Using the hints given by the experts, answer to the question.*

Algorithm 5.2 (Hard majority)**.** *We follow the answer given by the majority. If we incur in a mistake, we discard every wrong expert. If we run out of experts we repeat the algorithm by readding back all of them.*

Claim 5.3. *If there is a perfect expert, we make $\mathcal{O}(\log m)$ mistakes.*

Proof. Each time we make a mistake we reduce the number of experts by at least one half. □

Corollary 5.4. *We make $\mathcal{O}(r^* \log m)$ mistakes, where r^* is the number of mistakes of the best expert.*

Algorithm 5.5 (Weighted majority)**.** *For each expert $a \in [m]$, set $w^1(a) = 1$. We choose yes or no based on the weighted majority of the answers of the experts. Then we set:*

$$w^{t+1}(a) \leftarrow \begin{cases} w^t(a) & \text{if } a \text{ was right} \\ \frac{1}{2} \cdot w^t(a) & \text{if } a \text{ was wrong} \end{cases}$$

Claim 5.6. *The number of mistakes incurred by the weighted majority algorithm is at most:*

$$C \leq 2.41(C^* + \log m)$$

Proof. Let $W^t = \sum_a w^t(a)$, and let T the time step we want to consider, i.e. C^* is the number of mistakes the best expert a^* makes in the first T steps. First of all we have that:

$$w^{T+1}(a) \geq w^{T+1}(a^*) = w^1(a^*) \cdot \left(\frac{1}{2}\right)^{C^*} = \left(\frac{1}{2}\right)^{C^*}$$

On the other hand, suppose that we make a mistake at time t. This means that:

$$W_+^t := \sum_{a \text{ right}} w^t(a) \leq \sum_{a \text{ wrong}} w^t(a) =: W_-^t$$

in particular we can write $W_+^t = \frac{W^t}{2} - \varepsilon, W_-^t = \frac{W^t}{2} + \varepsilon$ for some $\varepsilon \geq 0$. Since we halve the weights of the wrong experts, the new total weight will be:

$$\begin{aligned} W^{t+1} &= W_+^t + \frac{1}{2} \cdot W_-^t \\ &= \frac{W^t}{2} - \varepsilon + \frac{1}{2}\left(\frac{W^t}{2} + \varepsilon\right) \\ &= \frac{W^t}{2} + \frac{W^t}{4} - \varepsilon + \frac{\varepsilon}{2} \\ &\leq \frac{3}{4} \cdot W^t \end{aligned}$$

Otherwise, if we are right we use the trivial bound $W^{t+1} \leq W^t$. Therefore, at time $T+1$ we will have, by inductively applying the argument above:

$$W^{T+1} \leq W^1 \left(\frac{3}{4}\right)^C = m\left(\frac{3}{4}\right)^C$$

Putting all together we get:

$$\left(\frac{1}{2}\right)^{C^*} \leq w^{T+1}(a) \leq W^{T+1} \leq m\left(\frac{3}{4}\right)^C$$

The claim follows by taking the logarithms. □

5.2 External regret

Definition 5.7 (Regret minimization setting)**.** *We have a game between an algorithm $\mathcal{A}$ and an adversary, during indefinitely many steps. At step t:*

- *The player chooses an action $a \in A$, where $|A| = m$;*
- *The adversary assigns a cost $c^t(a) \in [0, 1]$ to each $a \in A$, and the player will incur the cost of the chosen action.*

The goal of the player is to minimize the incurred cost.

We run the algorithm up to step T, and let a^* be the **best fixed action in hindsight**, i.e. the action which gives the lowest cost C^*:

$$C^* := \min_a \sum_{t=1}^{T} c^t(a)$$

Definition 5.8. *If $C_{\mathcal{A}}$ is the total cost incurred by the algorithm, then the quantity:*

$$C_R := C_{\mathcal{A}} - C^*$$

*is called **external regret** of $\mathcal{A}$ (note that it is a function of T). If $C_R = o(T)$, i.e.*

$$C_{\mathcal{A}} \leq C^* + o(T)$$

then $\mathcal{A}$ is said to be a no-regret algorithm.

Theorem 5.9. *Any deterministic algorithm incurs $\Omega(T)$ external regret.*

Proof. Consider an adversary which always sets $c^t(a) = 1$ and $c^t(b) = 0$ for $b \neq a$ when a will be chosen by the algorithm at step t. This will incur a cost of T, and the best fixed action in hindsight will have a cost of at most T/m (since the sum of the costs among the steps is T, one of the m actions must have total cost $\leq T/m$). $\square$

5.3 Multiplicative Weights Update

Algorithm 5.10 (Multiplicative Weights Update)**.** *We set $w^1(a) = 1$ as in the weighted majority algorithm. At each step we set $w^{t+1}(a) = w^t(a)(1-\eta)^{c^t(a)}$, and we choose a with probability:*

$$p^t(a) = \frac{w^t(a)}{\sum_a w^t(a)}$$

Theorem 5.11. *For $\eta = \sqrt{\frac{\ln m}{T}} \leq \frac{1}{2}$, the multiplicative weights update is a no-regret algorithm.*

Proof. Again, let $W^t = \sum_a w^t(a)$. If a^* is the best fixed action in hindsight then:

$$W^{T+1} \geq w^{T+1}(a^*) = w^1(a^*)(1-\eta)^{\sum_{t=1}^{T} c^t(a^*)} = (1-\eta)^{C^*}$$

On the other hand, we can use Bernoulli's inequality:

$$\begin{aligned}
W^{t+1} &= \sum_a w^{t+1}(a) \\
&= \sum_a w^t(a)(1-\eta)^{c^t(a)} \\
&\leq \sum_a w^t(a)(1-\eta \cdot c^t(a)) \\
&= \sum_a w^t(a) - \eta \sum_a w^t(a)c^t(a) \\
&= W^t - \eta W^t \sum_a \frac{w^t(a)}{W^t} c^t(a) \\
&= W^t - \eta W^t \sum_a p^t(a)c^t(a) \\
&= W^t(1-\eta C^t)
\end{aligned}$$

where C^t is the expected cost incurred by the algorithm at step t. By inductively applying the inequality we get:

$$(1-\eta)^{C^*} \leq W^{T+1} \leq W^1 \prod_{t=1}^{T}(1-\eta \cdot C^t) = m \prod_{t=1}^{T}(1-\eta \cdot C^t)$$

and, if we take the logarithms, we obtain:

$$C^* \log(1-\eta) \leq \log m + \sum_{t=1}^{T} \log(1-\eta \cdot C^t)$$

Since $\log(1-\eta) \geq \eta - \eta^2$ and $\log(1-\eta C^t) \leq -\eta C^t$:

$$-C^*(1+\eta) \leq \log m - \sum_{t=1}^{T} C^t = \frac{\log m}{\eta} - C$$

which gives:

$$C \leq (1+\eta)C^* + \frac{\log m}{\eta} \leq C^* + \eta T + \frac{\log m}{\eta} = C^* + 2\sqrt{T \log m} = C^* + o(T)$$

$\square$

5.4 Computing a coarse correlated equilibrium

Definition 5.12 (No-regret dynamics)**.** *A no-regret dynamics consists of a game $G = (n, S, c)$ where, at each step, players choose a strategy using a no-regret algorithm $\mathcal{A}$. Each player i receives a cost $c^t(s_i)$:*

$$c^t(s_i) := \mathbb{E}_{s_{-i} \sim \sigma_{-i}} \left[c^t(s_i, s_{-i}) \right]$$

where σ_{-i} is the marginal mixed strategy of the other players.

Theorem 5.13. *Let $\sigma^1, \ldots, \sigma^T$ be mixed states generated by a no-regret dynamics with external regret C_R. A mixed state $\sigma = \sigma^\tau$, where $\tau \sim \mathcal{U}[T]$ is a $\frac{C_R}{T}$-coarse correlated equilibrium.*

Proof. We use the law of total expectation on τ:

$$\mathbb{E}_{s\sim\sigma}\left[c_i(s)\right] - \mathbb{E}_{s\sim\sigma}\left[c_i(s_i', s)\right] = \frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}_{s\sim\sigma^t}\left[c_i(s)\right] - \mathbb{E}_{s\sim\sigma^t}\left[c_i(s_i', s)\right]\right)$$

Fixed a player i, it is immediate to see that:

$$\sum_{t=1}^{T}\mathbb{E}_{s\sim\sigma^t}\left[c_i(s)\right] = C$$

which is the expected cost of the algorithm used by player i (the costs are indeed from the adversary of i, i.e. the other players). On the other hand, if s_i^* is the best fixed action in hindsight, we have:

$$\begin{aligned}
\sum_{t=1}^{T}\mathbb{E}_{s\sim\sigma^t}\left[c_i(s_i', s_{-i})\right] &= \sum_{t=1}^{T} c^t(s_i') && \text{definition of no-regret dynamics}\\
&\geq \sum_{t=1}^{T} c^t(s_i^*)\\
&= C^*
\end{aligned}$$

Hence, we found that:

$$\mathbb{E}_{s\sim\sigma}\left[c_i(s)\right] \leq \mathbb{E}_{s\sim\sigma}\left[c_i(s_i', s)\right] + \frac{C - C^*}{T} = \mathbb{E}_{s\sim\sigma}\left[c_i(s_i', s)\right] + \frac{C_R}{T}$$

which is exactly the definition of $\frac{C_R}{T}$-coarse correlated equilibrium. □

Theorem 5.14. *A ε-coarse correlated equilibrium can be computed with $\mathcal{O}\left(\frac{\log m}{\varepsilon^2}\right)$ iterations of the no-regret dynamics.*

Proof. The multiplicative weight update we yields a no-regret dynamics with external regret $2\sqrt{T\log m}$, and the coarse correlated equilibrium will have a slack of:

$$2\sqrt{\frac{\log m}{T}}$$

If we let the dynamics evolve for $T = 4\frac{\log m}{\varepsilon^2}$ steps, the slack above will be exactly ε. □

6 Price of Stability

Definition 6.1. *Given a game $G = (n, S, u)$, the price of stability is defined as:*

$$PoS_{PNE} = \frac{\max_{s \in S} SW(s)}{\max_{s \in PNE(G)} SW(s)}$$

For cost games $G = (n, S, c)$:

$$PoS_{PNE} = \frac{\min_{s \in PNE(G)} cost(s)}{\min_{s \in S} cost(s)}$$

The definition can be extended to mixed Nash, correlated, and coarse correlated equilibria like we did for the price of anarchy.

Theorem 6.2. *The price of stability for symmetric fair cost sharing congestion games, i.e. with delay functions:*

$$d_r(n) = \frac{c_r}{n}$$

is 1.

Proof. In order to prove this, we show that any state s minimizing the social cost must be a pure Nash equilibrium. In fact, take such state. The total cost of a state is given by:

$$cost(s) = \sum_i c_i(s) = \sum_i \sum_{r \in s_i} \frac{c_r}{n_r(s)} = \sum_r \sum_{i : r \in s_i} \frac{c_r}{n_r(s)} = \sum_{r : n_r(s) \geq 1} c_r$$

Notice that, in state s, every player will take the exact same s-t path P: in fact, if this was not true, we would have to construct more resources for s and, by switching to a path P chosen by one of the players in s, we would use a proper subset of the resources allocated for s, i.e. we find a state s' with $cost(s') < cost(s)$, contradicting the minimality of s.

Now it is easy to see that s is a pure Nash equilibrium: if a player i could choose a path $P' \neq P$ strictly minimizing its cost, this would mean in particular that P' has a total cost strictly smaller than P, then by letting all the players switch to P' we would obtain a state s' with strictly smaller total cost than s, again contradicting the minimality of s. $\square$

Theorem 6.3. *$PoS_{PNE} \leq H_n$ for fair cost sharing congestion games.*

Proof. In order to bound the price of stability, we can use the Rosenthal function. Notice that:

$$\Phi(s) = \sum_r \sum_{k=1}^{n_r(s)} \frac{c_r}{k} \geq \sum_{r : n_r(s) \geq 1} c_r = cost(s)$$

On the other hand, we can easily find an upper bound:

$$\begin{aligned}
\Phi(s) &= \sum_r \sum_{k=1}^{n_r(s)} \frac{c_r}{k} \\
&= \sum_{r : n_r(s) \geq 1} c_r \sum_{k=1}^{n_r(s)} \frac{1}{k} \\
&\leq \sum_{r : n_r(s) \geq 1} c_r \cdot H_n && \text{since } n_r \leq n \\
&= cost(s) \cdot H_n
\end{aligned}$$

This is enough to provide the bound: if s is a minimizer of Φ, and s^* a social optimum, we obtain that:

$$cost(s) \leq \Phi(s) \leq \Phi(s^*) = cost(s^*) \cdot H_n \Longrightarrow \frac{cost(s)}{cost(s^*)} \leq H_n$$

and by Theorem 1.10, s is also a pure Nash equilibrium, implying that the bound on the right is also an upper bound on the price of stability for pure Nash equilibria. □

The bound is also tight (see Example 4 of lecture notes).

7 Mechanisms with Money

7.1 Setting (cost version)

Definition 7.1 (Setting, cost version)**.** *Consider a game $G = (n, S, u)$ where:*

- *We have a set $\mathcal{A}$ of feasible solutions and we have to pick one;*
- *Each player i has a true private cost $t_i : \mathcal{A} \to \mathbb{R}$ where $t_i(a)$ is the cost incurred by player i when solution a is chosen;*
- *A strategy $s_i \in S_i$ is a cost $c_i : \mathcal{A} \to \mathbb{R}$ to report (which may or may not be equal to t_i);*
- *In the end each player receive a payment p_i, and the utility of player i will be:*

$$u_i = p_i - t_i(a)$$

Definition 7.2 (Mechanism with money, cost version)**.** *Let $G = (n, S, u)$ be a game as defined in the setting above. A mechanism with money is a pair (A, P) of computable functions where:*

- *$A : S \to \mathcal{A}$ constructs a solution given the reported costs $c \in S$;*
- *$P : S \to \mathbb{R}^n$ is the payment function, which assigns the payments to the players.*

We denote with P_i the payment to player i. The utility u for each player can be rewritten as:

$$u_i(c|t_i) = P_i(c) - t_i(A(c))$$

where $u_i(\cdot|t_i)$ means that the utility is expressed under the assumption that t_i is player i's true cost function.

7.2 Setting (utility version)

Definition 7.3 (Setting, utility version)**.** *Consider a game $G = (n, S, u)$ where:*

- *We have a set $\mathcal{A}$ of feasible solutions and we have to pick one;*
- *Each player i has a true private valuation $v_i : \mathcal{A} \to \mathbb{R}$ where $v_i(a)$ is the utility gained by player i when solution a is chosen;*
- *A strategy $s_i \in S_i$ is a valuation $b_i : \mathcal{A} \to \mathbb{R}$ to report (which may or may not be equal to v_i);*
- *In the end each player will be charged a payment of p_i, and the utility of player i will be:*

$$u_i = v_i(a) - p_i$$

Definition 7.4 (Mechanism with money, utility version)**.** *Let $G = (n, S, u)$ be a game as defined in the setting above. A mechanism with money is a pair (A, P) of computable functions where:*

- *$A : S \to \mathcal{A}$ constructs a solution given the reported valuations $b \in S$;*
- *$P : S \to \mathbb{R}^n$ is the payment function, which assigns the payments from the players.*

We denote with P_i the payment from player i. The utility u for each player can be rewritten as:

$$u_i(c|v_i) = v_i(A(c)) - P_i(c)$$

where $u_i(\cdot|v_i)$ means that the utility is expressed under the assumption that v_i is player i's true valuation function.

7.3 Truthfulness and VCG mechanism

Definition 7.5 (Truthful mechanism). *A mechanism with money (A, P) is said to be truthful if, for any state $c \in S$ and $i \in [n]$:*

$$u_i(t_i, c_{-i}|t_i) \geq u_i(c|t_i)$$

i.e. the utility of player i is maximized when player i reports the true cost/valuation t_i.

Definition 7.6 (VCG mechanism, cost version). *A mechanism with money (A, P) is a VCG mechanism if:*

- *The algorithm A minimizes the social cost, i.e.*

  $$cost_c(a) = \sum_i c_i(a)$$

  is minimized when $a = A(c)$.

- *The payment function P is of the form:*

  $$P_i(c) = Q_i(c_{-i}) - \sum_{j \neq i} c_j(A(c))$$

  where Q_i is an arbitrary function independent of c_i.

Definition 7.7 (VCG mechanism, utility version). *A mechanism with money (A, P) is a VCG mechanism if:*

- *The algorithm A maximizes the social welfare, i.e.*

  $$SW_b(a) = \sum_i b_i(a)$$

  is maximized when $a = A(c)$.

- *The payment function P is of the form:*

  $$P_i(b) = Q_i(b_{-i}) - \sum_{j \neq i} b_j(A(b))$$

  where Q_i is an arbitrary function independent of b_i.

Theorem 7.8. *The VCG mechanism is truthful.*

Proof. We prove this for the cost version (for the utility version is sufficient to plug $c_i = -v_i$ and invert the utility function). Fix a player i and the costs c_{-i} reported by other players. Let

$$\bar{t} = (t_i, c_{-i})$$

be the state where player i is truth-telling. We rewrite the utility for player i:

$$\begin{aligned} u_i(c|t_i) &= P_i(c) - t_i(A(c)) \\ &= Q_i(c_{-i}) - \sum_{j \neq i} c_j(A(c)) - t_i(A(c)) \\ &= Q_i(c_{-i}) - cost_{\bar{t}}(A(c)) \end{aligned}$$

The utility is maximized when $cost_{\bar{t}}(a)$ is minimized (Q_i does not depend on player i by definition). If $c = \bar{t}$, then $cost_{\bar{t}}(A(c))$ reaches a minimum by optimality of A. $\square$

7.4 One-parameter mechanisms

Definition 7.9 (One-parameter setting). *The one-parameter setting is a setting with money where:*

- *Each player has a private type $t_i \in \mathbb{R}$;*
- *Chosen a solution $a \in \mathcal{A}$ player i incurs a cost of:*

$$w_i(a) \cdot t_i$$

  where w_i is public.

Definition 7.10 (Monotonicity). *We say that A is monotone if, for every i and c_{-i}*

$$w_i(A(c_i, c_{-i}))$$

is monotone non-increasing.

Lemma 7.11 (Myerson). *In the one-parameter setting, there exists a payment function P such that (A, P) is truthful if and only if A is monotone.*

Proof. If A is not monotone, then for some player i and some c_{-i} we have $c' < c''$ such that:

$$w_i(A(c', c_{-i})) < w_i(A(c'', c_{-i}))$$

Since P cannot depend on the private types t_i, it must handle both the following cases:

- **Case 1**: $t_i = c'$:

$$\begin{aligned} u_i(c'', c_{-i}|c') &\leq u_i(c', c_{-i}|c') \\ P_i(c'', c_{-i}) - w_i(A(c'', c_{-i})) \cdot c' &\leq P_i(c', c_{-i}) - w_i(A(c', c_{-i})) \cdot c' \\ P_i(c'', c_{-i}) &\leq P_i(c', c_{-i}) - \big(w_i(A(c', c_{-i})) - w_i(A(c'', c_{-i}))\big) \cdot c' \end{aligned}$$

- **Case 2**: $t_i = c''$:

$$\begin{aligned} u_i(c', c_{-i}|c'') &\leq u_i(c'', c_{-i}|c'') \\ P_i(c', c_{-i}) - w_i(A(c', c_{-i})) \cdot c'' &\leq P_i(c'', c_{-i}) - w_i(A(c'', c_{-i})) \cdot c'' \\ P_i(c', c_{-i}) &\leq P_i(c'', c_{-i}) + \big(w_i(A(c', c_{-i})) - w_i(A(c'', c_{-i}))\big) \cdot c'' \end{aligned}$$

We infer that:

$$\begin{aligned} P_i(c', c_{-i}) &\leq P_i(c'', c_{-i}) + \big(w_i(A(c', c_{-i})) - w_i(A(c'', c_{-i}))\big) \cdot c'' && \text{Case 2} \\ &< P_i(c'', c_{-i}) + \big(w_i(A(c', c_{-i})) - w_i(A(c'', c_{-i}))\big) \cdot c' && \text{since } c' < c'' \\ &\leq P_i(c', c_{-i}) && \text{Case 1} \end{aligned}$$

i.e. a contradictory statement must hold for P, implying that no P can make the mechanism truthful.

Now suppose that A is monotone, and consider the following payment function.

$$P_i(c_i, c_{-i}) = c_i \cdot w_i(A(c_i, c_{-i})) + \int_{c_i}^{\infty} w_i(A(u, c_{-i})du$$

and the utility of player i will then be:

$$u_i(c_i, c_{-i}|t_i) = c_i \cdot w_i(A(c_i, c_{-i})) + \int_{c_i}^{\infty} w_i(A(u, c_{-i}))du - t_i \cdot w_i(A(c_i, c_{-i}))$$

i.e. the utility of a truth-telling player i will be:

$$u_i(t_i, c_{-i}|t_i) = \int_{t_i}^{\infty} w_i(A(u, c_{-i})du$$

- Assume $c_i < t_i$. The integral can be decomposed:

$$\begin{aligned}\int_{c_i}^{\infty} w_i(A(u,c_{-i}))du &= \int_{c_i}^{t_i} w_i(A(u,c_{-i}))du + \int_{t_i}^{\infty} w_i(A(u,c_{-i}))du \\ &\leq \int_{c_i}^{t_i} w_i(A(c_i,c_{-i}))du + \int_{t_i}^{\infty} w_i(A(u,c_{-i}))du \qquad \text{by monotonicity} \\ &= (t_i - c_i)w_i(A(c_i,c_{-i})) + \int_{t_i}^{\infty} w_i(A(u,c_{-i}))du\end{aligned}$$

  which implies that the utility of player i is:

$$\begin{aligned}u_i(c_i,c_{-i}|t_i) &= c_i \cdot w_i(A(c_i,c_{-i})) + \int_{c_i}^{\infty} w_i(A(u,c_{-i}))du - t_i \cdot w_i(A(c_i,c_{-i})) \\ &\leq \int_{t_i}^{\infty} w_i(A(u,c_{-i}))du \\ &= u_i(t_i,c_{-i}|t_i)\end{aligned}$$

- Now assume $c_i > t_i$. The integral can be decomposed:

$$\begin{aligned}\int_{c_i}^{\infty} w_i(A(u,c_{-i}))du &= \int_{t_i}^{\infty} w_i(A(u,c_{-i}))du - \int_{t_i}^{c_i} w_i(A(u,c_{-i}))du \\ &\leq \int_{t_i}^{\infty} w_i(A(u,c_{-i}))du - \int_{t_i}^{c_i} w_i(A(c_i,c_{-i}))du \qquad \text{by monotonicity} \\ &= \int_{t_i}^{\infty} w_i(A(u,c_{-i}))du - (c_i - t_i)w_i(A(c_i,c_{-i}))\end{aligned}$$

  which implies that the utility of player i is:

$$\begin{aligned}u_i(c_i,c_{-i}|t_i) &= c_i \cdot w_i(A(c_i,c_{-i})) + \int_{c_i}^{\infty} w_i(A(u,c_{-i}))du - t_i \cdot w_i(A(c_i,c_{-i})) \\ &\leq \int_{t_i}^{\infty} w_i(A(u,c_{-i}))du \\ &= u_i(t_i,c_{-i}|t_i)\end{aligned}$$

This proves that the utility of a truth-telling player is maximized. The integrals above may not converge ($w_i(A(c_i, c_{-i}))$ may not even tend to 0, imagine a shortest path problem with a cut edge). In such cases, we can use the following payments:

$$P_i(c_i,c_{-i}) = Q_i(c_{-i}) + c_i \cdot w_i(A(c_i,c_{-i})) - \int_{0}^{c_i} w_i(A(u,c_{-i}))du$$

and a similar argument can be applied. □

Definition 7.12 (Voluntary participation)**.** *A mechanism with money is said to achieve voluntary participation if the utility of every truth-telling player is always non-negative.*

Corollary 7.13. *If the integral*

$$\int_{C}^{\infty} w_i(A(u,c_{-i}))du$$

converges for every i with a sufficiently large $C > 0$, a mechanism with money (A,P) with monotone A and P defined as:

$$P_i(c_i,c_{-i}) = c_i \cdot w_i(A(c_i,c_{-i})) + \int_{c_i}^{\infty} w_i(A(u,c_{-i})du$$

is truthful with voluntary participation.

Proof. It is sufficient to see that $u_i(t_i, c_{-i}|t_i)$ is non-negative, since it is the integral of a non-negative function. □

7.5 Selfish related machines

We have n machines (players) with private type t_i which is the time machine i takes to execute one unit of work. We also have k jobs with amounts of work $w_1, \ldots, w_k$, and $w_i(a)$ given by an allocation a is the total amount of work assigned to player i, which in turn will incur a total cost of $t_i \cdot w_i(a)$. The goal is to minimize the makespan:

$$mksp_t(a) := \max_i t_i \cdot w_i(a)$$

Algorithm 7.14 (Archer-Tardos). *Define a total order $\leq$ of allocations, and return an allocation of minimum makespan (enumerate them, they are finite). If two or more allocations minimize the makespan, then return the first of these allocations with respect to $\leq$.*

The algorithm is optimal by definition, although it does not run in polynomial time.

Theorem 7.15. *Algorithm 7.14 is monotone.*

Proof. Fix a player i and a state $c_{-i} \in S_{-i}$ of reported costs from the other players. Let $c_i < c'_i$ be two choices for player i and let $c = (c_i, c_{-i}), c' = (c'_i, c_{-i})$. Moreover, denote $a = A(c), a' = A(c')$. We must have $mksp_{c'}(a) \geq mksp_c(a)$ (the only changing weight is machine i's, which is increasing).

- If $mksp_{c'}(a) = mksp_c(a)$, then this makespan is still optimal: in fact, suppose for a contradiction we have $mksp_{c'}(a^*) < mksp_{c'}(a)$ for some a^*. Since going back from c' to c cannot increase the makespan of any allocation, we must have

$$mksp_c(a^*) \leq mksp_{c'}(a^*) < mksp_{c'}(a) = mksp_c(a)$$

  contradicting the optimality of a. Hence, $a' = a$ by the total order of the algorithm (no allocation before a in the order can become optimal by increasing c_i, as this would require a decrease in the makespan).

- If $mksp_{c'}(a) > mksp_c(a)$, this means that machine i is the heaviest in a, and any allocation assigning strictly more workload cannot minimize the makespan (as the makespan of a would be strictly smaller).

Thus we conclude that $w_i(a') \leq w_i(a)$ in any case. □

7.6 Selfish unrelated machines

Consider a natural extension of the problem above: each player i has a different true cost t_i^j for job j, and reports c_i^j.

An allocation $a \in \mathcal{A}$ is decomposed in values:

$$a_i^j = \begin{cases} 1 & \text{job } j \text{ is assigned to machine } i \\ 0 & \text{otherwise} \end{cases}$$

Hence, the makespan for allocation a with respect to costs c is:

$$mksp_c(a) = \max_i \sum_j a_i^j \cdot c_i^j = \max_i a_i^T c_i$$

Theorem 7.16. *There is no exact truthful mechanism for the selfish unrelated machines setting.*

Proof. Let A be an optimal algorithm and consider an instance with four jobs and two machines. Let $c_1 = c_2 = (1,1,1,1)$ and suppose without loss of generality that the first two jobs are allocated to machine 1 and the others to machine 2 in $A(c_1, c_2)$ (the optimal allocation must have this form). On the other hand, for reported costs $c'_1 = (\varepsilon, \varepsilon, 1+\varepsilon, 1+\varepsilon)$ for some $\varepsilon > 0$, $A(c'_1, c_2)$ must allocate the two ε jobs and one $1+\varepsilon$ job in the first machine, while the second machine gets the remaining job. The total makespan would be $1+3\varepsilon$ (giving more than one job to machine 2 yields a makespan of ≥ 2).

Truthfulness would require us to handle the following two cases:

- **Case 1**: $t_1 = c_1$

$$\begin{aligned} u_1(c_1, c_2|c_1) &\geq u_1(c'_1, c_2|c_1) \\ P_1(c_1, c_2) - A_1(c_1, c_2)^T c_1 &\geq P_1(c'_1, c_2) - A_1(c'_1, c_2)^T c_1 \\ P_1(c_1, c_2) - 2 &\geq P_1(c'_1, c_2) - 3 \end{aligned}$$

- **Case 2**: $t_1 = c'_1$

$$\begin{aligned} u_1(c'_1, c_2|c'_1) &\geq u_1(c_1, c_2|c'_1) \\ P_1(c'_1, c_2) - A_1(c'_1, c_2)^T c'_1 &\geq P_1(c_1, c_2) - A_1(c_1, c_2)^T c'_1 \\ P_1(c'_1, c_2) - (1+3\varepsilon) &\geq P_1(c_1, c_2) - 2\varepsilon \end{aligned}$$

Both cases give a constraint on P:

$$\begin{cases} P_1(c_1, c_2) \geq P_1(c'_1, c_2) - 1 \\ -P_1(c_1, c_2) \geq -P_1(c'_1, c_2) + (1+\varepsilon) \end{cases} \implies 0 \geq \varepsilon$$

which is contradictory. Thus, no P can make the mechanism truthful for this instance. □

Theorem 7.17. *The VCG construction yields a n-approximation truthful mechanism.*

Proof. The VCG construction is truthful, by Theorem 7.8. We also know that the VCG algorithm minimizes the sum of the costs, let a be an allocation returned by the algorithm and let a^* be a makespan minimizer.

$$\begin{aligned} mksp_c(a) &= \max_i c_i^T a_i \\ &\leq \sum_i c_i^T a_i \\ &\leq \sum_i c_i^T a_i^* && \text{by optimality of } a \\ &\leq \sum_i \max_j c_j^T a_j^* = n \cdot mksp_c(a^*) \end{aligned}$$

□

Corollary 7.18. *No α-approximation mechanism is truthful for $\alpha < 2$.*

Proof. We use the same idea as above, but we consider 2ℓ jobs. Suppose without loss of generality that, when $c = (1, \dots, 1)$ is reported, the algorithm allocates $x \geq \ell$ jobs to machine 2 (one of the two machines must have this number of jobs).

Construct c' from c where machine 1 declares weight ε for every job it obtained in $A(c)$, and weight $1+\varepsilon$ for the others.

The optimal solution is obtained by the following condition (assuming ε is sufficiently small):

$$\begin{gathered} a_1^T c_1' \overset{!}{=} a_2^T c_2' \\ (2\ell - x)\varepsilon + w_1(1+\varepsilon) = g - w_1 \end{gathered}$$

where w_1 is the number of jobs with weight $1+\varepsilon$ assigned to machine 1 by a. This gives us:

$$(2\ell - x)\varepsilon - g = -(2+\varepsilon)w_1$$

letting $\varepsilon \to 0$ we get $w_1 = g/2$ which is also the makespan since ε-jobs have 0 weight.

Again, we set up the two cases:

- **Case 1**: $t_1 = c_1$

$$\begin{gathered} u_1(c_1, c_2|c_1) \geq u_1(c_1', c_2|c_1) \\ P_1(c_1, c_2) - A_1(c)^T c_1 \geq P_1(c_1', c_2) - A_1(c')^T c_1 \end{gathered}$$

- **Case 2**: $t_1 = c_1'$

$$\begin{gathered} u_1(c_1', c_2|c_1') \geq u_1(c_1, c_2|c_1') \\ P_1(c_1', c_2) - A_1(c')^T c_1' \geq P_1(c_1, c_2) - A_1(c)^T c_1' \end{gathered}$$

By adding up the two inequalities we obtain the monotonicity condition:

$$\begin{gathered} A_1(c)^T c_1 + A_1(c')^T c_1' \leq A_1(c')^T c_1 + A_1(c)^T c_1' \\ (2\ell - x) + \varepsilon w_\varepsilon + (1+\varepsilon)w_{1+\varepsilon} \leq (w_\varepsilon + w_{1+\varepsilon}) + (2\ell - x)\varepsilon \\ (1-\varepsilon)w_\varepsilon + \varepsilon w_{1+\varepsilon} \geq (2\ell - x)(1-\varepsilon) \end{gathered}$$

which tells us, for $\varepsilon \to 0$, that all the ε-jobs will finish in the first machine. If $A(c')$ assigned other jobs to machine 1, then machine 1 would obtain more workload by reporting c' instead of c. □

7.7 Combinatorial auctions

Definition 7.19 (Combinatorial auction). *A combinatorial auction consists of a set M of m items and n bidders:*

- $\mathcal{A} = \{(S_1, \ldots, S_n) \in (2^M)^n \mid S_i \cap S_j = \emptyset, \forall i \neq j\}$ *is the set of feasible allocations;*
- *Each player has a true valuation $v_i : 2^M \to \mathbb{R}$ such that $v_i(S) \leq v_i(T)$ for $S \subseteq T$ and $v_i(\emptyset) = 0$;*
- *We want to maximize the social welfare*

$$SW(S_1, \ldots, S_n) = \sum_i v_i(S_i)$$

Definition 7.20. *A combinatorial auction is said to be single minded if each player has a unique (public) bundle S_i^* such that:*

$$v_i(S) = \begin{cases} v_i^* & \text{if } S_i^* \subseteq S_i \\ 0 & \text{otherwise} \end{cases}$$

The only private value is v_i^*.

Notice that single minded combinatorial auctions are one-parameter problems ($w_i(a) \in \{0,1\}$, and instead a private type t_i we have a private valuation v_i^*). A VCG mechanism yields an exact truthful mechanism for combinatorial auctions, but a polynomial-time optimal algorithm is required.

Theorem 7.21. *Computing a $o(\sqrt{m})$-approximation to the optimal allocation for combinatorial auctions is NP-hard.*

Proof. We prove the claim with a linear reduction from independent set. Considering an input graph $G = (V, E)$ with $n = |V|$, we set a single minded combinatorial auction with $v_i^* = 1$ for every i and $S_i^* = \{e \in E \mid i \in e\}$. An allocation of winning bidders is an independent set of G by construction, and the social welfare is exactly the size of such set. □

Theorem 7.22 (Hazan et al. 2006)**.** *Let $d = \max_i |S_i^*|$. Computing a $o\left(\frac{d}{\log d}\right)$-approximation of the best allocation for combinatorial auctions is NP-hard.*

7.8 Greedy approximation: greedy-by-value

Algorithm 7.23 (Greedy-by-value)**.** *Let $W = \emptyset$ be the set of winning bidders, and consider every bidder in decreasing order of bids $b_1 \geq \cdots \geq b_n$. For each bidder i, we add i to W if:*

$$S_i^* \cap \bigcup_{j \in W} S_j^* = \emptyset$$

Theorem 7.24. *Algorithm 7.23 is monotone.*

Proof. Fix a bidder i and the bids b_{-i} of other bidders. If bidder i wins in (b_i, b_{-i}), they will also win by bidding any $b_i' > b_i$: an higher bid cannot make bidder i go lower in the order of the algorithm, so the $W' \subseteq W$, where W', W are the set of winners bidder i will face when considered by the algorithm when bidding, respectively, b_i', b_i, i.e.

$$S_i^* \cap \bigcup_{j \in W} S_j^* = \emptyset \Longrightarrow S_i^* \cap \bigcup_{j \in W'} S_j^* = \emptyset$$

This is sufficient to prove that the algorithm is monotone. □

Folklore Theorem 7.25. *Algorithm 7.23 yields a d-approximation.*

Proof. We prove the approximation by a charging argument. Every $i \in W$ can block at most d bidders in OPT (by definition of d), let OPT_i be the set of bidders chosen by OPT blocked by $i \in W$. If $j \in OPT_i$, this means that there must have been an higher bidder $b_i \geq b_j$ with $i \in W$, i.e.

$$d \cdot b_i \geq \sum_{j \in OPT_i} b_j$$

By summing up for every $i \in W$ we obtain:

$$d \cdot \sum_{i \in W} b_i \geq \sum_{i \in W} \sum_{j \in OPT_i} b_j \geq \sum_{j \in OPT} b_j$$

where the last inequality is a union bound. □

Theorem 7.26. *Algorithm 7.23 cannot guarantee better than d-approximation.*

Proof. Consider an instance with $n = m + 1$ players and $m = d$ items. m players each claim a separate unique item with value 1, and the last player claims a bundle containing all the d items with value $1 + \varepsilon$, for arbitrarily small $\varepsilon > 0$. The algorithm will pick the last player and discard the others, yielding a social welfare of $1 + \varepsilon$, while the optimal solution is obtained by taking the first d players, with a total social welfare of d. □

7.9 Greedy approximation: greedy-by-value/density

Algorithm 7.27 (Greedy-by-value/density). *Let $W = \emptyset$ be the set of winning bidders, and consider every bidder in decreasing order of bids-to-size ratio.*

$$\frac{b_1}{\sqrt{|S_1^*|}} \geq \cdots \geq \frac{b_n}{\sqrt{|S_n^*|}}$$

For each bidder i, we add i to W if:

$$S_i^* \cap \bigcup_{j \in W} S_j^* = \emptyset$$

Theorem 7.28. *Algorithm 7.27 is monotone.*

Proof. The exact same argument for the greedy-by-value algorithm holds, since the sizes of the bundles cannot change. □

Theorem 7.29. *Algorithm 7.27 yields a $\sqrt{m}$-approximation.*

Proof. We use again a charging argument. A bidder $i \in W$ blocks a subset $j \in OPT_i$ of an optimum OPT because:

$$\frac{b_j}{\sqrt{|S_j^*|}} \leq \frac{b_i}{\sqrt{|S_i^*|}} \iff b_j \leq \sqrt{|S_j^*|} \cdot \frac{b_i}{\sqrt{|S_i^*|}}$$

and, by summing up over OPT_i we get:

$$\begin{aligned}
\sum_{j \in OPT_i} b_j &\leq \frac{b_i}{\sqrt{|S_i^*|}} \sum_{j \in OPT_i} \sqrt{|S_j^*|} \\
&\leq \frac{b_i}{\sqrt{|S_i^*|}} \sqrt{|OPT_i|} \sqrt{\sum_{j \in OPT_i} |S_j^*|} && \text{Cauchy-Schwartz inequality} \\
&\leq \frac{b_i}{\sqrt{|S_i^*|}} \sqrt{|OPT_i|} \sqrt{m} && \text{since } OPT_i \subseteq OPT \text{ is feasible}
\end{aligned}$$

The analysis ends by noticing that $|OPT_i| \leq |S_i^*|$: each item in S_i^* is claimed by at most one bidder in OPT_i (again, because OPT_i is a feasible solution). Hence, we are left with:

$$\sum_{j \in OPT_i} b_j \leq b_i \sqrt{m}$$

By summing up over every $i \in W$ and applying a union bound:

$$\sum_{j \in OPT} b_j \leq \sum_{i \in W} \sum_{j \in OPT_i} b_j \leq \sqrt{m} \cdot \sum_{i \in W} b_i$$

which concludes the proof. □

8 Voting Systems

Definition 8.1 (Voting setting)**.** *Let A be a set of alternatives, and consider n voters. Each voter has a (private) true preference order $\leq_i^* \in \mathcal{L}(A)$, where $\mathcal{L}(A)$ denotes the set of total orders over A.*

Definition 8.2. *A social choice function is a function $f : \mathcal{L}(A)^n \to A$.*

Definition 8.3. *A social welfare function is a function $F : \mathcal{L}(A)^n \to \mathcal{L}(A)$.*

Definition 8.4 (Strategic manipulation)**.** *Let $\leq_1, \dots, \leq_n \in \mathcal{L}(A)$ be a set of preferences for the voters, and denote $P = (\leq_1, \dots, \leq_n)$. A social choice function f is said to admit a strategic manipulation if, for some i we have:*

- *$a' \leq_i a$ i.e. a' is preferred over a by voter i;*
- *$f(P) = a$ nonetheless;*
- *There exists an alternative preference $\leq_i' \in \mathcal{L}(A)$ such that:*

$$f(\leq_1, \dots, \leq_{i-1}, \leq_i', \leq_{i+1}, \dots \leq_n) = a'$$

Definition 8.5 (Incentive compatibility)**.** *A social choice function f is said to be incentive compatible if it does not admit a strategic manipulation.*

8.1 Examples of voting systems

Theorem 8.6. *If $|A| = 2$, then the social choice function picking the majority among the preferences achieves incentive compatibility.*

Proof. Each voter has only two preferences: if $a' \leq_i a$ but $f(P) = a$, then the only way for i to switch preferences is to prefer a over a' as well, but this cannot make f change. □

Example 8.7 (Tournament voting)**.** *An elimination tournament where alternatives are paired up. For each pair, we kick out of the tournament the alternative with the least number of preferences over the other. The problem with this system is that the final outcome strongly depends on the initial pairings.*

Example 8.8 (Pairwise majority)**.** *For each pair X, Y of alternatives, we say $X \leq Y$ if the majority of voters prefer X over Y. The problem with this system is that we may have cycles $X \leq Y \leq Z \leq X$, i.e. no minimal element.*

Example 8.9 (Borda count)**.** *Each player assigns a score to each alternative $a \in A$:*

$$s_i(a) = \left|\left\{a' \in A \;\middle|\; a \leq_i a'\right\}\right|$$

At this point, the Borda count of alternative a is defined as:

$$s(a) = \sum_i s_i(a)$$

and the social choice function will select the alternative with the highest Borda count. Also this system is not incentive compatible: if Y is winning with a Borda count of 9 followed by X with score 8, a voter which has, say, X as top preference and Y as second preference can make X win by reporting Y as lowest preference (this significantly decreases Y's score).

8.2 Arrow's theorem

Definition 8.10 (Unanimity). *A social welfare function F satisfies unanimity if $X \leq_{F(P)} Y$ when $X \leq_i Y$ for every voter i in P.*

For an order $\leq \in \mathcal{L}(A)$ we denote $\leq|_S \in \mathcal{L}(S)$ the restriction of the order to the subset of alternatives $S \subseteq A$. For two profiles, we write $P|_S$ as the profile containing the restriction to S for every voter.

Definition 8.11 (Independence of irrelevant alternatives). *A social welfare function F satisfies is independent of irrelevant alternatives when, for any pair $X, Y \in A$ and two profiles P, Q:*

$$P|_{\{X,Y\}} = Q|_{\{X,Y\}} \implies \leq_{F(P)}|_{\{X,Y\}} = \leq_{F(Q)}|_{\{X,Y\}}$$

i.e. if all voters express the same preference between X, Y in P and in Q, then also $F(P)$ and $F(Q)$ must agree in this preference.

Definition 8.12 (Dictatorship). *A voter i is a dictator for a social welfare function F if:*

$$F(P) \equiv \leq_i$$

in this case, F is said to be a dictatorship.

Lemma 8.13 (Polarizing lemma). *Let F be a social welfare function independent of irrelevant alternatives with unanimity. If P is a preference profile where every voter places an alternative X first or last (also called* ***polarizing alternative****), then also $F(P)$ must place X as first or last.*

Proof. Consider such a profile P and assume for a contradiction X is not neither first nor last, i.e.

$$Y \leq_{F(P)} X \leq_{F(P)} Z$$

for some $Y, Z \in A$. We construct a new profile Q where, for each player i with $Y \leq_i^P Z$, we move Z right in front of Y. Then X is still a polarizing alternative in Q, and also all the other preferences (except the ones regarding Z) are preserved.

- By independence of irrelevant alternatives, $Y \leq_{F(Q)} X \leq_{F(Q)} Z$;
- By unanimity, since $Z \leq_i^Q Y$ for every i, we must also have $Z \leq_{F(Q)} Y$.

These two statements lead to a contradiction. □

Theorem 8.14 (Arrow). *When $|A| \geq 3$, if a social welfare function F is independent of irrelevant alternatives with unanimity, then F is a dictatorship.*

Proof. Consider the voters $v_1, \ldots, v_n$ in an arbitrary order, and let P be an arbitrary preference profile. Moreover, let P_i be a profile obtained from P by moving X to the top for voters $v_1, \ldots, v_i$ and to the bottom for $v_{i+1}, \ldots, v_n$. By unanimity $F(P_0), F(P_n)$ have X in its last and first positions, respectively. By Lemma 8.13, also $F(P_1), \ldots, F(P_{n-1})$ must rank X as first or last, and this implies there must be a voter x such that $F(P_{x-1}), F(P_x)$ ranks X as last and first, respectively (notice that this value is not necessarily unique, but let us pick the lowest one say).

We complete the proof by showing that voter x is a dictator in F. Consider two preferences $Y, Z \neq X$ where $Y \leq_x^P Z$, and let R be a profile obtained from P_x by moving Y in front of X for voter x (i.e. X becomes second for voter x, and this does not change the order of Y, Z for voter x). Using independence of irrelevant alternatives, we deduce that:

- $R|_{\{X,Z\}} = P_x|_{\{X,Z\}}$ and $X \leq_{F(P_x)} Z$ (X is first in $F(P_x)$), thus $X \leq_{F(R)} Z$;
- $R|_{\{X,Y\}} = P_{x-1}|_{\{X,Y\}}$ and $Y \leq_{F(P_{x-1})} X$ (X is last in $F(P_{x-1})$), thus $Y \leq_{F(R)} X$;

and this proves $Y \leq_{F(R)} Z$ and, since $R|_{\{Y,Z\}} = P|_{\{Y,Z\}}$, we must also have $Y \leq_{F(P)} Z$.

We apply the exact same reasoning on an alternative $W \neq X$, obtaining a voter w and, since $|A| \geq 3$, we evince we have a third alternative $V \neq X, W$. Assuming that $x \neq w$, P_{x-1} and P_x differ only in voter x's preference, and $V \leq_{F(P_{x-1})} X$ but $X \leq_{F(P_x)} V$ (X goes from last to first). This means that $X, V \neq W$ swapped in order without a change of voter w, which contradicts the reasoning above. Therefore, $x = w$ and also pairs containing X follow the ordering of this voter, i.e. we have a dictator. □

Theorem 8.15 (Gibbard-Satterthwaite)**.** *Any incentive compatible social choice function with at least three alternatives is a dictatorship.*

8.3 Single-peaked preferences

Definition 8.16 (Single-peaked domain)**.** *A single-peaked domain is a setting where each voter has a utility $u_i : A \to \mathbb{R}$ which reaches a maximum in exactly one alternative, and decreases monotonically with the distance from the preference.*

Definition 8.17. *A social choice function f over single-peaked domain is incentive compatible if, for any voter i and any reported preference c_{-i}*

$$u_i(f(c_i^*, c_{-i})) \geq u_i(f(c_i, c_{-i})) \quad \forall c_i \in A$$

where c_i^* is the true preference of voter i.

9 Mechanisms without Money

9.1 House allocation

Definition 9.1 (House allocation setting)**.** *Each player has a house, and a preference total order of the houses. We would like to rearrange the players in such a way that each player gets a better house with respect to their preference order.*

Algorithm 9.2 (Top trading cycles algorithm)**.** *Construct a directed graph where each player points to the player with the best house (with respect to its preference order). Reallocate players in each cycle of this graph, then remove these players and houses from the game and repeat until all players are removed.*

Theorem 9.3. *Algorithm 9.2 converges in $\mathcal{O}(n)$ iterations.*

Proof. In the constructed directed graph, each vertex has out-degree exactly 1, hence each component of this graph has exactly one cycle, hence at least one player is removed at each iteration. □

Theorem 9.4. *Algorithm 9.2 never allocates a worse house than the initial one.*

Proof. A player does not change its house until it is excluded. Either it enters a non-trivial cycle (getting a strictly better house than the initial one) or every house better than its initial one is taken by other players and excluded, implying that this player ends up with its initial house (it declares a self-loop). □

Theorem 9.5. *Algorithm 9.2 is incentive compatible.*

Proof. Let N_i be the set of players excluded at iteration i. Players in N_1 get their top preference, so they cannot improve by lying. On the other hand, the only way for players in N_i to improve is to become part of N_k for some $k < i$, and in order to do this it is necessary to enter a cycle, but a player can only change is out-edge. □

Definition 9.6 (Blocking coalition)**.** *A subset of players $S \subseteq [n]$ is said to be a blocking coalition if, starting from their initial houses, they can allocate the houses in such a way that no player of S gets a strictly worse house, but one of them gets a strictly better one. An allocation of houses is said to be a core allocation if no blocking coalition is present.*

Theorem 9.7. *Algorithm 9.2 yields a core allocation.*

Proof. Suppose for a contradiction we have a blocking coalition $S \subseteq [n]$ and suppose without loss of generality that this blocking coalition is minimal, which implies that the reallocation implied by S is exactly one cycle C passing through all the players of the coalition.

If $S \subseteq N_i$ for some i, then the algorithm did not reallocate according to the top preferences at iteration i as it should have chosen the cycle C (or one that is even better than C), which contradicts the definition of the algorithm.

Hence, suppose that the cycle C extends to different N_i. This means that there must be a player in $C \cap N_i$ that points to a player in $C \cap N_k$ with $k > i$, i.e. reallocating according to C would give to such player a strictly worse house, contradicting the definition of blocking coalition. □

Theorem 9.8. *The core allocation of a house allocation instance is unique.*

Proof. Players in N_1 must have the houses allocated by TTCA, otherwise we would have a blocking coalition formed by those players of N_1 who did not get their top preference. By inductively applying this argument, we find that any core allocation must make the same choices as TTCA. □

9.2 Kidney exchange

Definition 9.9 (Kidney exchange setting)**.** *We have a graph $G = (V, E)$ where each vertex represents a patient-donor pair, and two vertices are adjacent if and only if both pairs want to switch the donors with each other.*

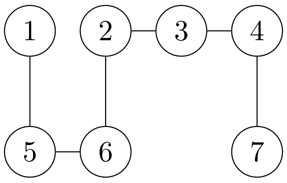
Each vertex reports the compatible vertices and an edge is added when both the endpoints report compatibility with the other.

Algorithm 9.10 (Prioritized matching)**.** *Fix an arbitrary order $v_1, \dots, v_n$ of the vertices. Starting with M_0 as the set of maximum matchings of G, construct M_i from M_{i-1} by excluding any matching that does not match v_i, or $M_i = M_{i-1}$ if v_i is never matched.*

Theorem 9.11. *The prioritized maximum matching algorithm is incentive compatible.*

Proof. It is sufficient to see that the matching returned by the algorithm does not change if we remove edges adjacent to an unmatched vertex. □

Incentive compatibility breaks down if we suppose that players are hospitals, which want to maximize the number of their matched vertices. In this graph (where top vertices are from hospital 1 and bottom vertices are from hospital 2), in any matching one of the two hospitals can hide edges, strictly increasing the number of its matched vertices.



9.3 Stable matching

Definition 9.12 (Stable matching)**.** *Consider a graph $G = (V, E)$, where each vertex $v \in V$ has a total order $<_v$ over its neighbors. A stable matching is a matching M such that, for any edge $(u, v) \in E \setminus M$, u, v do not prefer each other over their partners in M.*

From now on, we assume that $G = (A \cup B, E)$ is bipartite, reducing to a *stable marriage* problem. For sake of simplicity of notation, we assume that A is the set of men and B is the set of women. Also, we assume the bipartite graph is balanced and complete (thus it admits a perfect matching).

Algorithm 9.13 (Proposal algorithm)**.** *Take a man $u \in A$ and let him make a proposal to the top woman in their preference list. Upon proposal, woman discards or accepts the proposal (discarding previous requests). Men discard a rejecting woman from their preference lists. Repeat until all men are matched.*

Theorem 9.14. *Algorithm 9.13 converges in $\mathcal{O}(n^2)$ steps.*

Proof. A man asks a woman at most once, thus each man will make at most $\mathcal{O}(n)$ proposals, i.e. $\mathcal{O}(n^2)$ proposals overall. □

Theorem 9.15. *Algorithm 9.13 terminates with a perfect matching.*

Proof. Suppose for a contradiction that there is a pair (u, v) that is unmatched. u should have made the proposal to v sooner or later, and v should have accepted it in absence of better requests. A woman which gets a proposal is always matched with some man until the end of the algorithm, hence v should have been matched. □

Theorem 9.16. *Algorithm 9.13 yields a stable matching.*

Proof. Consider two unmatched persons u, v. Being unmatched means that either:

- u never proposed to v, which means that the woman u ended up with did not reject him (and it must be higher in u's preference list, by the order of proposals);
- v rejected u, which means that she received a better request.

In any case, one of u, v cannot prefer the other over its current match. $\square$

Theorem 9.17. *Let $h(u)$ be the best woman u can get in any stable matching. Algorithm 9.13 matches u with $h(u)$.*

Proof. Let A be the matching returned by the algorithm and, given a matching M, denote $M(u)$ the match of u in M.

Consider $(u, A(u))$ matched by the algorithm. Suppose for a contradiction that there is a stable matching $M \ni (u, M(u))$ with $M(u) <_u A(u)$. This means that u proposed to $M(u)$ (and was rejected) before proposing to $A(u)$, and we consider the first iteration when such rejection happens. The rejection means $M(u)$ received a proposal by $u' <_{M(u)} u$.

We deduce that $M(u) <_{u'} M(u')$: if this were not the case, then u' should have proposed to $M(u')$ first, and its rejection would occur before that of u (since it happens when u' proposes to $M(u)$), but this contradicts the fact that the rejection of u was the first.
This concludes the proof, as $u' <_{M(u)} u$ and $M(u) <_{u'} M(u')$ contradict the stability of M. $\square$

Theorem 9.18. *Algorithm 9.13 is incentive compatible for men.*

Proof. Suppose for a contradiction a man u lies and is matched with a better woman for this. Let M, M' be the resulting matchings of the algorithm for, respectively, a truth-telling and a lying u (keeping others fixed). Let R be the set of men who improve by switching from M to M', and define S as

$$S = \{M'(x) \mid x \in R\}$$

Let $v = M'(u)$, and notice that u prefers v over $M(u)$, by assumption. But v cannot prefer u over $M(v)$, otherwise M would not be stable (it is by Theorem 9.16). Hence,

$$M(v) <_v u$$

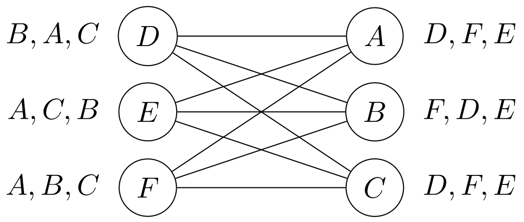
Now we use this to claim that $M(v) \in R$: if this is not true, then $M(v)$ would propose to v first, even with M', but this contradicts the fact that $(u, v) \in M'$, since $M(v) <_v u$. This argument shows that:

$$S = \{M(x) \mid x \in R\}$$

and, in particular, M, M' assigns two different men in R to each woman in S (since each man in R receives a strictly better match in M'). *Incomplete* $\square$

Theorem 9.19. *Algorithm 9.13 is not incentive compatible for women.*

Proof. Consider the following instance:



If the given preferences are the true ranks, the proposal algorithm gives the matching $A = \{\{D, B\}, \{E, C\}, \{F, A\}\}$. However, if A reports D, E, F, the algorithm matches her with D, which is better for her. $\square$

10 Best Response Mechanisms

10.1 Asynchronous best-response dynamics

Definition 10.1. *Let $G = (n, S, u)$ be a game in a state $s^0 \in S$. An asynchronous dynamics consists of an infinite sequence of steps, where at step t the game G is in state $s^t \in S$ and*

- *A non-empty subset $P_t \subseteq [n]$ of players is called to respond to s^t according to some response strategy $R_i : S \to S_i$;*
- *The state s^{t+1} is obtained from s^t by replacing s_i with $R_i(s^t)$ for every $i \in P_t$.*

The dynamics is said to converge if there exists a time step T such that:

$$s^t = s^T \quad \forall t > T$$

Every player must be called to respond an infinite number of times.

Definition 10.2 (Repeated game). *Let $G = (n, S, u)$ be a game. The repeated game G^* is defined as a game (n, S^*, u^*) where:*

- *$S_i^* = S_i^S$, i.e. the strategy of player i is a response strategy $R_i : S \to S_i$;*
- *G^* consists of an asynchronous dynamics on G as defined above, where s^0 is fixed and each player i responds to the current state s^t with $R_i(s^t)$;*
- *The utility u_i^* of player i is defined as:*

$$u_i^*(s) = \limsup_{t \to \infty} u_i(s^t)$$

Definition 10.3 (Incentive compatibility for best response). *Best response is incentive compatible in $G = (n, S, u)$ if, for any initial state $s^0 \in S$, the state $\bar{R} = (\bar{R}_1, \ldots, \bar{R}_n) \in S^*$ containing best response strategies is a pure Nash equilibrium in G^*, i.e.*

$$u_i^*(\bar{R}) \geq u_i^*(R_i', \bar{R}_{-i})$$

for any $i \in [n]$, $R_i \in S_i^*$.

Notice that best response strategies are unique up to total orders over S_i (tie breaking rules).

10.2 Never best response and convergence

Definition 10.4 (Never best response). *Let $G = (n, S, u)$ be a game. A strategy $s_i \in S_i$ is a never best response if, for some total order $\prec$ over S_i, for any $s_{-i} \in S_{-i}$, there exists $s_i' \in S_i$ such that:*

- *$u_i(s_i, s_{-i}) < u_i(s_i', s_{-i})$ or;*
- *$u_i(s_i, s_{-i}) = u_i(s_i', s_{-i})$ with $s_i' \prec s_i$.*

In other words, there exists a best response strategy $R_i \in S_i^*$ with $R_i(s) \neq s_i$ for any $s \in S$.

Definition 10.5 (NBR-solvability). *A game $G = (n, S, u)$ is NBR-solvable if, for some tie breaking rule $\prec$, we have a sequence of players $p_1, \ldots, p_\ell$ such that:*

- *$G_1 = G$ is the starting game;*
- *G_{i+1} is obtained from G_i by removing never best response strategies from S_{p_i};*

- *G_ℓ has only one possible state.*

Such sequence of players is called **elimination sequence** of G.

Theorem 10.6. *The unique state in G_ℓ is a pure Nash equilibrium in G.*

Proof. The dynamics above is achieved by a best response dynamics, which converges to a pure Nash equilibrium by Theorem 1.5. □

Definition 10.7 (Round)**.** *In an asynchronous dynamics, a round is a time interval where each player is called at least once. Since every player is called an infinite number of times, we have an infinite number of rounds. The first round starts at time $t = 1$.*

Theorem 10.8. *Asynchronous best-response dynamics converge for NBR-solvable games.*

Proof. We prove by induction on the rounds that in the k-th round the state of the game must be in G_k. Let r_j be the last step of the j-th round (assuming $r_0 = 0$). For $j = 0$, the state s^0 is certainly in $G_0 = G$. At the end of round $j > 0$, the sequence of players called in the dynamics must contain $p_1, \dots, p_j$ as subsequence (each of these players is called j times). By induction, $s^{r_{j-1}}$ is in G_{j-1}, and so also s^{r_j} is in G_{j-1}. Moreover, in the j round p_j is called once more, and it certainly does not choose any of its never best responses in G_{j-1}, i.e. the state enters G_j before the end of round j.

This also proves that, after ℓ rounds, the state is in G_ℓ, which is unique. □

10.3 Clear outcome and incentive compatibility

Example 10.9. *Consider the following game G:*

	A	*B*
A	*1* *2*	*0* *0*
B	*0* *3*	*2* *1*

(B, B) is the unique PNE but, in the repeated game G^*, a best-responding player 1 makes the dynamics converge to the PNE, which gives $u_1 = 2$, while a response strategy R_1 that makes the state oscillate in a counter-clockwise cycle yields:

$$u_i^*(R_1, \bar{R}_2) = \lim\sup_{t\to\infty} u_i(s^t) = 3$$

Definition 10.10 (Clear outcome)**.** *A NBR-solvable game $G = (n, S, u)$ has a clear outcome if, for every player p_a, there exists an elimination sequence $p_1, \dots, p_\ell$:*

- *When $p_a \neq p_1, \dots, p_{a-1}$, i.e. p_a appears for the first time;*
- *The PNE $s^* \in G_\ell$ is the global optimum for u_{p_a}, i.e.*

$$u_{p_a}(s) \leq u_{p_a}(s^*) \quad \forall s \in G_{a-1}$$

Theorem 10.11. *Best response in games with clear outcome is incentive compatible.*

Proof. Let $\{s^t\}_t$ and $\{\hat{s}^t\}_t$ be the sequence of states generated by an asynchronous dynamics where, respectively, every player best responds and every player but p_i best responds.

From Theorem 1.5 we know that $s^t = s^*$ for a sufficiently large t. Let $\bar{t}$ be the time step where player p_i is called for the first time after $p_1, \dots, p_{i-1}$, then from the proof of Theorem 10.8 the state will definitely be in G_{i-1} (since $p_1, \dots, p_{i-1}$ best respond), and the definition of clear outcome ensures that:

$$u_{p_i}(s) \leq u_{p_i}(s^*) \quad \forall s \in G_{i-1}$$

Since $s^t \in G_{i-1}$ for every $t > \bar{t}$, the pure Nash equilibrium will never give a worse utility to p_i, i.e. best responding always gives maximum utility. □

10.4 Gao-Rexford model in BGP

Definition 10.12 (BGP theoretical setting)**.** *Let $H = (V, E)$ be a graph, where each vertex is a player, except for a node d which is the destination node. Each player wants to send a message to node d, and can choose a strategy which is a possible path to d. If i's traffic is discarded along the chosen path, this path gives null utility to i.*

We give a type to each edge of G:

- $u \to v$, i.e. u is a **customer** of v;
- $u \leftarrow v$, i.e. u is a **provider** of v;
- $u \leftrightarrow v$, i.e. u and v are **peers**.

From a fixed vertex u, we denote a path to d as **customer**, **provider**, or **peer** path, depending on the relationship of the next hop in this path with u.

Definition 10.13 (Gao-Rexford model)**.** *The Gao-Rexford model imposes the following three constraints on the players:*

GR1. If $\prec$ is the order of preference of a player then

$$\textit{customer path} \prec \textit{peer path} \prec \textit{provider path} \prec \emptyset$$

i.e. customer paths are the most preferred, while $\emptyset$ (indicating that the traffic should be discarded) is the least preferred.

GR2. Allow transit traffic only if either previous or next hop are customers.

GR3. No player is indirectly a provider of itself.

Definition 10.14 (Dispute wheel)**.** *A dispute wheel is a cycle of players $p_1, \dots, p_\ell, p_{\ell+1} \equiv p_1$ all connected to d where p_i prefers forwarding traffic to player p_{i+1} rather than going directly to d. Note that between p_i and p_{i+1}, as well as between p_i and d, there may be a path of length ≥ 2.*

A dispute wheel may make the best response dynamics diverge (no PNE is present).

Lemma 10.15. *The Gao-Rexford constraints do not allow a dispute wheel.*

Proof. Consider a wheel of players $p_1, \dots, p_\ell$. Let Q_i be the path from p_i to d, while R_i is the path from p_i to p_{i+1} in the wheel. If this wheel forms a dispute wheel this implies that:

$$R_i Q_{i+1} \prec_{p_i} Q_i$$

i.e. p_i prefers to forward the traffic to the next player in the wheel. This means that p_{i+1} will forward the traffic to d and, by GR2, either

- R_i is a series of customer-provider edges (in particular, $p_i \to^* p_{i+1}$ indirectly);
- Q_{i+1} is a series of provider-customer edges (in particular, $d \to^* p_{i+1}$).

where $\to^*$ is the transitive closure of $\to$. If the first case holds for the entire wheel, then we would have:

$$p_1 \to^* p_2 \to^* \cdots \to^* p_\ell \to^* p_1$$

i.e. $p_1 \to^* p_1$, contradicting GR3. Thus, we must have $p_i \not\to^* p_{i+1}$ for some i and, in order to have the above preference for p_i, the second case must hold. At this point, p_{i+1} prefers sending directly to d through Q_{i+1}, since this is a customer path, by GR1, contradicting the fact that this wheel forms a dispute wheel. Notice that GR3 is needed because, if $p_1 \to^* p_1$, GR1 would allow a dispute wheel in the opposite direction. $\square$

Definition 10.16 (Happy path). *A happy path P is a path of G ending in d, where, for each vertex v, the subpath of P from v to d is v's top preference in subgame G_v.*

By definition, a vertex containing a happy path P must have the subpath of P to d as its top preference (when the preceding vertices in the path have excluded their best responses).

Lemma 10.17. *If no dispute wheel is present, best response in BGP is incentive compatible.*

Proof. It is sufficient to prove that, without a dispute wheel, the current subgame contains a happy path. At this point, we can replace the happy path with d and inductively reapply the argument. If no happy path is present, than we can construct a dispute wheel in the same way as we did in the lecture notes. □

10.5 TCP games

Definition 10.18 (TCP game). *We have a channel and n players which need to send a flow of data through this channel. A strategy $s_i \in S_i = [0, M_i]$ for player i is the rate at which this player sends its traffic through the channel (M_i is the maximum rate the player wants/needs to send). The utility u_i is the rate $r_i \in [0, s_i]$ of player i's data actually transmitted (and not dropped) by the channel. We denote as C the total capacity of the channel, i.e. the social welfare cannot exceed C.*

Algorithm 10.19 (Probing Increase Educated Decrease). *Player i sends with rate exactly s_i^*, where s_i^* is defined as:*

$$s_i^* := \max\{s_i \in [0, M_i] \mid r_i(s_i, s_{-i}) = s_i\}$$

We want to define an algorithm for the channel to distribute rates among the players.

Algorithm 10.20 (Proportional sharing). $r_i \leftarrow C\frac{s_i}{\sum_j s_j}$

Theorem 10.21. *PIED is not incentive compatible with proportional sharing.*

Proof. If we have n players with $M_i = C$, PIED would prescribe to send at rate C/n. However, if s_{-i} has all C/n, player i would strictly increase its utility by raising its sending rate. □

Algorithm 10.22 (Strict priority queuing). *Order the players arbitrarily. Then, for $i = 1, \dots, n$ assign:*

$$r_i \leftarrow \min\left(s_i, C - \sum_{j=1}^{i-1} r_j\right)$$

Theorem 10.23. *PIED is incentive compatible with strict priority queuing.*

Proof. We construct an elimination sequence with clear outcome, which follows the order of the queue. The first player p_1 sets its best response

$$s_1^* = \min(M_1, C)$$

(the tie breaking rule is the natural order $\leq$ over the reals, i.e. we take the smallest maximizer). The channel policy allocates the above regardless of the other players, thus $s_1 \neq s_1^*$ are never best responses. When we reach player i, players from $1, \dots, i-1$ have only one strategy by induction, and the best response of player i is:

$$s_i^* = \min\left(M_i, C - \sum_{j=1}^{i-1} s_j^*\right)$$

while others are never best responses for the same reason as above. Notice that the outcome is clear, because when player i is called, s_i^* will also be its utility, which is fixed and attained also at the final pure Nash equilibrium. □

Algorithm 10.24 (Fair queuing)**.** *Allocate evenly among players:*

$$r_i \leftarrow \min(C/n, s_i)$$

Recursively reallocate the residual capacity:

$$C \leftarrow C - \sum_i r_i \quad s_i \leftarrow s_i - r_i$$

until $C = 0$ or $s_i = 0$ for all i.

Theorem 10.25. *PIED is incentive compatible with fair queuing.*

Proof. We construct an elimination sequence based on the iterations of the algorithm. At the k-th iteration, we have n^k players with residual rate M_i^k, and the channel has total capacity C^k. We say that a player i survives at iteration k if $M_i^k > C^k/n^k$. For each iteration k we add the players who are still alive to the elimination sequence. A player surviving iteration k will have that any strategy

$$s_i < \max\left(M_i, \sum_{i=1}^{k} \frac{C^k}{n^k}\right)$$

is a never best response (this is trivially true in the first iteration, thus will be true in any iteration by induction, as players are called to remove the never best responses for each iteration). Thus, any player reaching iteration k will always reach it after these strategies are removed.

- If $M_i^k \leq C^k/n^k$, then player i stops at iteration k and the algorithm allocates all M_i. This means that $s_i \neq M_i$ are all never best responses.
- If $M_i^k > C^k/n^k$, then the algorithm will allocate all C^k/n^k for player i in this round and player i will always obtain at least $\sum_{i=1}^{k} \frac{C^k}{n^k}$ in this subgame (thus, any lower sending rate is a never best response).

The outcome is trivially clear for players who stop at some iteration, since they manage to send M_i in the final pure Nash equilibrium. On the other hand, the outcome is clear also for players who survive all the iterations: their utility can increase only if C^k increases or n^k decreases for some iteration, but this requires that some player switches to a strategy who makes them not survive to iteration k, and this player already excluded all such strategies in previous iterations. □

10.6 Stable matching

Theorem 10.26. *A stable matching may not exist in general graphs.*

Proof. Consider a triangle with cyclic preferences. □

We restrict to the example of interns-hospitals, where n hospitals have the same rank of n interns:

$$i_1 \prec \cdots \prec i_n$$

Definition 10.27 (Best response mechanism for stable matchings)**.** *In a best response mechanism, each vertex proposes to its best unmatched preference.*

Notice that a preference can be naturally translated to a utility function through ranks.

Theorem 10.28. *Best response is incentive compatible in the interns-hospitals setting.*

Proof. By the fact that the hospitals have the same rank, the elimination sequence $i_1, \cdots, i_n$ is valid: i_k's choice does not depend on $i_{k+1}, \ldots, i_n$, since any hospital would prefer i_k. Thus, i_k has its unmatched top preference as best response (and other as never best responses, as they are lower in its rank or already taken by $i_1, \ldots, i_{k-1}$.

The clear outcome comes from the fact that the choices of $i_{k+1}, \ldots, i_n$ cannot change i_k's match (and thus its utility will be unchanged), implying that the utility of i_k at the moment of its call is the same as in the final pure Nash equilibrium. □

Corollary 10.29. *The proposal algorithm is incentive compatible in the interns-hospitals setting.*

Proof. The proposal algorithm is simply a best response mechanism where the sequence of calls are arbitrary, and an intern will keep proposing if it gets rejected by some hospital. □

10.7 Single item auction

Definition 10.30 (Repeated first price auction)**.** *Let $G = (n, S, u)$ be a single item auction, where strategies of players are bids $b_i \in \mathbb{R}$, each player has a private valuation $v_i \in \mathbb{R}$. The utility of player i will be*

$$u_i(b) = \begin{cases} v_i - b_i & \textit{if } b_i \textit{ is the highest} \\ 0 & \textit{otherwise} \end{cases}$$

breaking ties arbitrarily. The repeated first price auction is the closure $G^* = (n, S^*, u^*)$ of G.

Theorem 10.31. *Best response strategies are incentive compatible in repeated first price auction with two bidders.*

Proof idea. Assume without loss of generality v_1 is the highest true valuation. Let G_δ be a game where possible bids are discretized in steps of δ. At this point, the pure Nash equilibrium is reached with an elimination sequence $p_1, \ldots, p_\ell$ alternating the two players for $\frac{2}{\delta} v_2$ steps.

A best response dynamics should go in steps of δ: the choice of a player does not depend on the true valuation of the other player. □

11 Sponsored Search

11.1 VCG vs GSP

Definition 11.1 (Sponsored search auction)**.** *We have n bidders and k slots. Slot j has a click-through-rate α_j, and player i has a private value-per-click v_i. We assume $\alpha_1 \geq \cdots \geq \alpha_k$. The utility of player i will be $\alpha_i \cdot v_i$, where α_i is the slot assigned to player i (or 0 is $i > k$).*

From now on, we will assume without loss of generality that bidders are sorted by descending bid.

$$b_1 \geq \cdots \geq b_n$$

and that $\alpha_{k+1} = \cdots = \alpha_n = 0$.

Definition 11.2 (VCG mechanism for sponsored search)**.** *Let (A, P) be a VCG mechanism where:*

- *A is an algorithm that greedily assigns α_i to b_i;*
- *P charges player i with:*

$$P_i(b) = \sum_{\ell=i}^{k} (\alpha_\ell - \alpha_{\ell+1}) \cdot b_{\ell+1}$$

Theorem 11.3. *The mechanism defined above is indeed a VCG mechanism.*

Proof. Algorithm A is a greedy optimum, and this can be proven by an exchange argument. The payment function P can be rewritten:

$$\begin{aligned}
P_i(b) &= \sum_{\ell=i}^{k} (\alpha_\ell - \alpha_{\ell+1}) \cdot b_{\ell+1} \\
&= \sum_{\ell=i}^{k} \alpha_\ell \cdot b_{\ell+1} - \sum_{\ell=i}^{k} \alpha_{\ell+1} \cdot b_{\ell+1} \\
&= \sum_{\ell=i}^{k} \alpha_\ell \cdot b_{\ell+1} - \sum_{\ell=i+1}^{n} \alpha_\ell \cdot b_\ell \\
&= \sum_{\ell=1}^{i-1} \alpha_\ell \cdot b_\ell + \sum_{\ell=i}^{k} \alpha_\ell \cdot b_{\ell+1} - \sum_{\ell \neq i} \alpha_\ell \cdot b_\ell \\
&=: Q_i(b_{-i}) - \sum_{\ell \neq i} b_\ell(A(b))
\end{aligned}$$

which is exactly the definition of VCG mechanism. □

Definition 11.4 (GSP mechanism)**.** *Let (A, P') be a mechanism where:*

- *A, again, greedily assigns higher slots to higher bids;*
- *P' charges players with:*

$$P'_i(b) = \alpha_i b_{i+1}$$

Theorem 11.5. *$P'_i(b) \geq P_i(b)$, thus GSP yields an higher revenue than VCG.*

Proof.

$$\begin{aligned}
P_i(b) &= \sum_{\ell=i}^{k}(\alpha_\ell - \alpha_{\ell+1}) \cdot b_{\ell+1} \\
&= \sum_{\ell=i}^{k} \alpha_\ell \cdot b_{\ell+1} - \sum_{\ell=i}^{k} \alpha_{\ell+1} \cdot b_{\ell+1} \\
&= \sum_{\ell=i}^{k} \alpha_\ell \cdot b_{\ell+1} - \sum_{\ell=i+1}^{k} \alpha_\ell \cdot b_\ell \\
&= \alpha_i \cdot b_{i+1} + \sum_{\ell=i+1}^{k} \alpha_\ell (b_{\ell+1} - b_\ell) \\
&\leq \alpha_i \cdot b_{i+1} = P'_i(b) && \text{since } b_{\ell+1} \leq b_\ell
\end{aligned}$$

□

The utility of player i will be:

$$u_i(b) = \alpha_i v_i - \alpha_i b_{i+1} = \alpha_i(v_i - b_{i+1})$$

Theorem 11.6. *The GSP mechanism is not truthful.*

Proof. Consider an instance with three players and three slots where $\alpha_1 = 50, \alpha_2 = 40, \alpha_1 0$. We have to bids:

$$\begin{aligned}
b &= (25, 30, 5) \\
b' &= (25, 15, 5)
\end{aligned}$$

If 30 is middle player's true valuation, its utility will be:

$$\begin{aligned}
u(b|30) &= 50(30 - 25) = 250 \\
u(b'|30) &= 40(30 - 5) = 1000
\end{aligned}$$

i.e. lying gives a strictly higher utility. □

11.2 Nash equilibria in GSP

Observation 11.7. *A set of bids b is a pure Nash equilibrium if and only if:*

$$\begin{aligned}
\alpha_s(v_s - p_s) &\geq \alpha_t(v_s - p_t) && \textit{for } t > s \\
\alpha_s(v_s - p_s) &\geq \alpha_t(v_s - p_{t-1}) && \textit{for } t < s
\end{aligned}$$

where $p_s = b_{s+1}$.

Proof. The utility of player s when following a pure Nash equilibrium b is:

$$u_s(b_s, b_{-s}) = \alpha_s(v_s - b_{s+1}) = \alpha_s(v_s - p_s)$$

If $t > s$, this means that bidder s lowers its bid enough to get slot t. In the new configuration, bidders with slots $> t$ are untouched, so b_{t+1} did not change, and the definition of pure Nash equilibrium would be:

$$\begin{aligned}
\alpha_s(v_s - b_{s+1}) &\geq \alpha_t(v_s - b_{t+1}) \\
\alpha_s(v_s - p_s) &\geq \alpha_t(v_s - p_t)
\end{aligned}$$

If $t < s$, bidder s is raising its bid enough to get slot t, and the bidder getting slot $t+1$ in the new configuration is exactly the bidder with slot t now, i.e. the definition of pure Nash equilibrium yields the following:

$$\begin{aligned} \alpha_s(v_s - b_{s+1}) &\geq \alpha_t(v_s - b_t) \\ \alpha_s(v_s - p_s) &\geq \alpha_t(v_s - p_{t-1}) \end{aligned}$$

□

Example 11.8. *Consider the following example:* $\alpha = (1, 0.7, 0.1), v = (20, 10, 5)$*, and two bids:*

$$\begin{aligned} b &= (10, 6, 30/7) \\ b' &= (6, 10, 30/7) \end{aligned}$$

Both bids are pure Nash equilibria, the social welfare is:

$$\begin{aligned} SW(b) &= 1 \cdot (20 - 6) + 0.7 \cdot (10 - 30/7) + 0.1 \cdot 5 = 18.5 \\ SW(b') &= 1 \cdot (10 - 6) + 0.7 \cdot (20 - 30/7) + 0.1 \cdot 5 = 15.5 \end{aligned}$$

which means that the price of anarchy is > 1.

Definition 11.9. *A symmetric Nash equilibrium in GSP satisfies, for every bidder* s*:*

$$\alpha_s(v_s - p_s) \geq \alpha_t(v_s - p_t) \quad \textit{for every } t$$

Theorem 11.10. *Every symmetric Nash equilibrium maximizes the social welfare.*

Proof. In order to prove the claim if suffices to show that, whenever $\alpha_s > \alpha_{s+1}$, $v_s \geq v_{s+1}$. Then an exchange argument applies as the allocation is the same as the greedy approach used by GSP and VCG. From the definition of symmetric Nash equilibrium we have:

$$\begin{cases} \alpha_s(v_s - p_s) \geq \alpha_{s+1}(v_s - p_{s+1}) \\ \alpha_{s+1}(v_{s+1} - p_{s+1}) \geq \alpha_s(v_{s+1} - p_s) \end{cases} \implies \begin{cases} (\alpha_s - \alpha_{s+1})v_s \geq \alpha_s p_s - \alpha_{s+1} p_{s+1} \\ (\alpha_{s+1} - \alpha_s)v_{s+1} \geq \alpha_{s+1} p_{s+1} - \alpha_s p_s \end{cases}$$

which yields, by adding the two inequalities:

$$(\alpha_s - \alpha_{s+1})(v_s - v_{s+1}) \geq 0$$

Since $\alpha_s > \alpha_{s+1}$, we must have $v_s \geq v_{s+1}$. □

Theorem 11.11 (Characterization of symmetric Nash equilibria)**.** *A set of bids* b *is a symmetric Nash equilibrium if and only if, for every bidder* s*:*

$$\begin{aligned} \alpha_s(v_s - p_s) &\geq \alpha_{s-1}(v_s - p_{s-1}) \\ \alpha_s(v_s - p_s) &\geq \alpha_{s+1}(v_s - p_{s+1}) \end{aligned}$$

where $p_s = b_{s+1}$.

Proof. If b is a symmetric Nash equilibrium, then the condition is easily derived by plugging $t = s-1, s+1$.

On the other hand, if the condition holds, then $v_i \geq v_{i+1}$ for every i, since the conditions of the claim are identical of the conditions in the proof of Theorem 11.10. Consider two bidders s_1, s_2 with $k = s_2 - s_1 \geq 1$. We now proceed by induction on k:

- If $k \leq 1$, the claim trivially holds, as the condition we already have, plugging $s = s_1$, does the job;

- If $k > 1$, we have the induction hypothesis:

  $$\alpha_{s_1}(v_{s_1} - p_{s_1}) \geq \alpha_{s_1+k-1}(v_{s_1} - p_{s_1+k-1})$$

  where $s_1 + k - 1 = s_2 - 1$. Putting together the following two conditions:

  $$\begin{cases} \alpha_{s_1}(v_{s_1} - p_{s_1}) \geq \alpha_{s_2-1}(v_{s_1} - p_{s_2-1}) \\ \alpha_{s_2-1}(v_{s_2-1} - p_{s_2-1}) \geq \alpha_{s_2}(v_{s_2-1} - p_{s_2}) \end{cases}$$

  We rearrange the terms:

  $$\begin{cases} (\alpha_{s_1} - \alpha_{s_2-1})v_{s_1} \geq \alpha_{s_1}p_{s_1} - \alpha_{s_2-1}p_{s_2-1} \\ (\alpha_{s_2-1} - \alpha_{s_2})v_{s_2-1} \geq \alpha_{s_2-1}p_{s_2-1} - \alpha_{s_2}p_{s_2} \end{cases}$$

  Using $v_{s_1} \geq v_{s_2-1}$ and adding the two inequalities to obtain:

  $$(\alpha_{s_1} - \alpha_{s_2})v_{s_1} \geq \alpha_{s_1}p_{s_1} - \alpha_{s_2}p_{s_2} \iff \alpha_{s_1}(v_{s_1} - p_{s_1}) \geq \alpha_{s_2}(v_{s_1} - p_{s_2})$$

  concluding the induction.

The exact same reasoning can be done for the other condition. □

11.3 Bounds on revenues for GSP

Lemma 11.12. *If b is a symmetric Nash equilibrium the following holds for every bidder s:*

$$\alpha_s b_{s+1} + (\alpha_{s-1} - \alpha_s)v_s \leq \alpha_{s-1}b_s \leq \alpha_s b_{s+1} + (\alpha_{s-1} - \alpha_s)v_{s-1}$$

Proof. We start again with the two conditions:

$$\begin{aligned} \alpha_s(v_s - p_s) &\geq \alpha_{s+1}(v_s - p_{s+1}) \\ \alpha_{s+1}(v_{s+1} - p_{s+1}) &\geq \alpha_s(v_{s+1} - p_s) \end{aligned}$$

Rearranging the terms we get:

$$(\alpha_s - \alpha_{s+1})v_s + \alpha_{s+1}p_{s+1} \geq \alpha_s p_s \geq (\alpha_s - \alpha_{s+1})v_{s+1} + \alpha_{s+1}p_{s+1}$$

Plugging $p_s = b_{s+1}$ and adjusting the indices (this holds for every s), we get the claim. □

Corollary 11.13. *GSP always yields a revenue at least as good as the truthful VCG equilibrium.*

Proof. The lower bound of Lemma 11.12 can be rewritten as:

$$\begin{aligned} P'_s(b) &\geq P'_{s+1}(b) + (\alpha_s - \alpha_{s+1})v_{s+1} \\ &\geq P'_{s+2}(b) + (\alpha_{s+1} - \alpha_{s+2})v_{s+2} + (\alpha_s - \alpha_{s+1})v_{s+1} \\ &\geq \cdots \\ &\geq P'_{k+1}(b) + \sum_{\ell=s}^{k}(\alpha_\ell - \alpha_{\ell+1})v_{\ell+1} \\ &= \sum_{\ell=s}^{k}(\alpha_\ell - \alpha_{\ell+1})v_{\ell+1} = P_s(v) \end{aligned}$$

□

Corollary 11.14. *The best symmetric Nash equilibrium yields the same revenue as the best pure Nash equilibrium.*