Seminar of Computer Networks:

Online Social Networks and Network Economics

Sapienza University of Rome
Academic Year 2010/2011

Prof. Guido Schäfer

Center for Mathematics and Computer Science (CWI)
Algorithms, Combinatorics and Optimization
Science Park 123, 1098 XG Amsterdam, The Netherlands

VU University Amsterdam
Department of Econometrics and Operations Research
De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands

Website: http://www.cwi.nl/~schaefer
Email: g.schaefer@cwi.nl

Document last modified:
June 8, 2011

Disclaimer: These lecture notes contain most of the material (and some additional one) of the lectures of the course "Online Social Networks and Network Economics", given at Sapienza University or Rome in Spring 2011. The flow of the lecture notes might differ from the one that I followed in class; also the notation might be slightly different here and there. Note that the lecture notes have undergone some rough proof-reading only. Please feel free to report any typos, mistakes, inconsistencies, etc. that you observe by sending me an email (g.schaefer@cwi.nl).

Contents

1 Potential Games **1**
1.1 Motivating Example: Connection Game 1
1.2 Potential Games and the Finite Improvement Property 3
1.3 Existence of Pure Nash Equilibria 5
1.4 Price of Anarchy and Price of Stability 5

2 Selfish Routing **7**
2.1 Motivating Example: Pigou Example 7
2.2 Model 8
2.3 Nash Flows and their Existence 8
2.4 Optimal Flows 11
2.5 Price of Anarchy 11
2.6 Upper Bounds on the Price of Anarchy 12
2.7 Lower Bounds on the Price of Anarchy 15
2.8 Motivating Example: Braess's Paradox 15
2.9 Detecting Braess's Paradox 16
2.10 Mini-introduction: Computational Complexity 17
2.11 Detecting Braess's Paradox — Continued 19

3 Congestion Games **21**
3.1 Model 21
3.2 Example: Atomic Network Congestion Game 21
3.3 Congestion Games are Exact Potential Games 22
3.4 Price of Anarchy 22

4 Smoothness of Games **24**
4.1 Correlated Equilibria 24
4.2 Learning and Regret Matching Strategies 26
4.2.1 Example: Play Against Nature 26
4.2.2 Regret Matching in Strategic Games 27
4.3 Coarse Correlated Equilibria 29
4.4 Robust Price of Anarchy 29
4.5 Example: Congestion Games 31

5 Combinatorial Auctions **32**
5.1 Vickrey Auction 32
5.2 Combinatorial Auctions and the VCG Mechanism 34
5.3 Single-Minded Bidders 36
5.4 Generalized Second-Price Auction 37
5.4.1 Examples 39
5.4.2 Existence of Equilibria 39
5.4.3 Price of Anarchy 41

1 Potential Games

In this section, we consider so-called *potential games* which constitutes a large class of strategic games having some nice properties. We will address issues like the existence of pure Nash equilibria, price of stability, price of anarchy and computational aspects.

1.1 Motivating Example: Connection Game

As a motivating example, we first consider the following *connection game*.

Definition 1.1. A *connection game* $\Gamma = (G = (V,A), (c_a)_{a\in A}, N, (s_i,t_i)_{i\in N})$ is given by

- a directed graph $G = (V,A)$;
- non-negative arc costs $c : A \to \mathbb{R}_+$;
- a set of players $N := [n]$;
- for every player $i \in N$ a terminal pair $(s_i,t_i) \in V \times V$.

The goal of each player $i \in N$ is to connect his terminal vertices s_i, t_i by buying a directed path P_i from s_i to t_i at smallest possible cost. Let $S = (P_1,\ldots,P_n)$ be the paths chosen by all players. The cost of an arc $a \in A$ is shared equally among the players that use this arc. That is, the total cost that player i experiences under strategy profile S is

$$c_i(S) := \sum_{a\in P_i} \frac{c_a}{n_a(S)},$$

where

$$n_a(S) = |\{i \in N \;:\; a \in P_i\}|.$$

Let $A(S)$ be the set of arcs that are used with respect to S, i.e., $A(S) := \cup_{i\in N} P_i$. The social cost of a strategy profile S is given by the sum of all arc costs used by the players:

$$C(S) := \sum_{a\in A(S)} c_a = \sum_{i\in N} c_i(S).$$

Example 1.1. Consider the connection game in Figure 1 (a). There are two Nash equilibria: One in which all players choose the left arc and one in which all players choose the right arc. Certainly, the optimal solution is to assign all players to the left arc. The example shows that the price of anarchy can be as large as n.

Example 1.2. Consider the connection game in Figure 1 (b). Here the unique Nash equilibrium is that every player uses his direct arc to the target vertex. The resulting cost is

$$H_n := 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n},$$

which is called the *n-th harmonic number*. (H_n is about $\log(n)$ for large enough n.) An optimal solution allocates all players to the $1+\varepsilon$ path. The example shows that the cost of a Nash equilibrium can be a factor H_n away from the optimal cost.

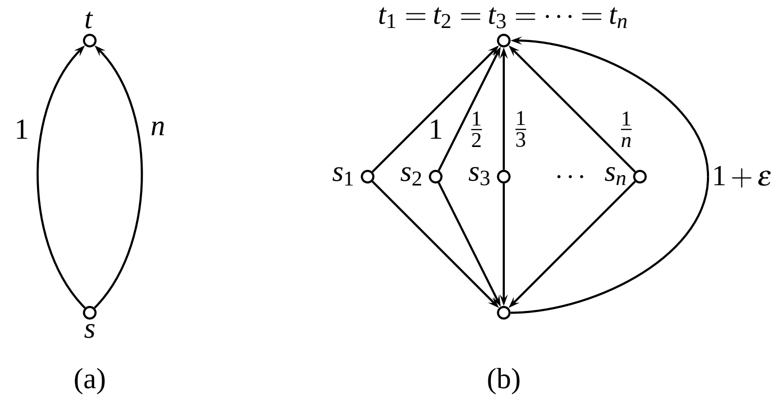

Figure 1: Examples of connection games showing that (a) Nash equilibria are not unique and (b) the price of stability is at least H_n.

Consider the following *potential function* Φ that maps every strategy profile $S = (P_1, \ldots, P_n)$ of a connection game to a real value:

$$\Phi(S) := \sum_{a \in A} c_a \left(1 + \frac{1}{2} + \cdots + \frac{1}{n_a(S)}\right) = \sum_{a \in A} c_a H_{n_a(S)}.$$

We derive some properties of $\Phi(S)$.

Lemma 1.1. *Consider an instance $\Gamma = (G, (c_a)_{a \in A}, N, (s_i, t_i)_{i \in N})$ of the connection game. We have for every strategy profile $S = (P_1, \ldots, P_n)$:*

$$C(S) \leq \Phi(S) \leq H_n C(S).$$

Proof. Recall that $A(S)$ refers to the set of arcs that are used in S. We first observe that $H_{n_a(S)} = 0$ for every arc $a \in A \setminus A(S)$ since $n_a(S) = 0$. Next observe that for every arc $a \in A(S)$ we have $c_a \leq c_a H_{n_a(S)} \leq c_a H_n$. Summing over all arcs concludes the proof. □

For a given strategy profile $S = (P_1, \ldots, P_n)$ we use (S_{-i}, P'_i) to refer to the strategy profile that we obtain from S if player i deviates to path P'_i, i.e.,

$$(S_{-i}, P'_i) = (P_1, \ldots, P_{i-1}, P'_i, P_{i+1}, \ldots, P_n).$$

The next lemma shows that the potential function reflects exactly the change in cost of a player if he deviates to an alternative strategy.

Lemma 1.2. *Consider an instance $\Gamma = (G, (c_a)_{a \in A}, N, (s_i, t_i)_{i \in N})$ of the connection game and let $S = (P_1, \ldots, P_n)$ be a strategy profile. Fix a player $i \in N$ and let $P'_i \neq P_i$ be an alternative s_i, t_i-path. Consider the strategy profile $S' = (S_{-i}, P'_i)$ that we obtain if player i deviates to P'_i. Then*

$$\Phi(S') - \Phi(S) = c_i(S') - c_i(S)$$

Proof. Note that for every $a \notin P_i \cup P_i'$ we have $n_a(S') = n_a(S)$. Moreover, for every $a \in P_i \cap P_i'$ we have $n_a(S') = n_a(S)$. We thus have

$$\begin{aligned}
\Phi(S') - \Phi(S) &= \sum_{a\in A} c_a H_{n_a(S')} - \sum_{a\in A} c_a H_{n_a(S)} \\
&= \sum_{a\in P_i'\setminus P_i} c_a(H_{n_a(S')} - H_{n_a(S)}) - \sum_{a\in P_i\setminus P_i'} c_a(H_{n_a(S)} - H_{n_a(S')}) \\
&= \sum_{a\in P_i'\setminus P_i} c_a(H_{n_a(S)+1} - H_{n_a(S)}) - \sum_{a\in P_i\setminus P_i'} c_a(H_{n_a(S)} - H_{n_a(S)-1}) \\
&= \sum_{a\in P_i'\setminus P_i} \frac{c_a}{n_a(S)+1} - \sum_{a\in P_i\setminus P_i'} \frac{c_a}{n_a(S)} = c_i(S') - c_i(S).
\end{aligned}$$

□

We will see in the next section that the above two lemmas imply the following theorem.

Theorem 1.1. *Let $\Gamma = (G, (c_a)_{a\in A}, N, (s_i,t_i)_{i\in N})$ be an instance of the connection game. Then Γ has a pure Nash equilibrium and the price of stability is at most H_n, where n is the number of players.*

1.2 Potential Games and the Finite Improvement Property

The above connection game is a special case of the general class of *potential games*, which we formalize next.

Definition 1.2. A finite strategic game $\Gamma = (N, (X_i)_{i\in N}, (u_i)_{i\in N})$ is given by

- a finite set $N = [n]$ of players;
- for every player $i \in N$, a finite set of strategies X_i;
- for every player $i \in N$, a utility function $u_i : X \to \mathbb{R}$ which maps every strategy profile $x \in X := X_1 \times \cdots \times X_n$ to a real-valued utility $u_i(x)$.

The goal of every player is to choose a strategy $x_i \in X_i$ so as to maximize his own utility $u_i(x)$.

A strategy profile $x = (x_1, \dots, x_n) \in X$ is a *pure Nash equilibrium* if for every player $i \in N$ and every strategy $y_i \in X_i$, we have

$$u_i(x) \geq u_i(x_{-i}, y_i).$$

Here x_{-i} denotes the strategy profile $(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$ excluding player i. Moreover, $(x_{-i}, y_i) = (x_1, \dots, x_{i-1}, y_i, x_{i+1}, \dots, x_n)$ refers to the strategy profile that we obtain from x if player i deviates to strategy y_i.

In general, Nash equilibria are not guaranteed to exist in strategic games. Suppose x is not a Nash equilibrium. Then there is at least one player $i \in N$ and a strategy $y_i \in X_i$

Algorithmus 1 IMPROVING MOVES

Input: arbitrary strategy profile $x \in X$
Output: Nash equilibrium x^*

$x^0 := x$
$k := 0$
while x^k is not a Nash equilibrium **do**
  determine a player $i \in N$ and $y_i \in X_i$, such that $u_i(x^k_{-i}, y_i) > u_i(x^k)$
  $x^{k+1} := (x^k_{-i}, y_i)$
  $k := k + 1$
end while
return $x^* := x^k$

such that

$$u_i(x) < u_i(x_{-i}, y_i).$$

We call the change from strategy x_i to y_i of player i an *improving move*.

A natural approach to determine a Nash equilibrium is as follows: Start with an arbitrary strategy profile $x^0 = x$. As long as there is an improving move, execute this move. The algorithm terminates if no improving move can be found. Let the resulting strategy profile be denoted by x^*. A formal description of the algorithm is given in Algorithm 1. Clearly, the algorithm computes a pure Nash equilibrium if it terminates.

Definition 1.3. We associate a directed *transition graph* $G(\Gamma) = (V, A)$ with a finite strategic game $\Gamma = (N, (X_i)_{i \in N}, (u_i)_{i \in N})$ as follows:

- every strategy profile $x \in X$ corresponds to a unique node of the transition graph $G(\Gamma)$;
- there is a directed edge from strategy x to $y = (x_{-i}, y_i)$ in $G(\Gamma)$ iff the change from x_i to y_i corresponds to an improving move of player $i \in N$.

Note that the transition graph is finite since the set of players N and the strategy set X_i of every player are finite. Every directed path $P = (x^0, x^1, \dots)$ in the transition graph corresponds to a sequence of improving moves. We therefore call P an *improvement path*. We call x^0 the *starting configuration* of P. If P is finite its last node is called the *terminal configuration*.

Definition 1.4. A strategic game $\Gamma = (N, (X_i)_{i \in N}, (u_i)_{i \in N})$ has the *finite improvement property (FIP)* if every improvement path in the transition graph $G(\Gamma)$ is finite.

Consider the execution of IMPROVING MOVES. The algorithm computes an improving path $P = (x^0, x^1, \dots)$ with starting configuration x^0 and is guaranteed to terminate if Γ has the FIP. That is, Γ admits a pure Nash equilibrium if it has the FIP. In order to characterize games that have the FIP, we introduce *potential games*.

Definition 1.5. A finite strategic game $\Gamma = (N, (X_i)_{i \in N}, (u_i)_{i \in N})$ is called *exact potential game* if there exists a function (also called *potential function*) $\Phi : X \to \mathbb{R}$ such that for every player $i \in N$ and for every $x_{-i} \in X_{-i}$ and $x_i, y_i \in X_i$:

$$u_i(x_{-i}, y_i) - u_i(x_{-i}, x_i) = \Phi(x_{-i}, x_i) - \Phi(x_{-i}, y_i).$$

Γ is an *ordinal potential game* if for every player $i \in N$ and for every $x_{-i} \in X_{-i}$ and $x_i, y_i \in X_i$:

$$u_i(x_{-i}, y_i) - u_i(x_{-i}, x_i) > 0 \quad \Leftrightarrow \quad \Phi(x_{-i}, x_i) - \Phi(x_{-i}, y_i) > 0.$$

Γ is a *generalized ordinal potential game* if for every player $i \in N$ and for every $x_{-i} \in X_{-i}$ and $x_i, y_i \in X_i$:

$$u_i(x_{-i}, y_i) - u_i(x_{-i}, x_i) > 0 \quad \Rightarrow \quad \Phi(x_{-i}, x_i) - \Phi(x_{-i}, y_i) > 0.$$

1.3 Existence of Pure Nash Equilibria

Theorem 1.2. *Let $\Gamma = (N, (X_i)_{i \in N}, (u_i)_{i \in N})$ be an ordinal potential game. The set of pure Nash equilibria of Γ coincides with the set of local minima of Φ, i.e., x is a Nash equilibrium of Γ iff*

$$\forall i \in N,\ \forall y_i \in X_i: \qquad \Phi(x) \leq \Phi(x_{-i}, y_i).$$

Proof. The proof follows directly from the definition of ordinal potential games. □

Theorem 1.3. *Every generalized ordinal potential game Γ has the FIP. In particular, Γ admits a pure Nash equilibrium.*

Proof. Consider an improvement path $P = (x^0, x^1, \dots)$ in the transition graph $G(\Gamma)$. Since Γ is a generalized ordinal potential game, we have

$$\Phi(x^0) > \Phi(x^1) > \dots$$

Because the transition graph has a finite number of nodes, the path P must be finite. Thus, Γ has the FIP. The existence follows now directly from the FIP and the IMPROVING MOVES algorithm. □

One can show the following equivalence (we omit the proof here).

Theorem 1.4. *Let Γ be a finite strategic game. Γ has the FIP if and only if Γ admits a generalized ordinal potential function.*

1.4 Price of Anarchy and Price of Stability

Consider an instance $\Gamma = (N, (X_i)_{i \in N}, (u_i)_{i \in N})$ of a potential game and suppose we are given a social cost function $c : X \to \mathbb{R}$ that maps every strategy profile $x \in X$ to some cost $c(x)$. We assume that the global objective is to minimize $c(x)$ over all $x \in X$. (The definitions are similar if we want to maximize $c(x)$.) Let $\mathsf{opt}(\Gamma)$ refer to the minimum cost of a strategy profile $x \in X$ and let $\mathsf{NE}(\Gamma)$ refer to the set of strategy profiles that are Nash equilibria of Γ.

The *price of stability* is defined as the worst case ratio over all instances of the game of the cost of a best Nash equilibrium over the optimal cost; more formally,

$$\mathsf{POS} := \max_{\Gamma} \min_{x \in \mathsf{NE}(\Gamma)} \frac{c(x)}{\mathsf{opt}(\Gamma)}.$$

In contrast, the *price of anarchy* is defined as the worst case ratio over all instances of the game of the cost of a worst Nash equilibrium over the optimal cost; more formally,

$$\mathsf{POA} := \max_{\Gamma} \max_{x \in \mathsf{NE}(\Gamma)} \frac{c(x)}{\mathsf{opt}(\Gamma)}.$$

Theorem 1.5. *Consider a potential game $\Gamma = (N, (X_i)_{i \in N}, (u_i)_{i \in N})$ with potential function Φ. Let $c : X \to \mathbb{R}_+$ be a social cost function. If Φ satisfies for every $x \in X$:*

$$\frac{1}{\alpha} c(x) \leq \Phi(x) \leq \beta c(x)$$

for some $\alpha, \beta > 0$, then the price of stability is at most $\alpha\beta$.

Proof. Let x be a strategy profile that minimizes Φ. Then x is a Nash equilibrium by Theorem 1.2. Let x^* be an optimal solution of cost $\mathsf{opt}(\Gamma)$. Note that

$$\Phi(x) \leq \Phi(x^*) \leq \beta c(x^*) = \beta \mathsf{opt}(\Gamma).$$

Moreover, we have $c(x) \leq \alpha \Phi(x)$, which concludes the proof. □

2 Selfish Routing

We consider network routing problems in which users choose their routes so as to minimize their own travel time. Our main focus will be to study the inefficiency of Nash equilibria and to identify effective means to decrease the inefficiency caused by selfish behavior.

2.1 Motivating Example: Pigou Example

We first consider an example:

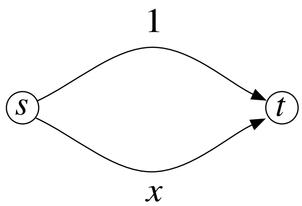

Figure 2: Pigou instance

Example 2.1 (Pigou's example). Consider the parallel-arc network in Figure 2. For every arc a, we have a latency function $\ell_a : \mathbb{R}_+ \to \mathbb{R}_+$, representing the load-dependent travel time or latency for traversing this arc. In the above example, we have for the upper arc $\ell_a(x) = 1$, i.e., the latency is one independently of the amount of flow on that arc. The lower arc has latency function $\ell_a(x) = x$, i.e., the latency grows linearly with the amount of flow on that arc. Suppose we want to send one unit of flow from s to t and that this one unit of flow corresponds to infinitely many users that want to travel from s to t.

Every selfish user will reason as follows: The latency of the upper arc is one (independently of the flow) while the latency of the lower arc is at most one (and even strictly less than one if some users are not using this arc). Thus, every user chooses the lower arc. The resulting flow is a Nash flow. Since every user experiences a latency of one, the total average latency of this Nash flow is one.

We next compute an optimal flow that minimizes the total average latency of the users. Assume we send $p \in [0,1]$ units of flow along the lower arc and $1-p$ units of flow along the upper arc. The total average latency is $(1-p)\cdot 1 + p \cdot p = 1 - p + p^2$. This function is minimized for $p = \frac{1}{2}$. Thus, the optimal flow sends one-half units of flow along the upper and one-half units of flow along the lower arc. Its total average latency is $\frac{3}{4}$.

This example shows that selfish user behavior may lead to outcomes that are inefficient: The resulting Nash flow is suboptimal with a total average latency that is $\frac{4}{3}$ times larger than the total average latency of an optimal flow. This raises the following natural questions: How large can this inefficiency ratio be in general networks? Does it depend on the topology of the network?

2.2 Model

We formalize the setting introduced above. An instance of a *selfish routing game* is given as follows:

- directed graph $G = (V, A)$ with vertex set V and arc set A;
- nondecreasing and continuous latency function $\ell_a : \mathbb{R}_+ \to \mathbb{R}_+$ for every arc $a \in A$.
- set of k commodities $[k] := \{1, \dots, k\}$, specifying for each commodity $i \in [k]$ a source vertex s_i and a target vertex t_i;
- for each commodity $i \in [k]$, a demand $r_i > 0$ that represents the amount of flow that has to be sent from s_i to t_i;

We use (G, r, ℓ) to refer to an instance for short.

Let $\mathscr{P}_i$ be the set of all simple paths from s_i to t_i in G and let $\mathscr{P} := \cup_i \mathscr{P}_i$. A *flow* is a function $f : \mathscr{P} \to \mathbb{R}_+$. The flow f is *feasible* (with respect to r) if for all $i \in [k]$, $\sum_{P \in \mathscr{P}_i} f_P = r_i$, i.e., the total flow send from s_i to t_i meets the demand r_i. For a given flow f, we define the aggregated flow on arc $a \in A$ as $f_a := \sum_{P \in \mathscr{P} : a \in P} f_P$.

The *total travel time* of a path $P \in \mathscr{P}$ with respect to f is defined as the sum of the latencies of the arcs on that path:

$$\ell_P(f) := \sum_{a \in P} \ell_a(f_a).$$

We assess the overall quality of a given flow f by means of a global cost function C. Though there are potentially many different cost functions that one may want to consider (depending on the application), we focus on the *total average latency* as cost function here.

Definition 2.1. The *total cost* of a flow f is defined as:

$$C(f) := \sum_{P \in \mathscr{P}} \ell_P(f) f_P. \tag{1}$$

Note that the total cost can equivalently be expressed as the sum of the average latencies on the arcs:

$$C(f) = \sum_{P \in \mathscr{P}} \ell_P(f) f_P = \sum_{P \in \mathscr{P}} \Big(\sum_{a \in P} \ell_a(f_a) \Big) f_P = \sum_{a \in A} \Big(\sum_{P \in \mathscr{P} : a \in P} f_P \Big) \ell_a(f_a) = \sum_{a \in A} \ell_a(f_a) f_a.$$

2.3 Nash Flows and their Existence

The basic viewpoint that we adopt here is that players act selfishly in that they attempt to minimize their own individual travel time. A standard solution concept to predict outcomes of selfish behavior is the one of an equilibrium outcome in which no player has an incentive to unilaterally deviate from its current strategy. In the context of nonatomic selfish routing games, this viewpoint translates to the following definition:

Definition 2.2. A feasible flow f for the instance (G,r,ℓ) is a *Nash flow* if for every commodity $i \in [k]$ and two paths P, $Q \in \mathscr{P}_i$ with $f_P > 0$ and for every $\delta \in (0, f_P]$, we have $\ell_P(f) \le \ell_Q(\tilde{f})$, where

$$\tilde{f}_P := \begin{cases} f_P - \delta & \text{if } P = P \\ f_P + \delta & \text{if } P = Q \\ f_P & \text{otherwise.} \end{cases}$$

Intuitively, the above definition states that for every commodity $i \in [k]$, shifting $\delta \in (0, f_P]$ units of flow from a flow carrying path $P \in \mathscr{P}_i$ to an arbitrary path $Q \in \mathscr{P}_i$ does not lead to a smaller latency.

A similar concept was introduced by Wardrop (1952) in his first principle: A flow for the nonatomic selfish routing game is a *Wardrop equilibrium* if for every source-target pair the latencies of the used routes are less than or equal to those of the unused routes.

Definition 2.3. A feasible flow f for the instance (G,r,ℓ) is a *Wardrop equilibrium* (or *Wardrop flow*) if

$$\forall i \in [k],\ \forall P,\ Q \in \mathscr{P}_i,\ f_P > 0: \quad \ell_P(f) \le \ell_Q(f). \tag{2}$$

For $\delta \to 0$ the definition of a Nash flow corresponds to the one of a Wardrop flow. Subsequently, we use the Wardrop flow definition; we slightly abuse naming here and will also refer to such flows as Nash flows.

Corollary 2.1. *Let f be a Nash flow for (G,r,ℓ) and define for every $i \in [k]$, $c_i(f) := \min_{P \in \mathscr{P}_i} \ell_P(f)$. Then $\ell_P(f) = c_i(f)$ for every $P \in \mathscr{P}_i$ with $f_P > 0$.*

Proof. By the definition of $c_i(f)$, we have that for every $P \in \mathscr{P}_i$: $\ell_P(f) \ge c_i(f)$. Using (2), we conclude that for every $P \in \mathscr{P}_i$ with $f_P > 0$: $\ell_P(f) \le c_i(f)$. □

Note that the above corollary states that for each commodity all flow carrying paths have the same latency and all other paths cannot have a smaller latency. The flow carrying paths are thus shortest paths with respect to the total latency.

We next argue that Nash flows always exist and that their cost is unique. In order to do so, we use a powerful result from convex optimization. Consider the following program (CP):

$$\begin{aligned} \min \quad & \sum_{a \in A} h_a(f_a) \\ \text{s.t.} \quad & \sum_{P \in \mathscr{P}_i} f_P = r_i && \forall i \in [k] \\ & f_a = \sum_{P \in \mathscr{P}: a \in P} f_P && \forall a \in A \\ & f_P \ge 0 && \forall P \in \mathscr{P}. \end{aligned}$$

Note that the set of all feasible solutions for (CP) corresponds exactly to the set of all flows that are feasible for our selfish routing instance (G, r, ℓ). The above program is a linear program if the functions $(h_a)_{a\in A}$ are linear. (CP) is a convex program if the functions $(h_a)_{a\in A}$ are convex. A convex program can be solved efficiently by using, e.g., the ellipsoid method. The following is a fundamental theorem in convex (or, more generally, non-linear) optimization:

Theorem 2.1 (Karush–Kuhn–Tucker (KKT) Optimality Conditions)**.** *Consider the program (CP) with continuously differentiable and convex functions $(h_a)_{a\in A}$. A feasible flow f is an optimal solution for (CP) if and only if*

$$\forall i \in [k],\ \forall P,\ Q \in \mathscr{P}_i,\ f_P > 0: \quad h'_P(f) := \sum_{a\in P} h'_a(f_a) \leq \sum_{a\in Q} h'_a(f_a) =: h'_Q(f), \tag{3}$$

where $h'_a(x)$ refers to the first derivative of $h_a(x)$.

Observe that (3) is very similar to the Wardrop equilibrium conditions (2). In fact, these two conditions coincide if we define for every $a \in A$:

$$h_a(f_a) := \int_0^{f_a} \ell_a(x)dx. \tag{4}$$

Corollary 2.2. *Let (G, r, ℓ) be a selfish routing instance with nondecreasing and continuous latency functions $(\ell_a)_{a\in A}$. A feasible flow f is a Nash flow if and only if it is an optimal solution to (CP) with functions $(h_a)_{a\in A}$ as defined in (4).*

Proof. For every arc $a \in A$, the function h_a is convex (since ℓ_a is nondecreasing) and continuously differentiable (since ℓ_a is continuous). The proof now follows from Theorem 2.1. □

We will also need the following theorem:

Theorem 2.2 (Extreme Value Theorem)**.** *Let X be a compact set and $f : X \to \mathbb{R}$ a continuous function. Then f attains both a maximum and a minimum on X.*

Corollary 2.3. *Let (G, r, ℓ) be a selfish routing instance with nondecreasing and continuous latency functions $(\ell_a)_{a\in A}$. Then a Nash flow f always exists. Moreover, its cost $C(f)$ is unique.*

Proof. The set of all feasible flows for (CP) is compact (closed and bounded). Moreover, the objective function of (CP) with (4) is continuous (since ℓ_a is continuous for every $a \in A$). Thus, the minimum of (CP) must exist (by the Extreme Value Theorem). Since the objective function of (CP) is convex, the optimal value of (CP) is unique. It is not hard to conclude that the cost $C(f)$ of a Nash flow is unique. □

Note that, in particular, the above observations imply that we can compute a Nash flow for a given nonatomic selfish routing instance (G, r, ℓ) efficiently by solving the convex program (CP) with (4).

2.4 Optimal Flows

We define an optimal flow as follows:

Definition 2.4. A feasible flow f^* for the instance (G,r,ℓ) is an *optimal flow* if $C(f^*) \leq C(x)$ for every feasible flow x.

The set of optimal flows corresponds to the set of all optimal solutions to (CP) if we define for every arc $a \in A$:

$$h_a(f_a) := \ell_a(f_a) f_a. \tag{5}$$

Since the cost function C is continuous (because ℓ_a is continuous for every $a \in A$), we conclude that an optimal flow always exists (again using the Extreme Value Theorem). Moreover, we will assume that h_a is convex and continuously differentiable for each arc $a \in A$; latency functions $(\ell_a)_{a\in A}$ that satisfy these conditions are called *standard*. Using Theorem 2.1, we obtain the following characterization of optimal flows:

Corollary 2.4. *Let the latency functions $(\ell_a)_{a\in A}$ be standard. A feasible flow f^* for the instance (G,r,ℓ) is an optimal flow if and only if:*

$$\forall i \in [k],\ \forall P,\ Q \in \mathscr{P}_i,\ f_P^* > 0: \quad \sum_{a\in P} \ell_a(f_a^*) + \ell_a'(f_a^*) f_a^* \leq \sum_{a\in Q} \ell_a(f_a^*) + \ell_a'(f_a^*) f_a^*.$$

That is, an optimal flow is a Nash flow with respect to so-called *marginal latency functions* $(\ell_a^*)_{a\in A}$, which are defined as

$$\ell_a^*(x) := \ell_a(x) + \ell_a'(x)x.$$

2.5 Price of Anarchy

We study the inefficiency of Nash flows in comparison to an optimal flow. A common measure of the inefficiency of equilibrium outcomes is the *price of anarchy*.

Definition 2.5. Let (G,r,ℓ) be an instance of the selfish routing game and let f and f^* be a Nash flow and an optimal flow, respectively. The *price of anarchy* $\rho(G,r,\ell)$ of the instance (G,r,ℓ) is defined as:

$$\rho(G,r,\ell) = \frac{C(f)}{C(f^*)}. \tag{6}$$

(Note that (6) is well-defined since the cost of Nash flows is unique.) The price of anarchy of a set of instances $\mathscr{I}$ is defined as

$$\rho(\mathscr{I}) = \sup_{(G,r,\ell)\in\mathscr{I}} \rho(G,r,\ell).$$

2.6 Upper Bounds on the Price of Anarchy

Subsequently, we derive upper bounds on the price of anarchy for selfish routing games. The following variational inequality will turn out to be very useful.

Lemma 2.1 (Variational inequality)**.** *A feasible flow f for the instance (G,r,ℓ) is a Nash flow if and only if it satisfies that for every feasible flow x:*

$$\sum_{a\in A} \ell_a(f_a)(f_a - x_a) \le 0. \tag{7}$$

Proof. Given a flow f satisfying (7), we first show that condition (2) of Definition 2.3 holds. Let $P,\ Q \in \mathscr{P}_i$ be two paths for some commodity $i \in [k]$ such that $\delta := f_P > 0$. Define a flow x as follows:

$$x_a := \begin{cases} f_a & \text{if } a \in P \cap Q \text{ or } a \notin P \cup Q \\ f_a - \delta & \text{if } a \in P \\ f_a + \delta & \text{if } a \in Q. \end{cases}$$

By construction x is feasible. Hence, from (7) we obtain:

$$\sum_{a\in A} \ell_a(f_a)(f_a - x_a) = \sum_{a\in P} \ell_a(f_a)(f_a - (f_a - \delta)) + \sum_{a\in Q} \ell_a(f_a)(f_a - (f_a + \delta)) \le 0.$$

We divide the inequality by $\delta > 0$, which yields the Wardrop conditions (2).

Now assume that f is a Nash flow. By Corollary 2.1, we have for every $i \in [k]$ and $P \in \mathscr{P}_i$ with $f_P > 0$: $\ell_P(f) = c_i(f)$. Furthermore, for $Q \in \mathscr{P}_i$ with $f_Q = 0$, we have $\ell_Q(f) \ge c_i(f)$. It follows that for every feasible flow x:

$$\begin{aligned} \sum_{a\in A} \ell_a(f_a) f_a &= \sum_{i\in[k]} \sum_{P\in\mathscr{P}_i} c_i(f) f_P = \sum_{i\in[k]} c_i(f) \left(\sum_{P\in\mathscr{P}_i} f_P \right) = \sum_{i\in[k]} c_i(f) \left(\sum_{P\in\mathscr{P}_i} x_P \right) \\ &= \sum_{i\in[k]} \sum_{P\in\mathscr{P}_i} c_i(f) x_P \le \sum_{i\in[k]} \sum_{P\in\mathscr{P}_i} \ell_P(f) x_P = \sum_{a\in A} \ell_a(f_a) x_a. \end{aligned}$$

□

We derive an upper bound on the price of anarchy for affine linear latency functions with nonnegative coefficients:

$$\mathscr{L}_1 := \{ g : \mathbb{R}_+ \to \mathbb{R}_+ \ :\ g(x) = q_1 x + q_0 \text{ with } q_0, q_1 \in \mathbb{R}_+ \}.$$

Theorem 2.3. *Let (G,r,ℓ) be an instance of a nonatomic routing game with affine linear latency functions $(\ell_a)_{a\in A} \in \mathscr{L}_1^A$. The price of anarchy $\rho(G,r,\ell)$ is at most $\frac{4}{3}$.*

Proof. Let f be a Nash flow and let x be an arbitrary feasible flow for (G,r,ℓ). Using

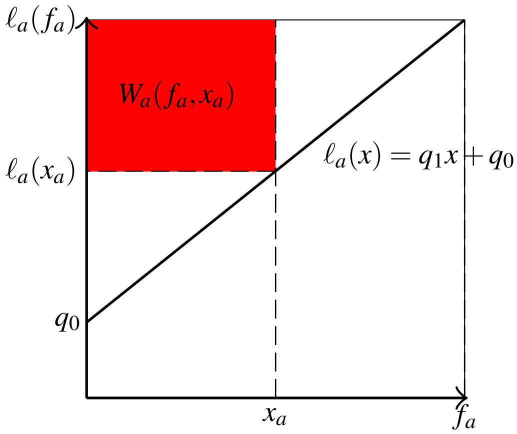

Figure 3: Illustration of the worst case ratio of $W_a(f_a, x_a)$ and $\ell_a(f_a)\, f_a$.

the variational inequality (7), we obtain

$$\begin{aligned} C(f) &= \sum_{a\in A} \ell_a(f_a) f_a \leq \sum_{a\in A} \ell_a(f_a) x_a = \sum_{a\in A} \ell_a(f_a) x_a + \ell_a(x_a) x_a - \ell_a(x_a) x_a \\ &= \sum_{a\in A} \ell_a(x_a) x_a + \underbrace{[\ell_a(f_a) - \ell_a(x_a)] x_a}_{=:W_a(f_a,x_a)} = \sum_{a\in A} \ell_a(x_a) x_a + \sum_{a\in A} W_a(f_a, x_a). \end{aligned}$$

We next bound the function $W_a(f_a, x_a)$ in terms of $\omega \cdot \ell_a(f_a) f_a$ for some $0 \leq \omega < 1$, where

$$\omega := \max_{f_a, x_a \geq 0} \frac{(\ell_a(f_a) - \ell_a(x_a)) x_a}{\ell_a(f_a) f_a} = \max_{f_a, x_a \geq 0} \frac{W_a(f_a, x_a)}{\ell_a(f_a) f_a}.$$

Note that for $x_a \geq f_a$ we have $\omega \leq 0$ (because latency functions are non-decreasing). Hence, we can assume $x_a \leq f_a$. See Figure 3 for a geometric interpretation. Since latency functions are affine linear, ω is upper bounded by $\frac{1}{4}$. We obtain

$$C(f) \leq C(x) + \sum_{a\in A} \frac{1}{4} \ell_a(f_a) f_a = C(x) + \frac{1}{4} C(f).$$

Rearranging terms and letting x be an optimal flow concludes the proof. □

We can extend the above proof to more general classes of latency functions. For the latency function ℓ_a of an arc $a \in A$, define

$$\omega(\ell_a) := \sup_{f_a, x_a \geq 0} \frac{(\ell_a(f_a) - \ell_a(x_a)) x_a}{\ell_a(f_a)\, f_a}. \tag{8}$$

We assume by convention $0/0 = 0$. See Figure 4 for a graphical illustration of this value. For a given class $\mathscr{L}$ of non-decreasing latency functions, we define

$$\omega(\mathscr{L}) := \sup_{\ell_a \in \mathscr{L}} \omega(\ell_a).$$

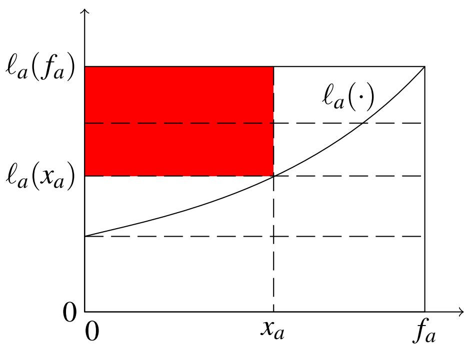

Figure 4: Illustration of $\omega(\ell_a)$.

Theorem 2.4. *Let (G,r,ℓ) be an instance of the nonatomic selfish routing game with latency functions $(\ell_a)_{a\in A} \in \mathscr{L}^A$. Let $0 \le \omega(\mathscr{L}) < 1$ be defined as above. The price of anarchy $\rho(G,r,\ell)$ is at most $(1-\omega(\mathscr{L}))^{-1}$.*

Proof. Let f be a Nash flow and let x be an arbitrary feasible flow. We have

$$\begin{aligned} C(f) &= \sum_{a\in A} \ell_a(f_a) f_a \le \sum_{a\in A} \ell_a(f_a) x_a = \sum_{a\in A} \ell_a(f_a) x_a + \ell_a(x_a)x_a - \ell_a(x_a)x_a \\ &= \sum_{a\in A} \ell_a(x_a)x_a + [\ell_a(f_a) - \ell_a(x_a)]x_a \le C(x) + \omega(\mathscr{L})C(f). \end{aligned}$$

Here, the first inequality follows from the variational inequality (7). The last inequality follows from the definition of $\omega(\mathscr{L})$. Since $\omega(\mathscr{L}) < 1$, the claim follows. □

In general, we define $\mathscr{L}_d$ as the set of latency functions $g : \mathbb{R}_+ \to \mathbb{R}_+$ that satisfy

$$g(\mu x) \ge \mu^d g(x) \quad \forall \mu \in [0,1].$$

Note that $\mathscr{L}_d$ contains polynomial latency functions with nonnegative coefficients and degree at most d.

Lemma 2.2. *Consider latency functions in $\mathscr{L}_d$. Then*

$$\omega(\mathscr{L}_d) \le \frac{d}{(d+1)^{(d+1)/d}}.$$

Proof. Recall the definition of $\omega(\ell_a)$:

$$\omega(\ell_a) = \sup_{f_a, x_a \ge 0} \frac{\big(\ell_a(f_a) - \ell_a(x_a)\big)x_a}{\ell_a(f_a) f_a}. \tag{9}$$

We can assume that $x_a \le f_a$ since otherwise $\omega(\ell_a) \le 0$. Let $\mu := \frac{x_a}{f_a} \in [0,1]$. Then

$$\omega(\ell_a) = \max_{\mu\in[0,1], f_a \ge 0} \left(\frac{\big(\ell_a(f_a) - \ell_a(\mu f_a)\big)\mu f_a}{\ell_a(f_a) f_a} \right) \le \max_{\mu\in[0,1], f_a \ge 0} \left(\frac{\big(\ell_a(f_a) - \mu^d \ell_a(f_a)\big)\mu f_a}{\ell_a(f_a) f_a} \right)$$

d	1	2	3	...
$\rho(G,r,\ell)$	≈ 1.333	≈ 1.626	≈ 1.896	

Table 1: The price of anarchy for polynomial latency functions of degree d.

$$= \max_{\mu \in [0,1]} (1-\mu^d)\mu. \tag{10}$$

Here, the first inequality holds since $\ell_a \in \mathscr{L}_d$. Since this is a strictly convex program, the unique global optimum is given by

$$\mu^* = \left(\frac{1}{d+1}\right)^{\frac{1}{d}}.$$

Replacing μ^* in (10) yields the claim. □

Theorem 2.5. *Let (G,r,ℓ) be an instance of a nonatomic routing game with latency functions $(\ell_a)_{a\in A} \in \mathscr{L}_d^A$. The price of anarchy $\rho(G,r,\ell)$ is at most*

$$\rho(G,r,\ell) \leq \left(1 - \frac{d}{(d+1)^{(d+1)/d}}\right)^{-1}.$$

Proof. The theorem follows immediately from Theorem 2.4 and Lemma 2.2. □

The price of anarchy for polynomial latency functions with nonnegative coefficients and degree d is given in Table 1 for small values of d.

2.7 Lower Bounds on the Price of Anarchy

We can show that the bound that we have derived in the previous section is actually tight.

Theorem 2.6. *Consider nonatomic selfish routing games with latency functions in $\mathscr{L}_d$. There exist instances such that the price of anarchy is at least*

$$\left(1 - \frac{d}{(d+1)^{(d+1)/d}}\right)^{-1}.$$

Proof. See assignments. □

2.8 Motivating Example: Braess's Paradox

Example 2.2 (Braess's paradox)**.** Consider the network in Figure 5 (left). Assume that we want to send one unit of flow from s to t. It is not hard to verify that the Nash flow splits evenly and sends one-half units of flow along the upper and lower arc, respectively. This flow is also optimal having a total average latency of $\frac{3}{2}$.

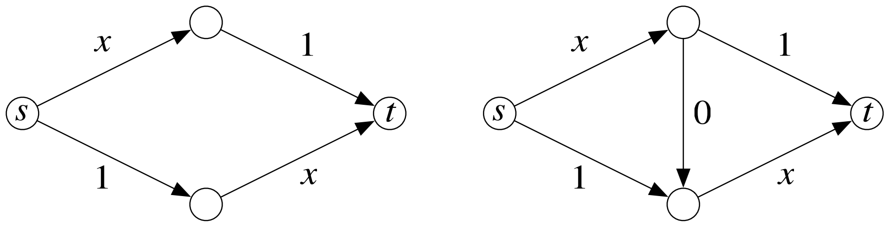

Figure 5: Braess Paradox

Now, suppose there is a global authority that wants to improve the overall traffic situation by building new roads. The network in Figure 5 (right) depicts an augmented network where an additional arc with constant latency zero has been introduced. How do selfish users react to this change? What happens is that every user chooses the zig-zag path, first traversing the upper left arc, then the newly introduced zero latency arc and then the lower right arc. The resulting Nash flow has a total average latency of 2.

The Braess Paradox shows that extending the network infrastructure does not necessarily lead to an improvement with respect to the total average latency if users choose their routes selfishly. In the above case, the total average latency degrades by a factor of $\frac{4}{3}$. In general, one may ask the following questions: How large can this degradation be? Can we develop efficient methods to detect such phenomena?

2.9 Detecting Braess's Paradox

Suppose we are given a single-commodity instance (G,r,ℓ) of the nonatomic selfish routing game. Let f be a Nash flow for (G,r,ℓ) and define $d(G,r,\ell) := c_1(f)$ as the common latency of all flow-carrying paths (see Corollary 2.1). We study the following optimization problem: Given (G,r,ℓ), find a subgraph $H \subseteq G$ that minimizes $d(H,r,\ell)$. We call this problem the NETWORK DESIGN problem.

Corollary 2.5. *Let (G,r,ℓ) be a single-commodity instance of the nonatomic selfish routing game with linear latency functions. Then for every subgragph $H \subseteq G$:*

$$d(G,r,\ell) \leq \frac{4}{3} d(H,r,\ell).$$

Proof. Let h and f be the Nash flows for the instances (H,r,ℓ) and (G,r,ℓ), respectively. By Corollary 2.1, the latency of every flow-carrying path in a Nash flow is equal. Thus, the costs of the Nash flows f and h, respectively, are $rd(G,r,\ell)$ and $rd(H,r,\ell)$. Using that h is a feasible flow for (G,r,ℓ) and the upper bound of $4/3$ on the price of anarchy for linear latencies, we obtain

$$C(f) = rd(G,r,\ell) \leq \frac{4}{3} C(h) = \frac{4}{3} rd(H,r,\ell).$$

□

We can generalize the above proof to obtain:

Corollary 2.6. *Let (G,r,ℓ) be a single-commodity instance of the nonatomic selfish routing game with polynomial latency functions in $\mathscr{L}_d$. Then for every subgraph $H \subseteq G$:*

$$d(G,r,\ell) \leq \left(1 - \frac{d}{(d+1)^{(d+1)/d}}\right)^{-1} d(H,r,\ell).$$

We next turn to designing approximation algorithms that compute a "good" subgraph H of G with a provable approximation guarantee. We review some basics from computational theory first. Readers that are familiar with this topic can continue with Section 2.11.

2.10 Mini-introduction: Computational Complexity

We briefly review some basics from complexity theory. The exposition here is kept at a rather high-level; the interested reader is referred to, e.g., the book *Computers and Intractability: A Guide to the Theory of NP-Completeness* by Garey and Johnson for more details.

Definition 2.6 (Optimization problem)**.** A cost minimization problem $\mathscr{P} = (\mathscr{I}, \mathscr{S}, c)$ is given by:

- a set of instances $\mathscr{I}$ of $\mathscr{P}$;
- for every instance $I \in \mathscr{I}$ a set of feasible solutions $\mathscr{S}_I$;
- for every feasible solution $S \in \mathscr{S}_I$ a real-valued cost $c(S)$.

The goal is to compute for a given instance $I \in \mathscr{I}$ a solution $S \in \mathscr{S}_I$ that minimizes $c(S)$. We use $\mathtt{opt}_I$ to refer to the cost of an optimal solution for I.

Definition 2.7 (Decision problem)**.** A decision problem $\mathscr{P} = (\mathscr{I}, \mathscr{S}, c, k)$ is given by:

- a set of instances $\mathscr{I}$ of $\mathscr{P}$;
- for every instance $I \in \mathscr{I}$ a set of feasible solutions $\mathscr{S}_I$;
- for every feasible solution $S \in \mathscr{S}_I$ a real-valued cost $c(S)$.

The goal is to decide whether for a given instance $I \in \mathscr{I}$ a solution $S \in \mathscr{S}_I$ exists such that the cost $c(S)$ of S is at most k. If there exists such a solution, we say that I is a *yes-instance*; otherwise, I is a *no-instance*.

Example 2.3 (Traveling salesman problem)**.** We are given an undirected graph $G = (V,E)$ with edge costs $c : E \to \mathbb{R}_+$. The *traveling salesman problem (TSP)* asks for the computation of a tour that visits every vertex exactly once and has minimum total cost. The decision problem asks for the computation of a tour of cost at most k.

Several optimization problems (and their respective decision problems) are hard in the sense that there are no polynomial-time algorithms known that solve the problem exactly. Here *polynomial-time algorithm* refers to an algorithm whose running time can be bound by a polynomial function in the *size* of the input instance. For example, an algorithm has polynomial running time if for every input of size n its running time is bound by n^k for some constant k. There are different ways to encode an input instance. Subsequently, we assume that the input is encoded in binary and the *size* of the input instance refers to the number of bits that one needs to represent the instance.

Definition 2.8 (Complexity class P)**.** A decision problem $\mathscr{P} = (\mathscr{I}, \mathscr{S}, c, k)$ belongs to the complexity class P (which stands for *polynomial time*) if for every instance $I \in \mathscr{I}$ one can find in polynomial time a feasible solution $S \in \mathscr{S}_I$ whose cost is at most k, or determine that no such solution exists.

Definition 2.9 (Complexity class NP)**.** A decision problem $\mathscr{P} = (\mathscr{I}, \mathscr{S}, c, k)$ belongs to the complexity class NP (which stands for *non-deterministic polynomial time*) if for every yes-instance $I \in \mathscr{I}$ there exists a solution S whose validity can be verified in polynomial time, i.e., whether $S \in\in \mathscr{S}_I$ and $c(S) \leq k$.

Clearly, $P \subseteq NP$. The question whether $P \neq NP$ is still unresolved and one of the biggest open questions to date.[1]

Definition 2.10 (Polynomial time reduction)**.** A decision problem $\mathscr{P}_1 = (\mathscr{I}_1, \mathscr{S}_1, c_1, k_1)$ is *polynomial time reducible* to a decision problem $\mathscr{P}_2 = (\mathscr{I}_2, \mathscr{S}_2, c_2, k_2)$ if every instance $I_1 \in \mathscr{I}_1$ of $\mathscr{P}_1$ can in polynomial time be mapped to an instance $I_2 \in \mathscr{I}_2$ of $\mathscr{P}_2$ such that: I_1 is a yes-instance of $\mathscr{P}_1$ if and only if I_2 is a yes-instance of $\mathscr{P}_2$.

Definition 2.11 (NP-completeness)**.** A problem $\mathscr{P} = (\mathscr{I}, \mathscr{S}, c, k)$ is *NP-complete* if

- $\mathscr{P}$ belongs to NP;
- every problem in NP is polynomial time reducible to $\mathscr{P}$.

Essentially, problems that are NP-complete are polynomial time equivalent: If we are able to solve one of these problems in polynomial time then we are able to solve all of them in polynomial time. Note that in order to show that a problem is NP-complete, it is sufficient to show that it is in NP and that an NP-complete problem is polynomial time reducible to this problem.

Example 2.4 (TSP)**.** The problem of deciding whether a traveling salesman tour of cost at most k exists is NP-complete.

Many fundamental problems are NP-complete and it is therefore unlikely (though not impossible) that efficient algorithms for solving these problems in polynomial time exist. One therefore often considers approximation algorithms:

[1] See also *The Millennium Prize Problems* at http://www.claymath.org/millennium.

Definition 2.12 (Approximation algorithm)**.** An algorithm ALG for a cost minimization problem $\mathscr{P} = (\mathscr{I}, \mathscr{S}, c)$ is called an *α-approximation algorithm* for some $\alpha \geq 1$ if for every given input instance $I \in \mathscr{I}$ of $\mathscr{P}$:

1. ALG computes in polynomial time a feasible solution $S \in \mathscr{S}_I$, and
2. the cost of S is at most α times larger than the optimal cost, i.e., $c(S) \leq \alpha \mathsf{opt}_I$.

α is also called the *approximation factor* or *approximation guarantee* of ALG.

2.11 Detecting Braess's Paradox — Continued

A trivial approximation algorithm (called TRIVIAL subsequently) for the NETWORK DESIGN problem is to simply return the original graph as a solution. Using the above corollaries, it follows that TRIVIAL has an approximation guarantee of

$$\left(1 - \frac{d}{(d+1)^{(d+1)/d}}\right)^{-1}$$

for latency functions in $\mathscr{L}_d$.

We will show that the performance guarantee of TRIVIAL is best possible, unless $P = NP$.

Theorem 2.7. *Assuming $P \neq NP$, for every $\varepsilon > 0$ there is no $(\frac{4}{3} - \varepsilon)$-approximation algorithm for the* NETWORK DESIGN *problem.*

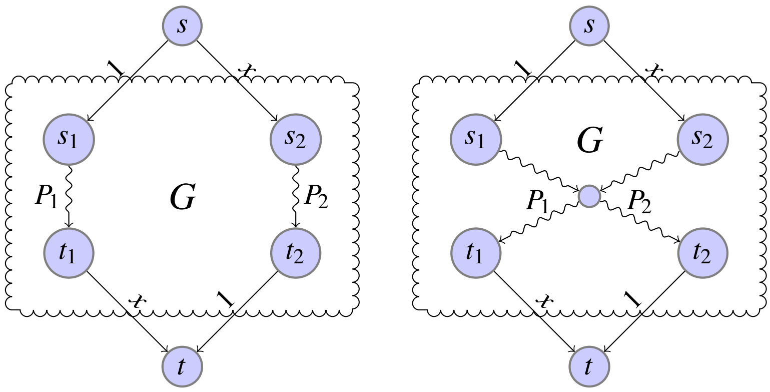

Figure 6: (a) Yes-instance of 2DDP. (b) No-instance of 2DDP.

Proof. We reduce from the *2-directed vertex-disjoint paths problem (*2DDP*)*, which is NP-complete. An instance of this problem is given by a directed graph $G = (V, A)$ and two vertex pairs (s_1, t_1), (s_2, t_2). The question is whether there exist a path P_1 from s_1 to t_1 and a path P_2 from s_2 to t_2 in G such that P_1 and P_2 are vertex disjoint. We will show that a $(\frac{4}{3} - \varepsilon)$-approximation algorithm could be used to differentiate between yes- and no-instances of 2DDP in polynomial time.

Suppose we are given an instance $\mathscr{I}$ of 2DDP. We construct a graph G' by adding a super source s and a super sink t to the network. We connect s to s_1 and s_2 and t_1 and t_2 to t, respectively. The latency functions of the added arcs are given as indicated in Figure 2.11, where we assume that all latency functions in the original graph G are set to zero. This can be done in polynomial time.

We will prove the following two statements:

(i) If $\mathscr{I}$ is a yes-instance of 2DDP then $d(H,1,\ell) = 3/2$ for some subgraph $H \subseteq G'$.

(ii) If $\mathscr{I}$ is a no-instance of 2DDP then $d(H,1,\ell) \geq 2$ for every subgraph $H \subseteq G'$.

Suppose for the sake of a contradiction that a $(\frac{4}{3} - \varepsilon)$-approximation algorithm ALG for the NETWORK DESIGN problem exists. ALG then computes in polynomial time a subnetwork $H \subseteq G'$ such that the cost of a Nash flow in H is at most $(\frac{4}{3} - \varepsilon)\mathsf{opt}$, where $\mathsf{opt} = \min_{H \subseteq G'} d(H,r,\ell)$. That is, the cost of a Nash flow for the subnetwork H computed by ALG is less than 2 for instances in (i) and it is at least 2 for instances in (ii). Thus, using ALG we can determine in polynomial time whether $\mathscr{I}$ is a yes- or no-instance, which is a contradiction to the assumption that $P \neq NP$. It remains to show the above two statements.

For (i), we simply delete all arcs in G that are not contained in P_1 and P_2. Then, splitting the flow evenly along these paths yields a Nash equilibrium with cost $d(H,1,\ell) = 3/2$.

For (ii), we can assume without loss of generality that any subgraph H contains an s,t-path. If H has an (s,s_2,t_1,t) path then routing the flow along this path yields a Nash flow with cost $d(H,1,\ell) = 2$. Suppose H does not contain an (s,s_2,t_1,t) path. Because $\mathscr{I}$ is a no-instance, we have three possibilities:

1. H contains an (s,s_1,t_1,t) path but no (s,s_2,t_2,t) paths (otherwise two such paths must share a vertex and H would contain an (s,s_2,t_1,t) path);
2. H contains an (s,s_2,t_2,t) path but no (s,s_1,t_1,t) path (otherwise two such paths must share a vertex and H would contain an (s,s_2,t_1,t) path);
3. every s,t-path in H is an (s,s_1,t_2,t) path.

It is not hard to verify that in either case, the cost of a Nash flow is $d(H,1,\ell) = 2$. □

3 Congestion Games

In this section, we consider a general class of resource allocation games, called *congestion games*.

3.1 Model

Definition 3.1 (Congestion model)**.** A *congestion model* $\mathscr{M} = (N, E, (X_i)_{i\in N}, (c_e)_{e\in E})$ is given by

- a set of players $N = [n]$;
- a set of facilities E;
- for every player $i \in N$, a set $X_i \subseteq 2^E$ of subsets of facilities in E;[2]
- for every facility $e \in E$, a cost function $c_e : \mathbb{N} \to \mathbb{R}$.

For every player $i \in N$, X_i is the strategy set from which i can choose. A strategy $x_i \in X_i$ is a subset of facilities; we think of x_i as the facilities that player i uses. Fix some strategy profile $x = (x_1, \dots, x_n) \in X := X_1 \times \cdots \times X_n$. The cost incurred for the usage of facility $e \in E$ with respect to x is defined as $c_e(n_e(x))$, where

$$n_e(x) := |\{i \in N \;:\; e \in x_i\}|$$

refers to the total number of players that use e.

Definition 3.2 (Congestion game)**.** The *congestion game* corresponding to the congestion model $\mathscr{M} = (N, E, (X_i)_{i\in N}, (c_e)_{e\in E})$ is the strategic game $\Gamma = (N, (X_i)_{i\in N}, (c_i)_{i\in N})$, where every player $i \in N$ wants to minimize his cost

$$c_i(x) = \sum_{e\in x_i} c_e(n_e(x)).$$

The game is called *symmetric* if all players have the same strategy set, i.e., $X_i = Q$ for all $i \in N$ and some $Q \subseteq 2^E$.

3.2 Example: Atomic Network Congestion Game

Example 3.1 (Atomic network congestion game)**.** The *atomic network congestion game* can be modeled as a congestion game: We are given a directed graph $G = (V, A)$, a single commodity $(s,t) \in V \times V$, and a cost function $c_a : \mathbb{N} \to \mathbb{R}_+$ for every arc $a \in A$. Every player $i \in N$ wants to send one unit of flow from s to t along a single path. The set of facilities is $E := A$ and the strategy set X_i of every player $i \in N$ is simply the set of all directed s,t-paths in G. (Note that the game is symmetric.) The goal of every player

[2]For a given set S, we use 2^S to refer to the *power set* of S, i.e., the set of all subsets of S.

$i \in N$ is to choose a path $x_i \in X_i$ so as to minimize his cost

$$c_i(x) := \sum_{a \in x_i} c_a(n_a(x)),$$

where $n_a(x)$ refers to the total number of players using arc a. This example corresponds to a selfish routing game, where every player controls one unit of flow (i.e., we have atomic players) and has to route his flow unsplittably from s to t.

3.3 Congestion Games are Exact Potential Games

Theorem 3.1. *Every congestion game $\Gamma = (N, (X_i)_{i \in N}, (c_i)_{i \in N})$ is an exact potential game.*

Proof. Rosenthal's potential function $\Phi : X \to \mathbb{R}$ is defined as

$$\Phi(x) := \sum_{e \in E} \sum_{k=1}^{n_e(x)} c_e(k). \tag{11}$$

We prove that Φ is an exact potential function for Γ. To see this, fix some $x \in X$, a player $i \in N$ and some $y_i \in X_i$. We have

$$\begin{aligned} \Phi(x_{-i}, y_i) &= \sum_{e \in E} \sum_{k=1}^{n_e(x)} c_e(k) + \sum_{e \in y_i \setminus x_i} c_e(n_e(x)+1) - \sum_{e \in x_i \setminus y_i} c_e(n_e(x)) \\ &= \Phi(x) + c_i(x_{-i}, y_i) - c_i(x). \end{aligned}$$

Thus, Φ is an exact potential function. □

By Theorem 1.3 it follows that every congestion game has the FIP and admits a pure Nash equilibrium.

3.4 Price of Anarchy

Define the *social cost* of a strategy profile $x \in X$ as the total cost of all players, i.e.,

$$C(x) := \sum_{i \in N} c_i(x) = \sum_{e \in E} n_e(x) c_e(n_e(x)).$$

We derive an upper bound on the price of anarchy for congestion games with respect to the social cost function c defined above. Here we only consider the case that the cost of every facility $e \in E$ is given as $c_e(k) = k$. The proof extends to arbitrary linear latency functions.

Theorem 3.2. *Let $\mathcal{M} = (N, (X_i)_{i \in N}, (c_i)_{i \in N})$ be a congestion model with linear latency functions $c_e(k) = k$ for every $e \in E$ and let $\Gamma = (N, (X_i)_{i \in N}, (u_i)_{i \in N})$ be the corresponding congestion game. The price of anarchy is at most $5/2$.*

We will use the following fact to prove this theorem (whose proof we leave as an exercise):

Fact 3.1. *Let α and β be two non-negative integers. Then*

$$\alpha(\beta+1) \leq \frac{5}{3}\alpha^2 + \frac{1}{3}\beta^2.$$

Proof of Theorem 3.2. Let x be a Nash equilibrium and x^* be an optimal strategy profile minimizing C. Since x is a Nash equilibrium, the cost of every player $i \in N$ does not decrease if he deviates to his optimal strategy x_i^*, i.e.,

$$c_i(x) \leq c_i(x_{-i}, x_i^*) = \sum_{e \in x_i^*} c_e(n_e(x_{-i}, x_i^*)) = \sum_{e \in x_i^*} n_e(x_{-i}, x_i^*) \leq \sum_{e \in x_i^*} n_e(x) + 1,$$

where the last inequality follows since player i increases the number of players on each $e \in x_i^*$ by at most 1 with respect to $n_e(x)$. Summing over all players, we obtain

$$C(x) = \sum_{i \in N} c_i(x) \leq \sum_{i \in N} \sum_{e \in x_i^*} n_e(x) + 1 = \sum_{e \in E} n_e(x^*)(n_e(x) + 1).$$

Using Fact 3.1, we therefore obtain

$$C(x) \leq \sum_{e \in E} n_e(x^*)(n_e(x) + 1) \leq \frac{5}{3} \sum_{e \in E} (n_e(x^*))^2 + \frac{1}{3} \sum_{e \in E} (n_e(x))^2 = \frac{5}{3} C(x^*) + \frac{1}{3} C(x),$$

where the last equality follows from $c_e(k) = k$ for every $e \in E$ and the definition of C. We conclude that $C(x) \leq \frac{5}{2} C(x^*)$. □

The following example shows that the bound is tight.

Example 3.2. Consider a congestion game with three players and six facilities $E = E_1 \cup E_2$, where $E_1 = \{h_1, h_2, h_3\}$ and $E_2 = \{g_1, g_2, g_3\}$. The delay functions are given by $d_e(x) = x$ for every $e \in E$.

Each player i has two pure strategies: $\{h_i, g_i\}$ and $\{h_{i-1}, h_{i+1}, g_{i+1}\}$ (all indices are modulo 3). The strategy profile in which every player selects his first strategy is a social optimum of cost 6. Consider the strategy profile s in which every player chooses his second strategy. Each player's cost with respect to s is 5. If a player unilaterally deviates to his first strategy, then his new cost is 5. Thus s is a Nash equilibrium of total cost 15. We conclude that the price of anarchy is at least $15/6 = 5/2$.

4 Smoothness of Games

Consider a strategic game $\Gamma = (N, (S_i)_{i\in N}, (c_i)_{i\in N})$ with social cost function

$$C(s) = \sum_{i\in N} c_i(s),$$

where $s \in S = S_1 \times \cdots \times S_n$.

The following definition will be useful in proving bounds on the price of anarchy of Γ.

Definition 4.1 ((λ,μ)-smoothness)**.** Let Γ be a strategic game with social cost function C. Γ is (λ,μ)-*smooth* iff for any two strategy profiles $s, s^* \in \Sigma$,

$$\sum_{i\in N} c_i(s_i^*, s_{-i}) \leq \lambda C(s^*) + \mu C(s). \tag{12}$$

Theorem 4.1. *Let Γ be a strategic game. If Γ is (λ,μ)-smooth with $\mu < 1$, then the price of anarchy of Γ is at most $\frac{\lambda}{1-\mu}$.*

Proof. Let s be a Nash equilibrium and s^* be a social optimum of Γ. Because s is a Nash equilibrium, we have for every $i \in N$: $c_i(s_i, s_{-i}) \leq c_i(s_i^*, s_{-i})$. Summing over all players, we obtain

$$C(s) = \sum_{i\in N} c_i(s_i, s_{-i}) \leq \sum_{i\in N} c_i(s_i^*, s_{-i}) \leq \lambda C(s^*) + \mu C(s),$$

where the last inequality holds because Γ is (λ,μ)-smooth. The proof now follows because $\mu < 1$. □

We define the *robust price of anarchy* as the best possible bound on the price of anarchy obtainable by a (λ,μ)-smoothness argument.

Definition 4.2. The *robust price of anarchy* of a strategic game Γ is defined as

$$\mathsf{RPOA}(\Gamma) = \inf\left\{\frac{\lambda}{1-\mu} \;:\; \Gamma \text{ is } (\lambda,\mu)\text{-smooth with } \mu < 1\right\}.$$

For a class $\mathscr{G}$ of games, we define

$$\mathsf{RPOA}(\mathscr{G}) = \sup\{\mathsf{RPOA}(\Gamma) \;:\; \Gamma \in \mathscr{G}\}.$$

We omit the explicit reference to the game (or class of games) if it is clear from the context.

4.1 Correlated Equilibria

Consider the following two player game, also called *chicken*: Two players are speeding towards an intersection. Each player has two options: stop or go. If both go, the outcome is a fatal crash and both players experience a utility of 0. If one goes and the other stops, the one that goes experiences a utility of 5 while the other experiences a utility of 1. If both stop, both experience a utility of 4.

	stop	go
stop	$(4,4)$	$\boxed{(1,5)}$
go	$\boxed{(5,1)}$	$(0,0)$

There are two pure Nash equilibria: (stop, go) and (go, stop). There is one mixed Nash equilibrium in which every player stops with probability $\frac{1}{2}$.

Let $p(s)$ be the probability of strategy profile $s \in S$. The above equilibria then correspond to the probability distributions

$$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \quad \begin{pmatrix} \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} \end{pmatrix}$$

Now consider the following probability distribution

$$\begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix}$$

Note that this probability distribution cannot be generated by two independent probability distributions of player 1 and 2.

Suppose some trusted third party (*mediator*) draws an outcome s from this distribution and recommends to each player individually and privately to play s_i according to the outcome. In the chicken game, this would mean that half of the time player 1 stops and player 2 goes and vice versa. That is, this probability distribution can be interpreted as a traffic signal.

Note that this probability distribution of recommendations is self-enforcing in the sense that no player has an incentive to deviate from the recommendation, assuming that all other players follow the recommendation.

In the above example, player 1 has an expected utility of

- $\frac{1}{2}(5+1) = 3$ if he always follows the recommendation of the mediator
- $\frac{1}{2}(5+0) = 2.5$ if he follows the recommendation go but deviates for stop
- $\frac{1}{2}(4+1) = 2.5$ if he follows the recommendation stop but deviates for go
- $\frac{1}{2}(4+0) = 2$ if he always deviates from the recommendation.

We conclude that it is best for player 1 to always follow the recommendation. A similar argument holds for player 2.

Definition 4.3. A *correlated equilibrium (CE)* of a strategic game $\Gamma = (N, (S_i)_{i \in N}, (u_i)_{i \in N})$ is a probability distribution $p : S \to [0,1]$ over $S = S_1 \times \cdots \times S_n$ such that for every player $i \in N$ and every two strategies $s_i, s_i' \in S_i$ of i, conditioned on the event that a strategy profile with s_i as his strategy is drawn from p, the expected utility of player i playing s_i is no smaller than that of playing s_i':

$$\sum_{s_{-i}\in S_{-i}} (u_i(s_{-i},s_i) - u_i(s_{-i},s_i'))p(s_{-i},s_i) \geq 0$$

Note that every mixed Nash equilibrium is also a correlated equilibrium but not the other way around. In the chicken game, the above conditions read as follows:

$$\begin{aligned}
(4-5)p_{11} + (1-0)p_{12} &\geq 0 \qquad \text{(player 1 plays stop)}\\
(5-4)p_{21} + (0-1)p_{22} &\geq 0 \qquad \text{(player 1 plays go)}\\
(4-5)p_{11} + (1-0)p_{21} &\geq 0 \qquad \text{(player 2 plays stop)}\\
(5-4)p_{12} + (0-1)p_{22} &\geq 0 \qquad \text{(player 2 plays go)}
\end{aligned}$$

It is not hard to verify that there is yet another correlated equilibrium, which is

$$\begin{pmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 0 \end{pmatrix}$$

Note that the correlated equilibrium conditions reduce to a set of linear inequalities. Moreover, we know that at least one solution must exist (because of Nash's existence theorem of mixed Nash equilibria). We can therefore find a correlated equilibrium efficiently by solving a linear program. This is in stark contrast to the problem of finding a mixed Nash equilibrium, which is computationally very hard.

4.2 Learning and Regret Matching Strategies

4.2.1 Example: Play Against Nature

Consider the following game in which one player (player 1) plays against nature (player N). Each day, the player can choose to either take an umbrella or to not take an umbrella and nature determines whether it will rain or not. The utility u_1 of player 1 is given as follows:

	rain (r)	no rain (nr)
umbrella (u)	1	0
no umbrella (nu)	0	1

Let $s_1(t) \in S_1 = \{\text{u}, \text{nu}\}$ denote the strategy chosen by player 1 on day t and let $s_N(t) \in S_N = \{\text{r}, \text{nr}\}$ be the event that occurs on day t. The utility of player 1 at day t is $u_1(s_1(t), s_N(t))$.

Define the *average utility at day t* of player 1 as

$$\bar{U}_1(t) = \frac{1}{t}\sum_{\tau=1}^{t} u_1(s_1(\tau), s_N(\tau))$$

Let the *average utility at day t for fixed strategy* $q \in S_1$ of player 1 be defined as

$$\bar{U}_1^q(t) = \frac{1}{t}\sum_{\tau=1}^{t} u_1(q, s_N(\tau))$$

The *average regret at day t with respect to fixed strategy* $q \in S_1$ of player 1 is given by

$$\bar{R}_1^q(t) = \bar{U}_1^q(t) - \bar{U}_1(t).$$

Consider the following example:

day (t)	1	2	3	4	5	6
nature	r	nr	r	r	nr	r
player 1	nu	u	nu	u	nu	nu
u_1	0	0	0	1	1	0
$\bar{U}_1(t)$	0	0	0	$\frac{1}{4}$	$\frac{2}{5}$	$\frac{2}{6}$
$\bar{U}_1^{\text{u}}(t)$	1	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{3}{4}$	$\frac{3}{5}$	$\frac{4}{6}$
$\bar{R}_1^{\text{u}}(t)$	1	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{1}{5}$	$\frac{1}{3}$
$\bar{U}_1^{\text{nu}}(t)$	0	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{2}{5}$	$\frac{2}{6}$
$\bar{R}_1^{\text{nu}}(t)$	0	$\frac{1}{2}$	$\frac{1}{3}$	0	0	0

Observe that whenever player 1 experiences some positive regret, he could have chosen a better fixed strategy in *hindsight*.

The question that arises is: Can player 1 choose his strategy every day so as to guarantee that positive regret vanishes asymptotically, irrespective of nature?

Consider the following *regret matching strategy*: At day $t+1$, player 1 plays strategy u and nu with probabilities $p_1^{\text{u}}(t+1)$ and $p_1^{\text{nu}}(t+1)$, respectively, with

$$p_1^{\text{u}}(t+1) = \frac{[\bar{R}_1^{\text{u}}(t)]_+}{[\bar{R}_1^{\text{u}}(t)]_+ + [\bar{R}_1^{\text{nu}}(t)]_+} \quad \text{and} \quad p_1^{\text{nu}}(t+1) = \frac{[\bar{R}_1^{\text{nu}}(t)]_+}{[\bar{R}_1^{\text{u}}(t)]_+ + [\bar{R}_1^{\text{nu}}(t)]_+},$$

where $[x]_+$ refers to $\max\{x, 0\}$. Note that the above is well-defined only if the denominator is positive. If not, we let $p_1(t+1)$ be an arbitrary probability distribution over S_1.

Using the above regret matching strategy, one can show that the player's positive regret vanishes, i.e.,

$$[\bar{R}_1^{\text{u}}(t)]_+ \to 0 \quad \text{and} \quad [\bar{R}_1^{\text{nu}}(t)]_+ \to 0 \quad \text{as } t \to \infty.$$

4.2.2 Regret Matching in Strategic Games

The above strategy can be generalized to arbitrary strategic games. Let $\Gamma = (N, (S_i)_{i\in N}, (u_i)_{i\in N})$ be a strategic game. Suppose Γ is played repeatedly over a discrete time horizon $t = 1, 2, \ldots$. We assume that at each time $t \in \{1, 2, \ldots\}$, each player

$i \in N$ simultaneously selects his strategy according to a probability distribution $p_i(t)$ over S_i, i.e., $p_i^q(t)$ is the probability of choosing $q \in S_i$ at time t. Let $s(1), s(2), \ldots$ be the sequence of outcomes of this repeated play.
The *average utility of player i at time t* is

$$\bar{U}_i(t) = \frac{1}{t} \sum_{\tau=1}^{t} u_i(s(\tau)).$$

The *average utility of player i for fixed strategy $q \in S_i$ at time t* is

$$\bar{U}_i^q(t) = \frac{1}{t} \sum_{\tau=1}^{t} u_i(q, s_{-i}(\tau)).$$

The *average regret of player i for fixed strategy $q \in S_i$ at time t* is defined as

$$\bar{R}_i^q(t) = \bar{U}_i^q(t) - \bar{U}_i(t).$$

Note that $\bar{R}_i^q(t)$ is the average regret of player i at time t of not having played q at each step.

We impose that each player updates his strategy $p_i(t)$ using a *learning rule* $f(\cdot)$ that is based on the history of the game, i.e.,

$$p_i(t+1) = f(s(1), \ldots, s(t), u_i).$$

Suppose each player updates his strategy according to the *regret matching strategy*: Define

$$p_i^q(t+1) = \frac{[\bar{R}_i^q(t)]_+}{\sum_{s_i \in S_i} [\bar{R}_i^{s_i}(t)]_+} \tag{13}$$

if the denominator is positive and let $p_i(t+1)$ be an arbitrary distribution of S_i otherwise.

We state the following theorem without proof:

Theorem 4.2. *Suppose that player i follows the regret matching strategy* (13). *Then, irrespective of what the other players do, for every strategy $q \in S_i$,*

$$[\bar{R}_i^q(t)]_+ \to 0 \quad as \quad t \to \infty.$$

An alternative interpretation of the regret matching strategy is as follows: Let $p(t)$ be the empirical distribution of *joint* play of all players, i.e., $p(t,s)$ is the frequency of strategy profile $s \in S$ up to time t. It is not hard to see that with this notation the *expected utility of player i at time t* is

$$\bar{U}_i(t) = \sum_{s \in S} u_i(s) p(t,s).$$

The *expected utility of player i for fixed strategy $q \in S_i$ at time t* is

$$\bar{U}_i^q(t) = \sum_{s_{-i} \in S_{-i}} u_i(q, s_{-i}) p_{-i}(t, s_{-i}),$$

where $p_{-i}(t, s_{-i})$ refers to the frequency of strategy profile s_{-i} played by players other than i up to time t. The *expected regret of player i for fixed strategy $q \in S_i$ at time t* is defined as

$$\bar{R}_i^q(t) = \bar{U}_i^q(t) - \bar{U}_i(t).$$

Note that the characteristic of a no-regret point (i.e., for sufficiently large t) is that for every player $i \in N$ and strategy $q \in S_i$, $\bar{R}_i^q(t) \leq 0$, which is equivalent to

$$\sum_{s_{-i} \in S_{-i}} u_i(q, s_{-i}) p_{-i}(t, s_{-i}) \leq \sum_{s \in S} u_i(s) p(t, s).$$

This coincides with the definition of a *coarse correlated equilibrium*, given in the next section.

Theorem 4.3. *Suppose every player plays the regret matching strategy according to (13). The empirical distribution $p(t)$ then converges to a coarse correlated equilibrium.*

4.3 Coarse Correlated Equilibria

Another solution concept is the one of a coarse correlated equilibrium. Given a probability distribution p, let the *marginal probability* $p_{-i}(s_{-i})$ that $s_{-i} \in S_{-i}$ will be realized be defined as

$$p_{-i}(s_{-i}) = \sum_{s_i \in S_i} p(s_i, s_{-i}).$$

Definition 4.4. A *coarse correlated equilibrium (CCE)* of a strategic game $\Gamma = (N, (S_i)_{i \in N}, (u_i)_{i \in N})$ is a probability distribution $p : S \longrightarrow [0,1]$ over $S = S_1 \times \cdots \times S_n$ such that for every player $i \in N$ and every $s_i' \in S_i$ of i

$$\sum_{s \in S} u_i(s_{-i}, s_i) p(s_{-i}, s_i) \geq \sum_{s_{-i} \in S_{-i}} u_i(s_{-i}, s_i') p_{-i}(s_{-i}).$$

Coarse correlated equilibria generalize correlated equilibria. The hierarchy of equilibrium concepts that have been introduced is depicted in Figure 7.

4.4 Robust Price of Anarchy

Suppose Γ is a strategic game with robust price of anarchy RPOA. It is not hard to verify that the proof of Theorem 4.1 continues to hold for the more general solution concepts mentioned above.

Theorem 4.4. *Let Γ be a strategic game. If Γ is (λ, μ)-smooth with $\mu < 1$, then the coarse correlated price of anarchy of Γ is at most $\frac{\lambda}{1-\mu}$.*

Proof. Let p be a coarse equilibrium for a (λ, μ)-smooth game, let s be a random variable with distribution p, and let $s^* \in S$ be an arbitrary strategy profile. The coarse

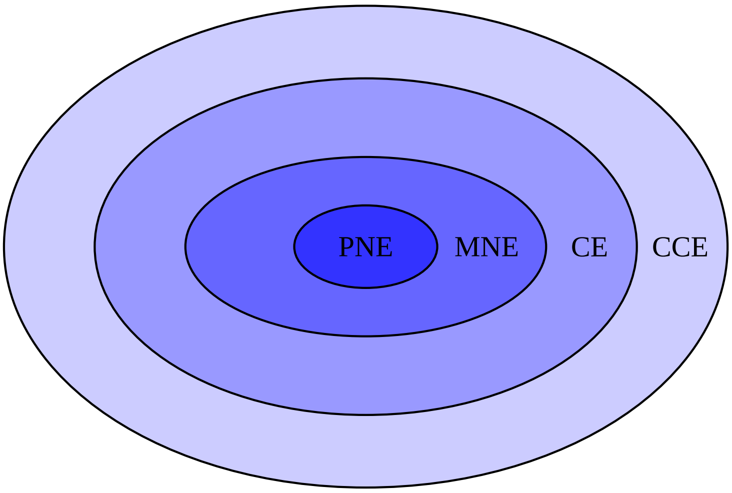

Figure 7: Hierarchy of equilibrium concepts.

equilibrium condition implies that for every player $i \in N$:

$$\mathbf{E}_{s \leftarrow p}[C_i(s)] \leq \mathbf{E}_{s_{-i} \leftarrow p_{-i}}[C_i(s_i^*, s_{-i})] = \mathbf{E}_{s \leftarrow p}[C_i(s_i^*, s_{-i})].$$

By linearity of expectation it then also holds that

$$\mathbf{E}_{s \leftarrow p}\left[\sum_{i \in N} C_i(s)\right] \leq \sum_{i \in N} \mathbf{E}_{s \leftarrow p}[C_i(s_i^*, s_{-i})] = \mathbf{E}_{s \leftarrow p}\left[\sum_{i \in N} C_i(s_i^*, s_{-i})\right].$$

Now we use the smoothness property (12) and obtain

$$\mathbf{E}_{s \leftarrow p}[C(s)] \leq \mathbf{E}_{s \leftarrow p}[\lambda C(s^*) + \mu C(s)] = \lambda C(s^*) + \mu \mathbf{E}_{s \leftarrow p}[C(s)].$$

Since $\mu < 1$, the coarse price of anarchy is at most $\frac{\lambda}{1-\mu}$. □

The smoothness condition also proves useful in the context of no-regret sequences. Consider a sequence $s^1, \ldots, s^T$ of outcomes of a (λ, μ)-smooth game Γ. Let s^* be an optimal outcome that minimizes the social cost function C. Define $\delta_i(s^t) = C_i(s^t) - C_i(s^t_{-i}, s^*_i)$ for every $i \in N$ and $t \in \{1, \ldots, T\}$. Let $\Delta(s^t) = \sum_{i=1}^n \delta_i(s^t)$. Exploiting the (λ, μ)-property, we obtain

$$\Delta(s^t) = \sum_{i=1}^n C_i(s^t) - C_i(s^t_{-i}, s^*_i) \geq C(s^t) - \lambda C(s^*) - \mu C(s^t) = (1-\mu)C(s^t) - \lambda C(s^*).$$

Thus,

$$C(s^t) \leq \frac{\lambda}{1-\mu} C(s^*) + \frac{\Delta(s^t)}{1-\mu}. \tag{14}$$

Suppose that $s^1, \ldots, s^T$ is a sequence of outcomes in which every player experiences vanishing average external regret, i.e., for every player $i \in N$

$$\sum_t C_i(s^t) \leq \left[\min_{s_i' \in S_i} \sum_t C_i(s_i', s_{-i}^t)\right] + o(T).$$

We obtain that for every player $i \in N$:

$$\frac{1}{T}\sum_{t=1}^{T} \delta_i(t) \leq \frac{1}{T}\left(\sum_t C_i(s^t) - \min_{s_i' \in S_i} \sum_t C_i(s_i', s_{-i}^t)\right) = o(1).$$

By summing over all players, we obtain that the average cost of the sequence of T outcomes is

$$\frac{1}{T}\sum_{t=1}^{T} C(s^t) \leq \frac{\lambda}{1-\mu} C(s^*) + \frac{1}{1-\mu}\sum_{i=1}^{n}\left(\frac{1}{T}\sum_{t=1}^{T}\delta_i(t)\right) \stackrel{T\to\infty}{\longrightarrow} \frac{\lambda}{1-\mu} C(s^*).$$

4.5 Example: Congestion Games

Theorem 4.5. *Let $\Gamma = (N, (X_i)_{i\in N}, (c_i)_{i\in N})$ be a congestion game with linear cost functions $c_e(x) = x$. Γ is $(\frac{5}{3}, \frac{1}{3})$-smooth.*

Proof. Let x and x^* be two arbitrary strategy profiles of Γ. We have

$$\begin{aligned}\sum_{i\in N} c_i(x_{-i}, x_i^*) &= \sum_{i\in N}\sum_{e\in x_i^*} c_e(n_e(x_{-i}, x_i^*)) = \sum_{i\in N}\sum_{e\in x_i^*} n_e(x_{-i}, x_i^*) \\ &\leq \sum_{i\in N}\sum_{e\in x_i^*} n_e(x) + 1 = \sum_{e\in E} n_e(x^*)(n_e(x)+1).\end{aligned}$$

Using Fact 3.1, we therefore obtain

$$\sum_{e\in E} n_e(x^*)(n_e(x)+1) \leq \frac{5}{3}\sum_{e\in E}(n_e(x^*))^2 + \frac{1}{3}\sum_{e\in E}(n_e(x))^2 = \frac{5}{3}C(x^*) + \frac{1}{3}C(x),$$

where the last equality follows from $c_e(k) = k$ for every $e \in E$ and the definition of C. □

5 Combinatorial Auctions

In this section, we present a few examples from the area of *mechanism design*. The fundamental questions that one attempts to address in mechanism design is the following: Assuming that players act strategically, how should we design the rules of the game such that the players' strategic behavior leads to a certain desirable outcome of the game? As a motivating example, we first consider one of the simplest auctions, known as *Vickrey Auction*. We then turn to more general combinatorial auctions.

5.1 Vickrey Auction

Suppose there is an auctioneer who wishes to auction off a single item. An instance of the *single-item auction* consists of

- a set of players $N = [n]$ that are interested in obtaining the item;
- every player $i \in N$ has a *private valuation* v_i which specifies how much the item is worth to player i; v_i is only known to player i.
- every player i has a *bid* b_i which represents the maximum amount player i declares to be willing to pay for the item.

The auctioneer receives the bids and needs to determine who receives the item and at what price. A *mechanism* can be thought of as a protocol (or algorithm) that the auctioneer runs in order to make this decision. That is, based on the submitted bids $(b_i)_{i \in N}$, the mechanism determines

1. a player i^* in N, called the *winner*, who receives the item, and
2. a price p that this player has to pay for the item.

We define $x_i = 1$ if player $i \in N$ wins the auction and $x_i = 0$ otherwise. We model a player's preferences over different outcomes of the game by means of a utility function. Lets assume that the utility function of player i represents the *net gain*, defined as $u_i = x_i(v_i - p)$. Note that the utility is zero if the player does not receive the item. Otherwise, it is his private valuation minus the price he has to pay. Such utility functions are also called *quasi-linear*.

So what is a good mechanism? There are several natural properties that we may want to achieve:

(P1) *Strategyproofness*: Every player maximizes his utility by bidding *truthfully*, i.e., $b_i = v_i$.

(P2) *Efficiency*: Assuming that every player bids truthfully, the mechanism computes an outcome that maximizes the *social welfare*, i.e., among all possible outcomes x it chooses one that maximizes the total valuation $\sum_{i \in N} x_i v_i$; here, this is equivalent to require that the mechanism chooses the player with maximum valuation as the winner.

(P3) *Polynomial-time computability*: The outcome should be computable in polynomial

Algorithmus 2 Vickrey Auction

1: Collect the bids $(b_i)_{i\in N}$ of all players.
2: Choose a player $i^* \in N$ with highest bid (break ties arbitrarily).
3: Charge i^* the second highest bid $p := \max_{i\neq i^*} b_i$.

time.

As it turns out, there is a remarkable mechanism due to Vickrey that satisfies all these properties; this mechanism is also known as *Vickrey auction* or *second-price auction* (see Algorithm 2).

Lemma 5.1. *In a Vickrey Auction, every player i maximizes his utility by bidding truthfully $b_i = v_i$. More precisely, for every player $i \in N$ and every bidding profile b_{-i} of the other players, we have*

$$u_i(b_{-i}, v_i) \geq u_i(b_{-i}, b_i) \quad \forall b_i.$$

Moreover, this holds true even if player i knows the bids of all the other players.

Proof. Consider player i and fix a bidding profile b_{-i} of the other players. Let $B = \max_{j\neq i} b_j$ be the highest bid if player i does not participate in the game.

Assume $v_i \leq B$. Then player i has zero utility if he bids truthfully: Note that player i loses if $v_i < B$ and may win if $v_i = B$ (depending on the tie breaking rule); however, in both cases his utility is zero. His utility remains zero for every bid $b_i < B$ or if $b_i = B$ and i loses (due to the tie breaking rule). Otherwise, $b_i = B$ and i wins or $b_i > B$. In both cases i wins and pays B. However, his utility is then $u_i = v_i - B \leq 0$, which is less than or equal to the utility he obtains if he bids truthfully.

Next assume that $v_i > B$. If player i bids truthfully, he wins and receives a positive utility $u_i = v_i - B > 0$. He is worse off by obtaining a utility of zero if he bids $b_i < B$ or if he bids $b_i = B$ and loses (due to the tie breaking rule). Otherwise $b_i = B$ and i wins or $b_i > B$. In both cases, i wins and receives a utility of $u_i = v_i - B > 0$, which is the same as if he had bid $b_i = v_i$. □

Moreover, we can prove that if a player does not bid truthfully, he may actually run the risk to be strictly worse off.

Lemma 5.2. *For every bid $b_i \neq v_i$ of player i there is a bidding profile b_{-i} of the other players such that $u_i(b_{-i}, b_i) < u_i(b_{-i}, v_i)$.*

Proof. See Assignment 2. □

It is easy to see that the Vickrey Auction satisfies (P2) and (P3) as well. More specifically, it satisfies (P2) since it selects the winner i^* to be a player whose valuation is maximum, assuming that every bidder bids truthfully. Moreover, its computation time is linear in the number of players n. We can thus summarize:

Theorem 5.1. *The Vickrey Auction is strategyproof, efficient and runs in polynomial time.*

5.2 Combinatorial Auctions and the VCG Mechanism

We now turn to a more general model of auctions. Suppose there is a set M of $m \geq 1$ items to be auctioned off to n players. A player may now be interested in a *bundle* $S \subseteq M$ of items. Every player $i \in N$ has a private valuation function $v_i : 2^M \to \mathbb{R}^+$, where $v_i(S)$ specifies player i's value for receiving the items in $S \subseteq M$. We say $v_i(S)$ is the valuation of player i for bundle S. We assume that $v_i(\emptyset) = 0$. (Although this and the assumption that $v_i(\cdot)$ is non-negative is not essential here).

If every player has a separate value for each item and the value of a subset $S \subseteq M$ is equal to the sum of all values of the items in S, then we can simply run a separate Vickrey Auction for every item. However, this assumption ignores the possibility that differnt bundles may have different values. More precisely, for a player i, items in $S \subseteq M$ might be

- *substitutes*: the player's valuation to obtain the entire bundle S might be less than or equal to the individual valuations of the items in S, i.e., $v_i(S) \leq \sum_{k \in S} v_i(\{k\})$; for example, if the items in S are (partially) redundant.
- *complements*: the player's valuation to obtain the entire bundle S might be greater or equal to the individual valuations of the items in S, i.e., $v_i(S) \geq \sum_{k \in S} v_i(\{k\})$; for example, if the items in S are (partially) dependent.

Here, we consider the most general setting, where we do not make any assumption on the valuation functions v_i of the players.

Let O denote the set of all possible allocations of the items in M to the players. An allocation $a \in O$ is a function $a : M \to N \cup \{\perp\}$ that maps every item to one of the players in N or to $\perp$, which means that the item remains unassigned. Let $a^{-1}(i)$ be the subset of items that player $i \in N$ receives. Every player declares a bid $b_i(S)$ for every bundle $S \subseteq M$. (Lets not care about polynomial-time computability for a moment.) For the sake of conciseness, we slightly abuse notation: Given an allocation $a \in O$, we write $v_i(a)$ and $b_i(a)$ to refer to $v_i(a^{-1}(i))$ and $b_i(a^{-1}(i))$, respectively. The auctioneer needs to decide how to distribute the items among the players in N and at what price. That is, he determines an allocation $a \in O$ and a pricing vector $p = (p_i)_{i \in N}$, where player i obtains the bundle $a^{-1}(i)$ at a price of p_i. As before, we consider quasi-linear utility functions: The utility of player i, given the outcome (a, p), is $u_i = v_i(a) - p_i$.

A mechanism is strategyproof in this setting if a dominant strategy for every player is to bid $b_i(S) = v_i(S)$ for every $S \subseteq M$. Moreover, a mechanism is efficient, if it outputs an allocation a^* that maximizes the total social welfare, i.e., $a^* = \arg\max_{a \in O} \sum_{i \in N} v_i(a)$, assuming that every player truthfully reports his valuation.

A powerful mechanism for this quite general class of combinatorial auctions is known as *VCG mechanism* due to Vickrey, Clarke and Groves (see Algorithm 3). In particular, as we will see, the VCG mechanism is strategyproof and efficient.

Algorithmus 3 VCG mechanism

1: Collect the bids $(b_i(S))$ for every player $i \in N$ and every set $S \subseteq M$.
2: Choose an allocation $a^* \in O$ such that

$$a^* = \arg\max_{a \in O} \sum_{i \in N} b_i(a).$$

3: Compute the price p_i of player i as

$$p_i := b_i(a^*) - \underbrace{\left(\max_{a \in O} \sum_{j \in N} b_j(a) - \max_{a \in O} \sum_{j \in N, j \neq i} b_j(a) \right)}_{i\text{'s contribution to the total social welfare}}.$$

4: **return** a^*

Theorem 5.2. *The VCG mechanism is strategyproof and efficient.*

Proof. Clearly, if every player bids truthfully the allocation a^* output by the VCG mechanism maximizes total social welfare. Thus, the VCG mechanism is efficient.

We next prove that the VCG mechanism is strategyproof. Consider an arbitrary player $i \in N$. Let $b = (b_{-i}, b_i)$ be the bid vector of some arbitrary bids and let $\bar{b} = (b_{-i}, v_i)$ be the same bid vector, except that player i reports his private valuations truthfully. Moreover, let (a^*, p) and $(\bar{a}^*, \bar{p}_i)$ be the outcome computed by the VCG mechanism for input b and $\bar{b}$, respectively. Observe that we have

$$\bar{b}_i(a) = v_i(a) \quad \forall a \in O \quad \text{and} \quad \bar{b}_j(a) = b_j(a) \quad \forall j \neq i,\ \forall a \in O. \tag{15}$$

Moreover, $\bar{a}^*$ has been chosen such that

$$\sum_{j \in N} \bar{b}_j(\bar{a}^*) \geq \sum_{j \in N} \bar{b}_j(a) \quad \forall a \in O. \tag{16}$$

Using these two observations, we can infer:

$$\begin{aligned}
v_i(\bar{a}^*) - \bar{p}_i = v_i(\bar{a}^*) - \left[\bar{b}_i(\bar{a}^*) - \left(\max_{a \in O} \sum_{j \in N} \bar{b}_j(a) - \max_{a \in O} \sum_{j \in N, j \neq i} \bar{b}_j(a) \right) \right] \\
&\overset{(15)}{=} \max_{a \in O} \sum_{j \in N} \bar{b}_j(a) - \max_{a \in O} \sum_{j \in N, j \neq i} b_j(a) \\
&= \sum_{j \in N} \bar{b}_j(\bar{a}^*) - \max_{a \in O} \sum_{j \in N, j \neq i} b_j(a) \\
&\overset{(16)}{\geq} \sum_{j \in N} \bar{b}_j(a^*) - \max_{a \in O} \sum_{j \in N, j \neq i} b_j(a) \\
&\overset{(15)}{=} \sum_{j \in N, j \neq i} b_j(a^*) + v_i(a^*) - \max_{a \in O} \sum_{j \in N, j \neq i} b_j(a)
\end{aligned}$$

$$= v_i(a^*) - \left[b_i(a^*) - \left(\max_{a \in O} \sum_{j \in N} b_j(a) - \max_{a \in O} \sum_{j \in N, j \neq i} b_j(a)\right)\right]$$
$$= v_i(a^*) - p_i.$$

Thus, $b_i = v_i$ is a dominant strategy for player i. □

Although the VCG mechanism satisfies strategyproofness and efficiency, it is highly computationally intractable. In particular, there are two sources of inefficiency:

1. Collecting the bids of a single player already takes exponential time (in the number m of objects).
2. Computing the optimal allocation $a^* \in O$ may be a computationally hard problem. This problem is typically also called the *allocation* problem.

5.3 Single-Minded Bidders

In this section, we consider the special case of a combinatorial auction, where all bidders are said to be *single-minded*. More precisely, we say that player i is single-minded if there is some (private) set $\Sigma_i \subseteq M$ and a (private) value $\theta_i \geq 0$ such that for every $T \subseteq M$,

$$v_i(T) = \begin{cases} \theta_i & \text{if } T \supseteq \Sigma_i \\ 0 & \text{otherwise.} \end{cases}$$

Intuitively, player i is only interested in getting the items in Σ_i (or some more) and its valuation for these items is θ_i.

Note that in the single-minded case, the first source of inefficiency mentioned above vanishes since now every player simply reports a pair (S_i, b_i) (not necessarily equal to (Σ_i, θ_i)) to the auctioneer. Nevertheless, the second source of inefficiency remains, as will be proven below.

The allocation problem for the single-minded case is as follows: Given the bids $\{(S_i, b_i)_{i \in N}\}$, determine a subset $W \subseteq N$ of *winners* such that $S_i \cap S_j = \emptyset$ for every $i, j \in W$, $i \neq j$, with maximum social welfare $\sum_{i \in W} b_i$.

Theorem 5.3. *The allocation problem for single-minded bidders is NP-hard.*

Proof. We give a polynomial-time reduction from the NP-complete problem *independent set*. The independent set problem is as follows: Given an undirected graph $G = (V, E)$ and a non-negative integer k, determine whether there exists an independent set of size k.[3]

Given an instance (G, k) of the independent set problem, we can construct a single-minded combinatorial auction as follows: The set of items M corresponds to the edge set E of G. We associate a player $i \in N$ with every vertex $u_i \in V$ of G. The bundle that

[3] Recall that an independent set $I \subseteq V$ of G is a subset of the vertices such that no two vertices in I are connected by an edge.

player i desires corresponds to the set of all adjacent edges, i.e., $S_i := \{e = \{u_i, u_j\} \in E\}$, and the value that i assigns to its bundle S_i is $b_i = 1$.

Now observe that a set $W \subseteq N$ of winners satisfies $S_i \cap S_j = \emptyset$ for every $i \neq j \in W$ iff the set of vertices corresponding to W constitute an independent set in G. Moreover, the social welfare obtained for W is exactly the size of the independet set. □

Given the above hardness result and insisting on polynomial-time computability, we are thus forced to consider approximation algorithms. The idea is to relax the efficiency condition and to ask for an outcome that is (only) approximately efficient. We call a mechanism *α-efficient* for some $\alpha \geq 1$ if it computes an allocation $a \in O$ (assuming truthful bidding $(S_i, b_i) = (\Sigma_i, \theta_i)$ for all $i \in N$) such that

$$\sum_{i \in N} v_i(a) \geq \frac{1}{\alpha} \max_{a \in O} \sum_{i \in N} v_i(a).$$

The proof of Theorem 5.3 even shows that the reduction is *approximation preserving*. That is, it specifies a bijection that preserves the objective function values of the corresponding solutions (of the allocation problem and the independent set problem). It is known that the independent set problem is hard even from an approximation point of view:

Fact 5.1. *For every fixed $\varepsilon > 0$, there is no $O(n^{1-\varepsilon})$-approximation algorithm for the independent set problem, where n denotes the number of vertices in the graph (unless $NP \subseteq ZPP$).*

Since the number of edges in a (simple) directed graph is at most $O(n^2)$, we obtain the following corollary:

Corollary 5.1. *For every fixed $\varepsilon > 0$, there is no $O(m^{1/2-\varepsilon})$-efficient mechanism for single-minded bidders, where m denotes the number of items (unless $NP \subseteq ZPP$).*

5.4 Generalized Second-Price Auction

The main revenue of search engines like Google or Yahoo! comes from sponsored search auctions, also called *ad-auctions*. When a user enters a query into a search engine, these auctions allocate a limited number of advertisement slots to advertisers that have bid for keywords contained in the query. Advertisers are then charged per click if the user follows the link displayed in the advertisement. There are different advertisement slots and, typically, advertisers prefer slots that are at the top of the page (because users more likely click on it) rather than the ones at the bottom of the page.

An instance of the *ad-auction* problem is given as follows:

- A set $N = [n]$ of players (advertisers) is interested in advertising using one of m many available slots.
- Every player $i \in N$ has a valuation $v_i \geq 0$ which represents his (private) value per click.

Algorithmus 4 GSP mechanism

1: Collect the bids b_i of all players $i \in N$.

2: Order players according to non-decreasing bids (ties are broken arbitrarily). Let $\pi(k)$ be the player with the k-th highest bid, i.e.,

$$b_{\pi(1)} \geq b_{\pi(2)} \geq \dots b_{\pi(n)}$$

3: Player $\pi(k)$ is allocated to slot k and receives α_k clicks. For each such click, player $\pi(k)$ pays the next highest bid, i.e., $b_{\pi(k+1)}$.

- Every player $i \in N$ submits a bid $b_i \geq 0$ which expresses the maximum amount he is willing to pay per click.
- Every slot $k \in [m]$ has a *click-through rate* α_k which represents the number of clicks that the player being assigned to slot k can expect.

We make a few assumptions without loss of generality.

1. We assume that $n = m$. If there are less than n slots, we add artificial slots with click-through rate zero. Similarly, if there are less than m players, we add artificial players that have valuation 0.
2. We assume without loss of generality that the players are numbered such that $v_1 \geq v_2 \geq \dots \geq v_n$.
3. We assume without loss of generality that $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_n$.

One of the most successful auction mechanisms for ad-auctions is the so-called *generalized second-price auction (GSP)*. GSP proceeds as described in Algorithm 4.

Define the utility of a player i when being allocated to slot k as

$$u_i(b) = \alpha_k(v_i - b_{\pi(k+1)}).$$

Let the *social welfare* of assignment π be defined as

$$\sum_{k=1}^{n} \alpha_k v_{\pi(k)}.$$

Note that by the ordering of the players and the click-through rates, the social optimum of the game is $\sum_{i=1}^{n} \alpha_i v_i$.

The strategy profile $b = (b_1, \dots, b_n)$ is a pure Nash equilibrium if for every player $i \in N$:

$$u_i(b_i, b_{-i}) \geq u_i(b_i', b_{-i}) \quad \forall b_i' \geq 0.$$

5.4.1 Examples

As we will see in the examples below, GSP has a number of pathologies that VCG was designed to avoid:

1. Truth-telling is not necessarily an equilibrium.
2. Equilibria are not unique.
3. Equilibria do not necessarily correspond to socially optimal outcomes.

Example 5.1 (Truth-telling might not be an equilibrium)**.** Consider the case of three players with valuations per click $v_1 = 7$, $v_2 = 6$ and $v_3 = 1$. Let the click-through rates of the three slots be $\alpha_1 = 10$, $\alpha_2 = 4$ and $\alpha_3 = 0$. Suppose every player bids his true valuations. The slots 1, 2, 3 are then assigned players 1, 2 and 3, respectively. The utility of player 1 is $\alpha_1(v_1 - b_{\pi(2)}) = 10(7-6) = 10$. Suppose player 1 shades his bid and bids 5 instead of 7. Then the slots 1, 2, 3 are assigned to players 2, 1 and 3, respectively. As a consequence, player 1 experiences a utility of $\alpha_2(v_1 - b_{\pi(3)}) = 4(7-1) = 24 > 10$.

Example 5.2 (Socially optimal equilibrium)**.** Consider the same example as above but assume that players bid $b_1 = 5$, $b_2 = 4$ and $b_3 = 2$. The slots 1, 2, 3 are assigned to players 1, 2 and 3, respectively. Player 3's utility is zero and it remains zero unless he bids more than $b_2 = 4$. But then his utility becomes negative because $v_3 = 1 < 4 = b_2$. Consider player 1. His utility is $\alpha_1(v_1 - b_2) = 10(7-4) = 30$. If he lowers his bid below b_2, his utility changes to $\alpha_2(v_1 - b_3) = 4(7-2) = 20$. If he lowers it below b_1, his utility changes to 0. Similarly, player 2's utility is $\alpha_2(v_2 - b_3) = 4(6-2) = 16$. If he raises his bid above b_1, his utility changes to $\alpha_1(v_2 - b_1) = 10(6-5) = 10$. If he lowers his bid below b_3 his utility becomes 0. Note that this equilibrium also corresponds to a social optimum of social welfare $70 + 24 = 94$.

Example 5.3 (Inefficient equilibrium)**.** Consider the same example as above but assume that players bid $b_1 = 3$, $b_2 = 5$ and $b_3 = 1$. The slots 1, 2, 3 are assigned to players 2, 1 and 3, respectively. Player 3's utility is zero and it remains zero unless he bids more than $b_1 = 4$. But then his utility becomes negative because $v_3 = 1 < 4 = b_1$. Consider player 2. His utility is $\alpha_1(v_2 - b_1) = 10(6-3) = 30$. If he lowers his bid below b_1, his utility changes to $\alpha_2(v_2 - b_3) = 4(6-1) = 20$. If he lowers it below b_3, his utility changes to 0. Similarly, player 1's utility is $\alpha_2(v_1 - b_3) = 4(7-1) = 24$. If he raises his bid above b_2, his utility changes to $\alpha_1(v_1 - b_2) = 10(7-5) = 20$. If he lowers his bid below b_3 his utility becomes 0. Note that this equilibrium does not correspond to a social optimum: The social welfare of this equilibrium is $60 + 28 = 88 < 94$.

5.4.2 Existence of Equilibria

The general existence proof of equilibria for GSP is based on the concept of *matching markets*: A *matching market* $(B, S, (v_{ik}))$ is given as follows:

- A set B of n buyers and a set S of n sellers, each selling a house (say).

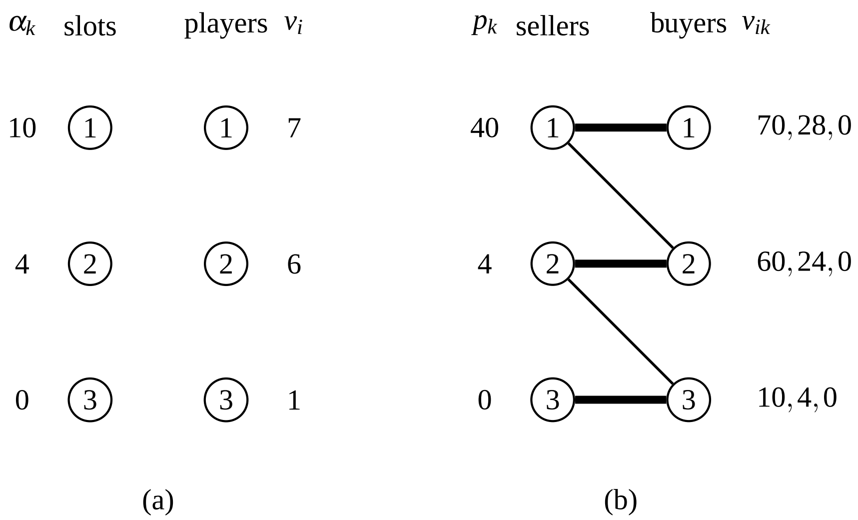

Figure 8: Ad-auction instance (a) and its corresponding matching market with market-clearing prices (b).

- Every buyer $i \in B$ has an integer *valuation* $v_{ik} \geq 0$ for the house sold by $k \in S$.

Let p_k be the price that seller $k \in S$ announces for his house. If buyer i buys the house from seller k his payoff is $\max\{v_{ik} - p_k, 0\}$. (We assume that a buyer is not interested in receiving the house if his payoff is negative.) The set of sellers that maximize buyer i's payoff are called the *preferred sellers of i* and are denoted by $P(i)$.

Definition 5.1. Suppose we are given a matching market $(B, S, (v_{ik}))$ together with some prices $(p_k)_{k \in S}$. We define the *preferred seller graph G* as follows: $G = (B \cup S, E)$ is a bipartite graph with the set of buyers on one side and the set of sellers on the other side of the bipartition. The set of edges E contains an edge $\{i, k\}$ for every buyer $i \in B$ and every preferred seller $k \in P(i)$ of i.

Informally, a set of prices are *market-clearing* if every buyer is assigned to a seller that is also one of his preferred sellers. That is, no buyer has an incentive to disagree with the assignment.

Definition 5.2. Given a matching market $(B, S, (v_{ik}))$, we say that the prices $(p_k)_{k \in S}$ are *market-clearing* if the preferred-seller graph has a perfect matching.

We will use the following theorem without proof.

Theorem 5.1. *Given a matching market $(B, S, (v_{ik}))$ there always exists a set of prices $(p_k)_{k \in S}$ that are market-clearing. Moreover, a matching in the preferred-seller graph has maximum total valuation among all possible assignments of buyers to sellers.*

We will exploit this result in order to define extract bids that constitute an equilibrium with respect to GSP. Moreover, this equilibrium will be a social optimum.

Theorem 5.2. *Pure Nash equilibria exist for GSP. Moreover, there always exists a pure Nash equilibrium that is a social optimum.*

Proof. Given an instance of the ad-auction, we can formulate a matching market as follows: The buyers correspond to the players and the sellers correspond to the slots. Each buyer has a valuation $v_{ik} = v_i \alpha_k$ for the slot (house) of seller $k \in S$. Let $(p_k)_{k \in S}$ be some market-clearing prices for the resulting matching market. Note that by Theorem 5.1 such prices exist. Note also that the resulting assignment maximizes the social welfare. Our understanding is that the price p_k represents the *cumulative* price for all clicks of slot k. We can easily extract the price per click p_k^* from this by defining $p_k^* = p_k / \alpha_k$ for every $k \in [n]$.

We first argue that $p_1^* \geq p_2^* \geq \cdots \geq p_n^*$. Consider two slots j and k with $j < k$. Because prices are market clearing, player k prefers slot k over j. The payoff of player k is equal to $\alpha_k(v_k - p_k^*)$. In slot j his payoff would be $\alpha_j(v_k - p_j^*)$. Now $\alpha_j \geq \alpha_k$ and the fact that player k prefers slot k over j thus implies that $v_k - p_j^* < v_k - p_k^*$, or, equivalently, $p_j^* \geq p_k^*$.

Now construct the bids as follows: Player i with $i > 1$ bids $b_i = p_{i-1}^*$ and player 1 bids any value larger than p_1^*. Note that with these bids GSP assigns slots $1, 2, \ldots, n$ to players $1, 2, \ldots, n$ and charges prices $p_1^*, p_2^*, \ldots, p_n^*$ as desired.

We argue that these bids constitute a pure Nash equilibium. Consider a player i. If player i lowers his bid just below the current bid of player $k > i$, he gets slot k instead. But because the prices are market-clearing i prefers its current slot at its current price over k's slot at k's current price. Suppose player i raises his bid, say just above the current bid of player $k < i$. He would then have to pay the bid of player k for slot k, which is larger than what he would currently pay for slot k. Because prices are market-clearing, player i does not prefer to be assigned to slot k under the current pricing. Therefore, he is clearly not interested in being assigned to slot k for an even higher price. We conclude that this set of bids $(b_i)_{i \in N}$ constitutes a pure Nash equilibrium with respect to GSP. □

5.4.3 Price of Anarchy

It is not hard to see that the price of anarchy of GSP can be unbounded in general.

Example 5.4. Consider $n = 2$ players with valuations $v_1 = 1$ and $v_2 = 0$ and two slots with click-through rates $\alpha_1 = 1$ and $\alpha_2 = 0$, respectively. It is easy to verify that $b_1 = 0$ and $b_2 = 1$ is a pure Nash equilibrium where players 1 and 2 are assigned to slots 2 and 1, respectively. The social welfare of this equilibrium is 0 while the social optimum is 1. Thus the price of anarchy is unbounded.

Note, however, that the bidding above is not very natural because player 2 is exposed to obtain a negative utility. To see this imagine that another player would enter the game and bid anything in the interval $(0, 1)$. Then this would result in a negative utility for player 2 under GSP. More generally, for every player $i \in N$ bidding $b_i > v_i$ is dominated by bidding v_i if GSP is used. This leads to the natural assumption that players are *conservative*, i.e., they do not overbid their actual valuations.

The following recent result shows that the price of anarchy for pure Nash equilibria of GSP is bounded by $\frac{1+\sqrt{5}}{2}$ if one assumes that players are conservative.

Theorem 5.3. *For conservative bidders, the price of anarchy for pure Nash equilibria of GSP is at most the golden ratio* $\frac{1+\sqrt{5}}{2} \approx 1.618$.