Notes on Algorithmic Game Theory

Thomas Archbold

1 Tragedy of the Commons

- n players sharing a common resource of limited capacity, normalized e.g. to 1
- each player i picks their own level of usage $x_i \in [0, 1]$
- the quality of the resource deteriorates with increasing usage

Together the players will pick a vector $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, indicating each player's usage. Denote total load $\mathcal{L} = \sum_j x_j$, and hence spare capacity $= 1 - \mathcal{L}$.

Denote by $u_i(\boldsymbol{x})$ each player's satisfaction/utility of their allocation with respect to each other player's use of the resource. If the total load is greater than 1, no player gets any satisfaction. Else, the satisfaction for player i decreases with increasing total usage, but increases with his own usage (up to a point).

$$u_i(\boldsymbol{x}) = \begin{cases} x_i(1 - \mathcal{L}) & \mathcal{L} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

1.1 Socially optimal solutions

Denote by $u(\boldsymbol{x})$ the total satisfaction, i.e. the sum of each player's satisfaction according to the allocation vector $\boldsymbol{x}$:

$$\begin{aligned} u(\boldsymbol{x}) &:= \sum_j u_j(\boldsymbol{x}) \\ &= \sum_j x_j(1 - \mathcal{L}) \\ &= (1 - \mathcal{L}) \sum_j x_j \\ &= (1 - \mathcal{L})\mathcal{L} \\ &= \mathcal{L} - \mathcal{L}^2 \end{aligned} \tag{2}$$

Define $u(\mathcal{L}) := \mathcal{L} - \mathcal{L}^2$, i.e. the total satisfaction as a function of the total load.

A socially optimal outcome is achieved by maximising this function, so $\frac{d}{d\mathcal{L}} u(\mathcal{L}) = -2\mathcal{L} + 1$, which is maximised by $\mathcal{L}^* = \frac{1}{2}$. Therefore, a benevolent dictator would assign each player an allocation of the shared resource such the sum of all the players' allocations is $\frac{1}{2}$. Note that the specific allocation to each player doesn't matter, only that the total load is $\frac{1}{2}$.

The total satisfaction of this socially optimal solution is $u(\mathcal{L}^*) = u(\frac{1}{2}) = \frac{1}{2} - (\frac{1}{2})^2 = \frac{1}{4}$. But is this solution stable?

1.2 The stable solution

Self-interest assumption: each player strives to maximise $x_i(1-\mathcal{L})$. With this in mind, define $\mathcal{L}_{-i} := \sum_{j\neq i} x_j = (\sum_j x_j) - x_i$. Assuming $\mathcal{L} < 1$, then:

$$\begin{aligned} u_i(\boldsymbol{x}) &= x_i(1-\mathcal{L}) \\ &= x_i(1-(\mathcal{L}_{-i}+x_i)) \\ &= x_i(1-\mathcal{L}_{-i}-x_i) \end{aligned} \tag{3}$$

Now define $u_i(x_i) := x_i(1-\mathcal{L}_{-i}-x_i)$.

Claim 1.1. *$u_i(x_i)$ is maximised by $x_i^* = \frac{1-\mathcal{L}_{-i}}{2}$*

Proof. $\frac{d}{dx_i}u_i(x_i) = 1-\mathcal{L}_{-i}-2x_i$, which is maximised at $x_i^* = \frac{1-\mathcal{L}_{-i}}{2}$ □

Definition 1.2 (Stable solution)**.** A solution $\boldsymbol{x} = (x_1, \ldots, x_n)$ of the Tragedy of the Commons game is stable if no player wants to deviate from it. That is, for all i, x_i is the best response to $\boldsymbol{x}_{-i}$, where $\boldsymbol{x}_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$.

Claim 1.3. *The only stable solution in the Tragedy of the Commons game is $\boldsymbol{x} = (\frac{1}{n+1}, \ldots, \frac{1}{n+1})$*

Proof. The solution is stable when no player wishes to deviate. As $u_i(x_i)$ has one maximum, each player chooses their allocation as $x_i = \frac{1-\mathcal{L}_{-i}}{2}$. The total load $\mathcal{L}$ under this allocation is:

$$\sum_i x_i = \sum_i \frac{1}{2}(1-\mathcal{L}_{-i}) = \sum_i \frac{1}{2}(1-(\mathcal{L}-x_i))$$

Hence:

$$2\mathcal{L} = \sum_i 1 - \sum_i \mathcal{L} + \sum_i x_i = n - n\mathcal{L} + \mathcal{L}$$

So $\mathcal{L} = \frac{n}{n+1} \Rightarrow x_i = \frac{1}{n+1}$ for all i. □

1.3 The tragedy

The total satisfaction of a socially optimal solution $u(\boldsymbol{x}^*) = \sum_i u_i(\boldsymbol{x}^*) = \sum_i x_i(1-\frac{1}{2}) = \frac{1}{2}\sum_i x_i = \frac{1}{4}$. The spare capacity of the shared resource in the stable (self-interested) solution is $1-\mathcal{L} = 1-\frac{n}{n+1} = \frac{1}{n+1}$, so the total satisfaction in the stable solution is:

$$\begin{aligned} \sum_i x_i\left(\frac{1}{n+1}\right) &= \sum_i \frac{1}{n+1}\frac{1}{n+1} \\ &= \frac{n}{(n+1)^2} < \frac{1}{n} \end{aligned} \tag{4}$$

The tragedy is that not only is the total utility strictly smaller than for a socially optimal solution, but it is also bounded by $\frac{1}{n}$, meaning the more players there are, the less benefit they get.

2 Strategic Form Games

Definition 2.1 (Strategic Form Game)**.** A *strategic form game* consists of:

- Players $1, \ldots, n$
- For each player i, a finite set S_i of strategies
- For each player i, the function $u_i : S_1 \times \ldots \times S_n \to \mathbb{R}$ that maps a *strategy profile* to a real-numbered payoff to player i

A *strategy profile* is a vector of strategies that fully specifies all actions in a game. It must include exactly one strategy for each player. Denote by $\mathcal{S} = \times_i S_i$ the set of all strategy profiles, that is, $\mathcal{S}$ is the set of all possible combinations of players' strategies.

Remark. Each player's utility function is defined as $u_i : \mathcal{S} \to \mathbb{R}$, and not, say, $u_i : S_i \to \mathbb{R}$. That is, the payoff to player i depends both on his own strategy and the strategy chosen by each other player (if the latter were the case, the game would only consist of n separate optimization problems, one for each i).

Playing a game consists of:

1. simultaneously[1], each player i chooses a strategy $s_i \in S_i$ that strives to maximise their own utility (**self-interest assumption**)
2. each player i receives utility $u_i(s_1, \ldots, s_n)$

Definition 2.2 (Strict dominance)**.** A strategy $s \in S_i$ *strictly dominates* another strategy $t \in S_i$ if for all strategy profiles $s_{-i} \in S_{-i}$:

$$u_i(s_{-i}, s) > u_i(s_{-i}, t) \tag{5}$$

Fact. If strategy profiles $s, t \in S_i$ are such that s strictly dominates t in game Γ, then the games Γ and $\Gamma \setminus t$ are equivalent.

Definition 2.3 (Dominance-solvability)**.** A game is *dominance solvable* if Iterated Elimination of Strictly Dominated Strategies (IESDS) yields a unique strategy profile (i.e. a $1 \times 1 \times \ldots$ game).

Fact. The outcome of IESDS is always unique, meaning the order of elimination does not matter.

Fact. Not all games are dominance-solvable.

3 Best responses and Nash Equilibria

Definition 3.1 (Best response)**.** A strategy $s^* \in S_i$ is a *best response* of player i to a strategy profile s_{-i} if for every other strategy $t \in S_i$:

$$u_i(s_{-i}, s^*) \geq u_i(s_{-i}, t) \tag{6}$$

s^* is therefore the best choice for player i given that the other players are playing strategies in s_{-i}. Player i has no incentive to UNILATERALLY DEVIATE from s^*.

Alternatively and equivalently:

$$u_i(s_{-i}, s^*) = \max\{u_i(s_{-i}, t) \mid t \in S_i\} = \max_{t \in S_i} u_i(s_{-i}, t) \tag{7}$$

Definition 3.2 (Pure Strategy Nash Equilibrium)**.** A strategy profile $s = (s_1, \ldots, s_n)$ is a Pure Strategy Nash Equilibrium (PNE) if for every player i, the strategy s_i is a best response to s_{-i}.

In other words, in $s = (s_1, \ldots, s_n)$ no player has an incentive to UNILATERALLY DEVIATE. A PNE is a strategy profile containing only best responses for each player.

Definition 3.3 (Mixed strategy)**.** A *mixed strategy* for player i is a probability distribution over the set of pure strategies: $\sigma_i = (p(s_1), \ldots, p(s_{|S_i|})) \in \Delta^{S_i}$.

The set of probability distributions $\Delta^{S_i} := \{(x_1, \ldots, x_{|S_i|}) : \sum_j^{|S_i|} x_j = 1 \wedge x_1, \ldots, x_{|S_i|} \geq 0\}$ defines all probability distributions for player i's strategies. A probability distribution σ_i is a vector detailing the probabilities of player i choosing each strategy $s \in S_i$, e.g. $\sigma_{\mathrm{I}} = (\frac{1}{3}, 0, \frac{2}{3})$.

Definition 3.4 (Expected utility)**.** Utility functions generalise from pure strategy profiles to mixed strategy profiles; the expected utlity for player i under strategy profile $\sigma = (\sigma_1, \ldots, \sigma_n)$ is given by:

$$u_i(\sigma_1, \ldots, \sigma_n) = \sum_{(S_1, \ldots, S_n) \in \mathcal{S}} \prod_j \sigma_j(s_j)\, u_i(s_1, \ldots, s_n) \tag{8}$$

[1] i.e., players do not know what everyone else is doing as they are doing it

Definition 3.5 (Mixed strategy best response)**.** A mixed strategy σ_i is a best response of player i to a mixed strategy profile σ_{-i} if:

$$u_i(\sigma_{-i}, \sigma_i) = \max_{\tau \in \Delta^{S_i}} u_i(\sigma_{-i}, \tau) \tag{9}$$

Definition 3.6 (Mixed strategy Nash Equilibrium)**.** Strategy profile $\sigma = (\sigma_1, \ldots, \sigma_n)$ is a *Mixed Nash Equilibrium* (MNE) if for every i, the strategy σ_i is a best response to σ_{-i}.

Hence in a MNE, no player has an incentive to unilaterally deviate.

4 Finding Mixed Equilibria in 2×2 2-player games

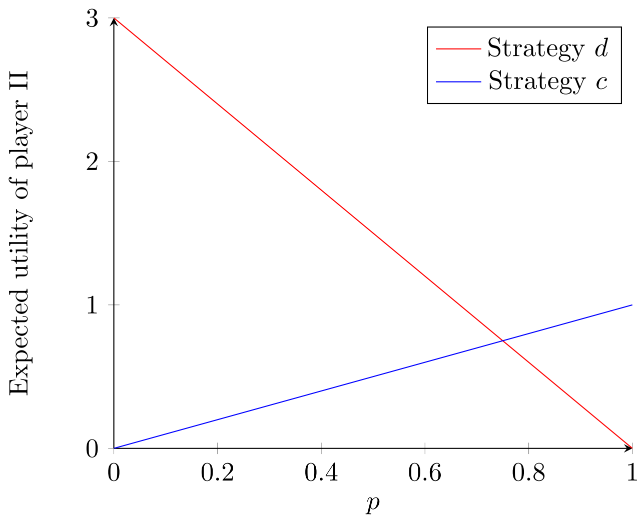
Suppose player I and II have two pure strategies $S_{\mathrm{I}} = \{A, B\}$ and $S_{\mathrm{II}} = \{c, d\}$ respectively, and mixed strategies $\sigma_{\mathrm{I}} = (p, 1-p)$ and $\sigma_{\mathrm{II}} = (q, 1-q)$. That is, for example, player I plays strategy A with probability p and hence strategy B with probability $1-p$ (and similarly for player II). Consider to following payoff matrix:

I, II	c	d
A	3,1	0,0
B	0,0	1,3

To compute the mixed equilibria of this game:

1. Draw the diagram of best pure responses of player II to the mixed strategy $\sigma_{\mathrm{I}} = (p, 1-p)$ of player I.
2. Compute $p^* \in [0, 1]$ which makes both pure strategies c and d best responses to σ_{I} [2].
3. Repeat for player I

In the above example, playing strategy c player II gets expected payoff that ranges from 0 (at $p = 0$) to 1 (at $p = 1$). For strategy d player II's expected payoff ranges from 3 (at $p = 0$) to 0 (at $p = 1$). The resulting lines are shown in Fig. 4 – these are straight lines due to linearity of expectation.



Think of it this way: if player I plays strategy A with increasing probability, and player II always plays c, player II's utility will increase from 0 to 1. Similarly, if player I plays strategy A with increasing probability, and player II always plays d, players II's utility will fall from 3 to 0.

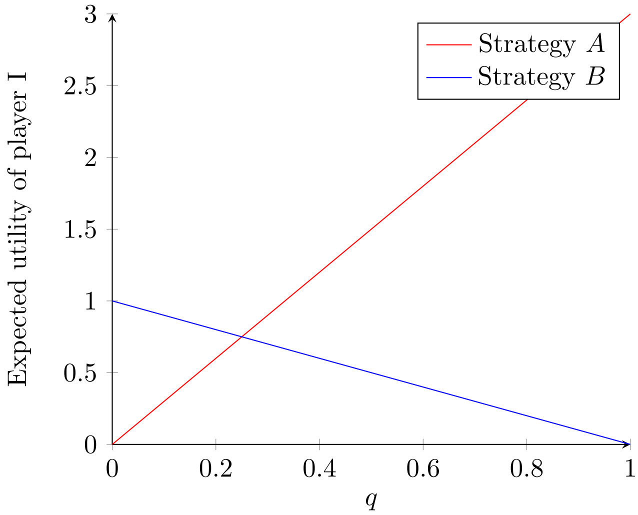
These lines meet at point p^*, at which point player II is indifferent to his choice of strategy (both are best responses).

[2]Equivalently, find the mixed strategy $(p^*, 1-p^*)$ which makes player II *indifferent* between c and d.

For step 2, we have:

$$
\begin{aligned}
\mathbb{E}[u_{\mathrm{I}}((p^*, 1-p^*), c) &= \mathbb{E}[u_{\mathrm{I}}((p^*, 1-p^*), d) \\
1p^* + 0(1-p^*) &= 0p^* + 3(1-p^*) \\
p^* &= \frac{3}{4}
\end{aligned}
$$

Repeating for player I, we obtain the following graph:



Again: if player II plays c with increasing probability and I always plays A, I's utility will increase from 0 to 3. If player II plays c with increasing probability and I always plays B, I's utility will decrease from 1 to 0.

These lines meet at q^*. Similarly to before, we can compute q^* as before. Note that:

$$
\begin{aligned}
\mathbb{E}[u_{\mathrm{I}}(A, (q^*, 1-q^*))] &= \mathbb{P}r[A]\mathbb{P}r[c]\, u_{\mathrm{I}}(A, c) + \mathbb{P}r[A]\mathbb{P}r[d]\, u_{\mathrm{I}}(A, d) + \mathbb{P}r[B]\mathbb{P}r[c]\, u_{\mathrm{I}}(B, c) + \mathbb{P}r[B]\mathbb{P}r[d]\, u_{\mathrm{I}}(B, d) \\
&= \mathbb{P}r[A]\mathbb{P}r[c]\, u_{\mathrm{I}}(A, c) + \mathbb{P}r[A]\mathbb{P}r[d]\, u_{\mathrm{I}}(A, d)
\end{aligned}
\tag{10}
$$

The second line comes about as the probability of choosing B in the mixed strategy $(1, 0)$ is 0, so we can disregard terms involving choosing B:

$$
\begin{aligned}
\mathbb{E}[u_{\mathrm{I}}(A, (q^*, 1-q^*)) &= \mathbb{E}[u_{\mathrm{I}}(B, (q^*, 1-q^*)) \\
3q^* &= 1(1-q^*) \\
q^* &= \frac{1}{4}
\end{aligned}
$$

Therefore, if such $p^*, q^* \in (0, 1)$ exist, then the mixed strategy profile $((p^*, 1-p^*), (q^*, 1-q^*))$ is a truly mixed-strategy Nash Equilibrium.

5 Vector-matrix notation for 2-player games

Let the payoffs to player I be denoted by the matrix $\boldsymbol{A}$, where entry $\boldsymbol{A}_{ij}$ gives the payoff to player I when playing strategy $i \in S_{\mathrm{I}}$ given that player II is playing strategy $j \in S_{\mathrm{II}}$. Define player II's payoff matrix $\boldsymbol{B}$ in a similar fashion, where the rows still represent player I's strategies and the columns player II's.

Let $\boldsymbol{x} \in \Delta^{m_1}$ denote player I's mixed strategy, and $\boldsymbol{y} \in \Delta^{m_2}$ denote player II's. Then player I's expected payoff from playing pure strategy $i \in S_{\mathrm{I}}$ is given by the entry $(\boldsymbol{Ay})_i$, and player II's expected payoff from playing pure strategy $j \in S_{\mathrm{II}}$ is given by $(\boldsymbol{x}^\top \boldsymbol{B})_j$.

Note when computing player II's expected utility each entry in the resulting vector is for some fixed column, while when computing player I's then each entry is for some fixed row.

5.1 Example

Suppose we have the following two-player zero-sum (i.e. $\boldsymbol{B} = -\boldsymbol{A}$) game:

$$\boldsymbol{A} = \begin{pmatrix} 28 & 1 & -38 & -11 \\ 4 & 3 & 2 & -3 \\ 5 & -3 & 4 & 3 \\ -19 & -9 & 29 & 1 \end{pmatrix}, \boldsymbol{x} = \begin{pmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{2} \\ 0 \end{pmatrix}, \boldsymbol{y} = \begin{pmatrix} 0 \\ \frac{1}{2} \\ 0 \\ \frac{1}{2} \end{pmatrix}$$

In this example, player I's expected utilities $\boldsymbol{A}\boldsymbol{y} = \begin{pmatrix} -5 & 0 & 0 & 4 \end{pmatrix}$ and player II's expected utilities $\boldsymbol{x}^\top \boldsymbol{B} = \begin{pmatrix} -\frac{9}{2} & 0 & -3 & 0 \end{pmatrix}$. Hence pure strategies 2 and 3 are best responses for player I to $\boldsymbol{y}$ (as they yield the highest expected utility, 0), and furthermore player I is indifferent to them. Similarly pure strategies 2 and 4 are best responses for player II to $\boldsymbol{x}$, and is also indifferent to them.

Note that:

$$\boldsymbol{x} = \begin{pmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{2} \\ 0 \end{pmatrix}, \boldsymbol{A}\boldsymbol{y} = \begin{pmatrix} -5 \\ 0 \\ 0 \\ -4 \end{pmatrix}$$

Thus $\boldsymbol{x}$ is a best response to $\boldsymbol{y}$ (strategies 2 and 3 for I are best responses to $\boldsymbol{y}$, and player I mixes between strategies 2 and 3). Similarly:

$$\boldsymbol{y} = \begin{pmatrix} 0 \\ \frac{1}{2} \\ 0 \\ \frac{1}{2} \end{pmatrix}, \boldsymbol{x}^\top \boldsymbol{B} = \begin{pmatrix} -\frac{9}{2} \\ 0 \\ -3 \\ 0 \end{pmatrix}$$

Hence $\boldsymbol{y}$ is a best response to $\boldsymbol{x}$ (strategies 1 and 3 are best responses, and II mixes between strategies 1 and 3).

6 Linear Programming and LP Duality

A linear program is a linear optimisation problem – we are maximising (minimising) a linear objective function, subject to a set of linear constraints (inequalities).

Definition 6.1 (Linear Program)**.** Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and vectors $\boldsymbol{b} \in \mathbb{R}^m, \boldsymbol{c} \in \mathbb{R}^n$, we want to find a vector $\boldsymbol{x} \in \mathbb{R}^n$ such that:

$$\begin{aligned} \boldsymbol{x} \in \arg\max \boldsymbol{c}^\top \boldsymbol{x} \\ \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ \boldsymbol{x} \geq \boldsymbol{0} \end{aligned} \tag{11}$$

6.1 Taking the Dual

There are four steps to taking the dual: swap maximise with minimise, swap vectors $\boldsymbol{b}$ and $\boldsymbol{c}$, transpose $\boldsymbol{A}$, and change the direction of the (non-trivial) inequalities. In the dual there is one variable for every constraint, and one constraint for every variable. Hence for a primal LP with n variables and m constraints, the dual LP will have m variables and n constraints.

Consider the primal linear program, P:

$$\begin{aligned} \text{maximise } \boldsymbol{c}^\top \boldsymbol{x} \text{ subject to } \boldsymbol{A}\boldsymbol{x} &\leq \boldsymbol{b} \\ \boldsymbol{x} &\geq \boldsymbol{0} \end{aligned} \tag{12}$$

The dual of the linear program P, D, is as follows:

$$\begin{aligned} \text{minimise } \boldsymbol{b}^\top \boldsymbol{y} \text{ subject to } \boldsymbol{A}^\top \boldsymbol{y} &\geq \boldsymbol{c} \\ \boldsymbol{y} &\geq \boldsymbol{0} \end{aligned} \tag{13}$$

Theorem 6.2 (Strong Duality)**.** *If P or D has an optimal solution of finite value, then so too does the other, and the value of the objective function of the optimal solutions are equal.*

7 Two-player zero-sum games

7.1 Minimax and maximin

Consider a zero-sum game determined by the matrix $\boldsymbol{A} \in \mathbb{R}^{m_1 \times m_2}_{\geq 0}$ which gives the payoffs to player I. Since the game is zero-sum means that the payoff matrix for player II $\boldsymbol{B} = -\boldsymbol{A}$.

Definition 7.1 (Maximin value)**.** The maximin value is the highest payoff a player can be sure to receive without knowing the actions of the other players. Equivalently, this is the lowest value other players can force the player to receive when they know the player's action:

$$\underline{v_i} = \max_{s_i} \min_{s_{-i}} u_i(s_i, s_{-i}) \tag{14}$$

Definition 7.2 (Minimax value)**.** The minimax value is the smallest value the other players can force the player to receive, without knowing the player's actions. Equivalently, it is the largest value the player can be sure to get when they know the actions of the other players:

$$\overline{v_i} = \min_{s_{-i}} \max_{s_i} u_i(s_i, s_{-i}) \tag{15}$$

Consider the maximin values of the players in a two-player zero-sum game. Since $u_{\mathrm{I}} + u_{\mathrm{II}} = 0$, we can denote u as the payoff to player I and $-u$ as the payoff to player II. Player I's maximin is:

$$\underline{v_{\mathrm{I}}} = \max_{s_{\mathrm{I}} \in S_{\mathrm{I}}} \min_{s_{\mathrm{II}} \in S_{\mathrm{II}}} u(s_{\mathrm{I}}, s_{\mathrm{II}})$$

Player II's maximin value is:

$$\underline{v_{\mathrm{II}}} = \max_{s_{\mathrm{II}} \in S_{\mathrm{II}}} \min_{s_{\mathrm{I}} \in S_{\mathrm{I}}} -u(s_{\mathrm{I}}, s_{\mathrm{II}}) = -\min_{s_{\mathrm{II}} \in S_{\mathrm{II}}} \max_{s_{\mathrm{I}} \in S_{\mathrm{I}}} u(s_{\mathrm{I}}, s_{\mathrm{II}})$$

Definition 7.3 (Maximin and minimax value in two-player zero-sum games)**.** The *maximin value* $\underline{v}$ is the minimum that player I can guarantee themselves to get. The *minimax value* $\overline{v}$ is the maximum that player II can guarantee they will pay to player I:

$$\begin{aligned} \underline{v} &= \max_{s_{\mathrm{I}} \in S_{\mathrm{I}}} \min_{s_{\mathrm{II}} \in S_{\mathrm{II}}} u(s_{\mathrm{I}}, s_{\mathrm{II}}) \\ \overline{v} &= \min_{s_{\mathrm{II}} \in S_{\mathrm{II}}} \max_{s_{\mathrm{I}} \in S_{\mathrm{I}}} u(s_{\mathrm{I}}, s_{\mathrm{II}}) \end{aligned} \tag{16}$$

In other words:

- maximin value: "maximum of row minima"
- minimax value: "minimum of column maxima"

Example: Consider the following zero-sum game:

	L	R
T	-2	5
B	3	0

We have:

$$\underline{v} = \max\{-2, 0\} = 0$$
$$\overline{v} = \min\{3, 5\} = 3$$

The maximin strategy for player I is the strategy guaranteeing the maximin value for player I, while the minimax strategy for player II is the strategy guaranteeing the minimax value for player II. In the above example, player I's maximin strategy is B and player II's minimax strategy is L.

The minimax and maximin values may be equal or, as the example indicates, different. It must however be the case that $\underline{v} \leq \overline{v}$, since the highest amount that player I can get must be at most the most that player II will ever pay out to player I.

Fact. Maximin value is at most the minimax value: $\underline{v} \leq \overline{v}$

Definition 7.4 (Value of a game)**.** A two-player game has a *value* if $\overline{v} = \underline{v}$. In this case the quantity $v := \overline{v} = \underline{v}$ is called the *value of the game.* Any maximin and minimax strategies of player I and player II respectively are called *optimal strategies.*

Definition 7.5 (Maximin)**.** A strategy $\boldsymbol{x}^* \in \Delta^{m_1}$ for player I is a *maximin* strategy if

$$\boldsymbol{x}^* \in \arg\max_{\boldsymbol{x} \in \Delta^{m_1}} \left[\min_{j \in [m_2]} (\boldsymbol{x}^\top \boldsymbol{A})_j \right] \tag{17}$$

Definition 7.6 (Minimax)**.** A strategy $\boldsymbol{y}^* \in \Delta^{m_2}$ for player II is a *minimax* strategy if

$$\boldsymbol{y}^* \in \arg\min_{\boldsymbol{y} \in \Delta^{m_2}} \left[\max_{i \in [m_1]} (\boldsymbol{A}\boldsymbol{y})_i \right] \tag{18}$$

A maximin strategy "maximizes the minimum expected payoff to player I". A minimax strategy "minimises the maximum expected payoff to player II". Maximin and minimax values can be thought of as *safety utilities* – no matter what the other plays, they can set a minimum payoff they will receive.

- MAXIMIN – utility that player I can get if he commits to move first
- MINIMAX – utility that plauer II can limit player I's utility to if player II commits to move first

Theorem 7.7 (von Neumann, 1928)**.** *Let $X \subset \mathbb{R}^n, Y \subset \mathbb{R}^m$ be compact convex sets. If $f : X \times Y \to \mathbb{R}$ is a continuous function that is concave-convex, that is:*

$$f(\cdot, y) : X \to \mathbb{R} \textit{ is concave for fixed } y$$
$$f(x, \cdot) : Y \to \mathbb{R} \textit{ is convex for fixed } x$$

Then we have that:

$$\max_{x \in X} \min_{y \in Y} f(x, y) = \min_{y \in Y} \max_{x \in X} f(x, y) \tag{19}$$

7.2 Two-player games as linear programs

The maximization problem is as follows:

$$\begin{aligned} &\text{maximise}_{\boldsymbol{x}, v} v \text{ subject to } \boldsymbol{x} \in \Delta^{m_1} \\ &\boldsymbol{x}^\top \boldsymbol{A} \geq \begin{pmatrix} v \\ \vdots \\ v \end{pmatrix} \end{aligned} \tag{20}$$

Note that this is not yet a Linear Program due to the first constraint. We can therefore rewrite the optimisation problem as an LP in standard form as:

$$\begin{aligned} \text{maximise}_{\boldsymbol{x},v} v \text{ subject to } & -\boldsymbol{x}^\top \boldsymbol{A} + \begin{pmatrix} v \\ \vdots \\ v \end{pmatrix} \leq 0 \\ & \mathbb{1}^\top \boldsymbol{x} \leq 1 \\ & \boldsymbol{x} \geq 0 \end{aligned} \tag{21}$$

The primal LP formulation is the following:

$$\begin{aligned} \text{maximise}_{\boldsymbol{x},v} (0, \dots, 0, 1) \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1} \\ v \end{pmatrix} \text{ subject to } & \begin{pmatrix} & -\boldsymbol{A}^\top & & 1 \\ & & & \vdots \\ & & & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1} \\ v \end{pmatrix} \leq \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \\ & \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1} \\ v \end{pmatrix} \geq 0 \end{aligned} \tag{22}$$

The dual LP is:

$$\begin{aligned} \text{minimise}_{\boldsymbol{y},w} (0, \dots, 0, 1) \begin{pmatrix} y_1 \\ \vdots \\ y_{m_2} \\ w \end{pmatrix} \text{ subject to } & \begin{pmatrix} & -\boldsymbol{A} & & 1 \\ & & & \vdots \\ & & & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_{m_2} \\ w \end{pmatrix} \geq \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \\ & \begin{pmatrix} y_1 \\ \vdots \\ y_{m_2} \\ w \end{pmatrix} \geq 0 \end{aligned} \tag{23}$$

7.3 LP duality and zero-sum games

We have the primal (P):

$$\begin{aligned} \max_{\boldsymbol{x},v} v \text{ subject to } & \boldsymbol{A}^\top \boldsymbol{x} \geq \boldsymbol{v} \\ & \mathbb{1}^\top \boldsymbol{x} \leq 1 \\ & \boldsymbol{x} \geq 0 \\ & \boldsymbol{v} \geq 0 \end{aligned} \tag{24}$$

We denote by $\boldsymbol{v}$ (and analogously for $\boldsymbol{w}$) the column vector $\begin{pmatrix} v \\ \vdots \\ v \end{pmatrix}$. The dual (D) is:

$$\begin{aligned} \min_{\boldsymbol{y},w} v \text{ subject to } & \boldsymbol{A}\boldsymbol{y} \leq \boldsymbol{w} \\ & \mathbb{1}^\top \boldsymbol{y} \leq 1 \\ & \boldsymbol{y} \geq 0 \\ & \boldsymbol{w} \geq 0 \end{aligned} \tag{25}$$

Fact. For every (P) feasible solution $(\boldsymbol{x}, v)$ and for every mixed strategy $\boldsymbol{y} \in \Delta^{m_2}$, $(\boldsymbol{x}^\top \boldsymbol{A})\boldsymbol{y} \geq \boldsymbol{v}$.

This says that the expected utility for player I by playing $\boldsymbol{x}$, if player II chooses any $\boldsymbol{y}$, is at least $\boldsymbol{v}$.

Proof. By the definition of the dot product:

$$\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} = \sum_{j \in [m_2]} (\boldsymbol{x}^\top \boldsymbol{A})_j \cdot \boldsymbol{y}_j$$

By (P)-feasibility, $\boldsymbol{A}^\top \boldsymbol{x} \geq 0$ and $\boldsymbol{y}_j \geq 0$, so:

$$\sum_{j \in [m_2]} (\boldsymbol{x}^\top \boldsymbol{A})_j \cdot \boldsymbol{y}_j \geq \sum_{j \in [m_2]} v \cdot \boldsymbol{y}_j = v \sum_{j \in [m_2]} \boldsymbol{y}_j = v$$

□

Fact. If $\boldsymbol{A} \geq 0$, then for every (P)-optimal solution $(\boldsymbol{x}, v)$ we have that $\boldsymbol{x} \in \Delta^{m_1}$ is a mixed strategy for player I.

Proof. If $\mathbb{1}\boldsymbol{x} > 1 - \varepsilon$ for some $\varepsilon > 0$, then $(\frac{\boldsymbol{x}}{1-\varepsilon}, \frac{v}{1-\varepsilon})$ is also (P)-feasible. TODO □

Fact. For every (D)-feasible solution $(\boldsymbol{y}, w)$ and for every mixed strategy $\boldsymbol{x} \in \Delta^{m_1}$, $\boldsymbol{x}^\top(\boldsymbol{A}\boldsymbol{y}) \leq w$.

Fact. If $\boldsymbol{A} \geq 0$, then for every (D)-optimal solution $(\boldsymbol{y}, w)$ we have that $\boldsymbol{y} \in \Delta^{m_2}$ is a mixed strategy for player II.

Fact. (P) has an optimal solution of finite value.

Proof. The feasible set is non-empty: TODO □

Theorem 7.8 (Existence and structure of MNE in zero-sum games)**.** *For every two-player zero-sum game $\boldsymbol{A} \in \mathbb{R}_{\geq 0}^{m_1 \times m_2}$, there are mixed strategies $\boldsymbol{x} \in \Delta^{m_1}, \boldsymbol{y} \in \Delta^{m_2}$ such that:*

- *$\boldsymbol{x}$ and $\boldsymbol{y}$ are maximin and minimax strategies, respectively*
- *$(\boldsymbol{x}, \boldsymbol{y})$ is a Nash Equilibrium*
- *all Nash Equilibria have the same expected payoffs (the* value *of the game)*

Proof. Let (x, v) and (y, w) be optimal solutions of (P) and (D), respectively.

By the Strong Duality Theorem, we have that $v = w$. Using Facts 7.3 to 7.3, we know that x is a maximin strategy for player I, and y is a minimax strategy for player II.

x and y are mutual best responses, so (x, y) is a Nash Equilibrium. □

Corollary. Equilibrium computation in two-player zero-sum games is computable in polynomial time.

8 Lemke-Howson Algorithm

Recall by Definition 3.6 that a strategy profile σ^* is a MNE if for every i, the strategy σ_i^* is a best response to σ_{-i}^* – that is, for every player i and for every $\sigma_i \in \Delta^{|S_i|}$ then $u_i(\sigma_{-i}^*, \sigma_i^*) \geq u_i(\sigma_{-i}^*, \sigma_i)$. We have the following lemma:

Lemma 8.1. *Strategy profile σ^* is a Mixed Nash Equilibrium iff for each i, each pure strategy of i is either a best response to σ_{-i}^* or played with probability 0.*

The Lemke-Howson algorithm computes Nash Equilibria. We start by drawing the best response diagrams, yielding polyhedra, then convert these to polytopes by connecting all lines that go to infinity into a single point. Then we label each edge of this polytope with a strategy if that strategy is either a best response, or if it is played with probability 0. A Nash Equilibrium is represented by a fully-labelled (i.e. all pure strategies appear as labels) pair of points in the polytopes. See Fig. 1 for an example in a two-player 3×3 game.

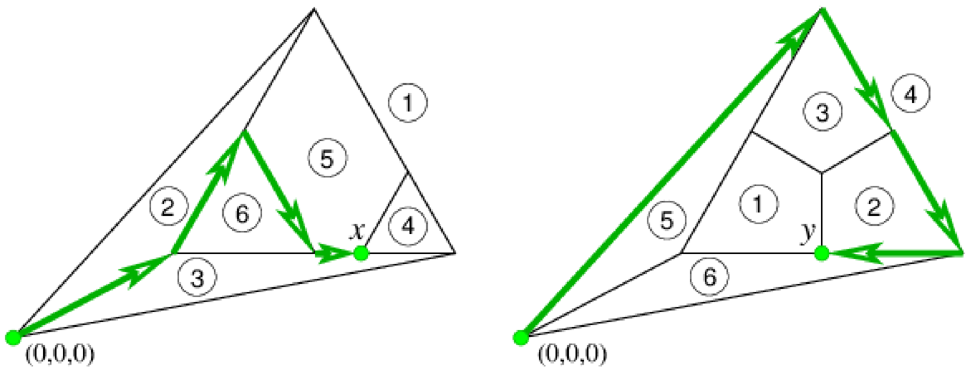


Figure 1: A run of the Lemke-Howson algorithm

The Lemke-Howson algorithm works by iterating over these two polytopes to find a pair of fully-labelled vertices. As the number of strategies grows, however, the number of points on the polytopes grows exponentially.

The procedure is:

- Start at a fully-labelled point e.g. $x_1 = \ldots = x_{|S_i|} = 0$.
- "Drop" a label by traversing an edge in the diagram (this will "gain" another label)
- Repeat step 2 until we are at another pair of fully-labelled vertices

If the input game is non-degenerate, the the Lemke-Howson algorithm finds at least one Nash Equilibrium because:

- there are finitely many vertices
- for a fixed label, the starting edge is unique
- for a fixed label, there is a unique continuation from that point

Hence we cannot visit any vertex twice.

8.1 Slack Variables

We can introduce variables into the constraints to turn the inequalities into equalities, so that we are only left with non-negativity constraints. Consider the following game:

I, II	4	5
1	3, 1	3, 0
2	2, 0	5, 2
3	0, 4	6, 3

We get the following polyhedron for player II:

$$\begin{aligned} H_{\text{II}} = \{(y_4, y_5, v) \,|\, 3y_4 + 3y_5 &\leq v \\ 2y_4 + 5y_5 &\leq v \\ 6y_5 &\leq v \\ y_4 + y_5 &= 1 \\ y_4, y_5 &\geq 0\} \end{aligned}$$

Let's convert this polyhedron to a polytope by dividing each inequality by v:

$$
\begin{aligned}
Q_{\mathrm{II}} &= \{y \mid \boldsymbol{A}y \le 1, y \ge 0\} && (26)\\
&= \{(y_4, y_5) \mid 3y_4 + 3y_5 \ \le 1 && (27)\\
&2y_4 + 5y_5 \qquad\qquad \le 1 && (28)\\
&6y_5 \qquad\qquad\qquad\quad \le 1 && (29)\\
&y_4 + y_5 \qquad\qquad\quad = 1 && (30)\\
&y_4, y_5 \qquad\qquad\quad\ \ \ge 0\} && (31)
\end{aligned}
$$

Now introduce new variables x_1, x_2, x_3 to turn each inequality into an equality:

$$
\begin{aligned}
x_1 + 3y_4 + 3y_5 &= 1\\
x_2 + 2y_4 + 5y_5 &= 1\\
x_3 + 6y_5 &= 1\\
x_1, x_2, x_3, y_4, y_5 &\ge 0
\end{aligned}
$$

These new variables measure the distance of a point within the polytope to the matching facet[3]. A vertex in the polytope joins two line segments, so represents two variables being set to 0. Furthermore, variable i is set to 0 if and only if label i is adjacent to the vertex. A solution to a system of equations is called *basic* if exactly two variables are zero.

Fact. Non-degeneracy of a game implies that at most two lines cross at a point.

8.2 Dictionary Form

A system of equations is in *dictionary form* when we express each slack variable as a function of the original variables. There is exactly one dictionary per basic solution. In our example, this is:

$$
\begin{aligned}
x_1 &= 1 - 3y_4 - 3y_5\\
x_2 &= 1 - 2y_4 - 5y_5\\
x_3 &= 1 - 6y_5
\end{aligned}
$$

8.3 Pivoting

Imagine we are increasing the slack variable such that the equality still holds. This corresponds to dropping that label in the polytope. We can only increase the variable so much before some variable on the right hand side (one of the original variables) becomes 0. In the example above, since $y_4, y_5 \ge 0$, we can see that (ignoring y_5) x_1 limits y_4 to be at most $\frac{1}{3}$, x_2 limits y_4 to be at most $\frac{1}{2}$, while x_3 does not limit y_4.

9 Nash's Theorem

We prove that every finite game has a Nash Equilibrium by drawing together two seemingly unrelated things.

9.1 Sperner's Lemma

Consider the graph formed from the triangulation of a triangle – that is, a graph that is made up only of smaller triangles.

Definition 9.1 (Valid colouring)**.** A valid colouring is a colouring of the vertices, each with a colour in the set $\{R, G, B\}$, such that:

[3]The facets of an n-polytope are the faces of the polytope with dimension $n-1$. For example, a 3-dimensional cube's facets are its square faces, but also has 1-dimensional (edges), and 0-dimensional (points) faces.

- each of the three corners is a distinct colour
- no vertex on the side of the big triangle has the same colour as the opposite corner

We call a triangle *panchromatic* if each of its vertices have a unique colour. We will now state Sperner's Lemma and prove it in two different ways.

Theorem 9.2 (Sperner's Lemma)**.** *Every valid colouring has an odd number of panchromatic triangles.*

Proof 1 (by counting RB triangles). Let:

- $\triangle^{RGB}$:= number of RGB triangles
- $\triangle^{RB}$:= number of RB triangles
- $|_X^{RB}$:= number of external RB edges, $|_I^{RB}$:= number of internal RB edges

By an external edge, we mean an edge that was formed directly from the original triangle, and an internal edge is every edge contained inside the triangulation.

Claim. $|_X^{RB}$ is odd.

Proof. Consider two corners of the large triangle, where the first is coloured red, and the other blue. Since we have a valid colouring, no vertex along the edges joining these two corners is coloured green, so we only have RB, RR, or BB edges. We get a new RB edge whenever we change colour. Since the colours of the corners are different, we must change colour an odd number of times, so the number of RB edges on the exterior is odd. ■

Claim. Each RGB triangle has one RB edge. Each RB triangle has two RB edges. All other triangles have 0 RB edges.

Claim. $\triangle^{RGB} + 2\triangle^{RB} = |_X^{RB} + 2|_I^{RB}$

Proof. The number of RB edges is equal to $\triangle^{RGB} + 2\triangle^{RB}$. This counts all the interior RB edges twice, as each internal edge separates two triangles, and each external RB edge once, as exactly one triangle touches it. ■

Hence $\triangle^{RGB} = |_X^{RB} + 2|_I^{RB} - 2\triangle^{RB} = 2(|_I^{RB} - \triangle^{RB}) + |_X^{RB}$, which is odd.

□

Proof 2 (by following the End-of-a-Line). Construct a directed graph based on the triangulation as follows:

- add edges from the B corner to all vertices on the RB side
- create a node for each cell (internal triangle), and one for the "outside" edge
- create a directed arc across every RB edge, where R is "to the left" as we traverse the arc)

Every node has in-degree at most one and out-degree at most one. Thus the graph consists only of isolated vertices, edge-disjoint paths, and edge-disjoint cycles. By the Handshaking Lemma, the number of nodes of degree exactly one is even. Nodes with degree exactly one appear in panchromatic triangles (one way in, no way out), and only RGB triangles contain these nodes of degree exactly one. We began by creating an artificial node outside of the large triangle – every other node of degree one corresponds to a panchromatic triangle. Hence the number of nodes of degree exactly one in the original graph is odd, so the number of RGB triangles in the original graph is odd. □

9.2 Brouwer's Fixed Point Theorem

A k-simplex is a k-dimensional polytope that is the convex hull of its $k+1$ vertices. The standard simplex that we will make use of is a simplex formed from $k+1$ standard unit vectors:

$$S^d = \{(x_1, \ldots, x_{d+1}) \mid x_1 + \ldots + x_{d+1} = 1, x_i \geq 0 \text{ for } i = 1, \ldots, d+1\} \tag{32}$$

A set is compact if it is *closed* – it contains all its limit points – and *bounded* – all its points lie within some fixed distance from one another.

A convex set is a set of points such that, given any two points in the set, the line joining them lies entirely within that set.

A fixed point of a function $f : A \to B$ is an element of the function's domain that maps to itself, that is, $c \in A$ is a fixed point of f if $f(c) = c$.

Theorem 9.3 (Brouwer's Fixed Point Theorem)**.** *If $f : S \to S$ is a continuous function from a compact convex set to itself, then f has a fixed point.*

Proof (for d=1). S^1 is the unit interval. Let $f : [0,1] \to [0,1]$ be a continuous function. If $f(0) = 0$ or $f(1) = 1$ then we have found our fixed point. Otherwise, we have $f(0) > 0$ and $f(1) < 1$. The function $g(x) := x - f(x)$ is continuous, and $g(0) < 0$ and $g(1) > 0$. By the Intermediate Value Theorem, there must be some $x \in [0,1]$ for which $g(x) = 0$. Hence $x = f(x)$. □

Proof (for d=2). The function f sends point (α, β, γ) to another point in the triangle: $f(\alpha, \beta, \gamma) = (\overline{\alpha}, \overline{\beta}, \overline{\gamma})$. We begin the proof by defining a valid colouring of S^2 based on f:

- R if $\alpha > \overline{\alpha}$
- B if $\alpha \leq \overline{\alpha}$ and $\beta > \overline{\beta}$
- G if $\alpha \leq \overline{\alpha}$ and $\beta \leq \overline{\beta}$ and $\gamma < \overline{\gamma}$

Fact. If a point has no colour, then it is a fixed point.

Claim. The colouring is valid.

Proof. Recall that a colouring is valid if each of the three vertices of a triangle receives a different colour and no point on the side is assigned the colour equal to the colour on the opposite corner. The colours of the points $(1,0,0), (0,1,0)$, and $(0,0,1)$ are R, B, and G respectively. Furthermore,

- no point on the $(1,0,0) - (0,1,0)$ edge has colour G
- no point on the $(0,0,1) - (1,0,0)$ edge has colour B
- no point on the $(0,1,0) - (0,0,1)$ edge has colour R

■

We then use Sperner's Lemma and the compactness of S^2 to find a fixed point. Let $T_1, T_2, \ldots$ be triangulations of S^2 such that the largest diameter of the cells converges to 0. By Sperner's Lemma, every T_i has a panchromatic cell with corners:

- $x_i = (\alpha_i, \beta_i, \gamma_i)$ with colour R
- $x_i' = (\alpha_i', \beta_i', \gamma_i')$ with colour B
- $x_i'' = (\alpha_i'', \beta_i'', \gamma_i'')$ with colour G

In other words, assuming that we have:

$$\begin{aligned} f(x_i) &= (\overline{\alpha}_i, \overline{\beta}_i, \overline{\gamma}_i) \\ f(x_i') &= (\overline{\alpha}_i', \overline{\beta}_i', \overline{\gamma}_i') \\ f(x_i'') &= (\overline{\alpha}_i'', \overline{\beta}_i'', \overline{\gamma}_i'') \end{aligned} \tag{33}$$

Then:

$$\begin{aligned} \alpha_i &> \overline{\alpha}_i \text{ (as } x_i \text{ is R)} \\ \beta_i' &> \overline{\beta}_i' \text{ (as } x_i' \text{ is B)} \\ \gamma_i'' &> \overline{\gamma}_i'' \text{ (as } x_i'' \text{ is G)} \end{aligned} \tag{34}$$

Since S^2 is a compact (closed and bounded) subset of $\mathbb{R}^3$, by the Bolzano-Weierstrass Theorem[4], there is a sequence $i_1 < i_2 < \ldots$ such that $\lim_{k\to\infty} x_{i_k}$ exists (the sequence $\langle x_{i_k}\rangle$ converges).

Since the diameters of the cells in triangulations $T_{i_1}, T_{i_2}, \ldots$ converges to 0, we have $\lim_{k\to\infty} x_{i_k} = \lim_{k\to\infty} x'_{i_k} = \lim_{k\to\infty} x''_{i_k}$. Let $z = (\alpha, \beta, \gamma)$ and $f(z) = (\overline{\alpha}, \overline{\beta}, \overline{\gamma})$. Using (34), we have $\alpha \geq \overline{\alpha}, \beta \geq \overline{\beta}, \gamma \geq \overline{\gamma}$. Thus it follows that $\alpha = \overline{\alpha}, \beta = \overline{\beta}, \gamma = \overline{\gamma}$ since $\alpha + \beta + \gamma = 1$ and $\overline{\alpha} + \overline{\beta} + \overline{\gamma}$, so we have $f(z) = z$, a fixed point. □

Claim 9.4. *Brouwer's Theorem can be proved for $d \geq 3$ by induction.*

Remark. Brouwer's Fixed Point Theorem holds for continuous functions $f : C \to C$ where C is convex and compact. Any convex compact set can be continuously deformed into S^d for some d.

9.3 Nash's Theorem

We now prove the big Nash's Theorem for two players by bringing together the previous two results. The proof for n players is a straightforward generalisation. Note that by a finite game, we mean a game in which there is a finite number of players each with a finite number of pure strategies.

Theorem 9.5 (Nash, 1951)**.** *Every finite game has a Nash Equilibrium.*

Proof of Nash's Theorem. Player I has payoff matrix $\boldsymbol{A}$ and plays mixed strategy $\boldsymbol{x} \in \Delta^{m_1}$, while player II has payoff matrix $\boldsymbol{B}$ and plays mixed strategy $\boldsymbol{y} \in \Delta^{m_2}$.

Let $k_i(x, y)$ denote player I's gain in utility from switching to strategy $i \in S_{\mathrm{I}}$, and $k'_j(x, y)$ denote player II's gain in utility from switching to strategy $j \in S_{\mathrm{II}}$:

$$\begin{aligned} k_i(x,y) &:= \max\{0, (\boldsymbol{Ay})_i - \boldsymbol{x}^\top \boldsymbol{Ay}\} \\ k'_j(x,y) &:= \max\{0, (\boldsymbol{x}^\top \boldsymbol{B})_j - \boldsymbol{x}^\top \boldsymbol{Ay}\} \end{aligned} \tag{35}$$

We are at a Nash Equilibrium when $k_i(x, y) = k'_j(x, y)$ for some strategies $i \in S_{\mathrm{I}}, j \in S_{\mathrm{II}}$. Define the following mixed strategy for player I:

$$g(x,y) := \frac{1}{1 + \sum_{i\in[m_1]} k_i(x,y)} \begin{pmatrix} x_1 + k_1(x,y) \\ \vdots \\ x_{m_1} + k_{m_1}(x,y) \end{pmatrix} \tag{36}$$

And the same for player II:

$$h(x,y) := \frac{1}{1 + \sum_{j\in[m_2]} k'_j(x,y)} \begin{pmatrix} x_1 + k'_1(x,y) \\ \vdots \\ x_{m_2} + k'_{m_2}(x,y) \end{pmatrix} \tag{37}$$

Claim. $g(x, y)$ and $h(x, y)$ are valid mixed strategies.

Proof. We will prove for $g(x, y)$, and the same reasoning applies for $h(x, y)$. Let $\overline{x} := \begin{pmatrix} x_1 + k_1(x,y) \\ \vdots \\ x_{m_1} + k_{m_1}(x,y) \end{pmatrix}$.

In a valid mixed strategy the sum of entries must be equal to 1. We have:

$$\begin{aligned} \sum_i \overline{x}_i &= \sum_i (x_i + k_i(x,y)) = \sum_i x_i + \sum_i k_i(x,y) \\ &= 1 + \sum_i k_i(x,y) \end{aligned}$$

Hence, $\frac{1}{1+\sum_i k_i(x,y)} \sum_i \overline{x}_i = 1$ and each $\overline{x}_i \geq 0$, since $k_i(x, y) \geq 0$ by definition. ■

[4]Each bounded sequence in $\mathbb{R}^n$ has a convergent subsequence.

Now define the continuous function $f : \Delta^{m_1} \times \Delta^{m_2} \to \Delta^{m_1} \times \Delta^{m_2}$ as follows:

$$f(x, y) = (g(x, y), h(x, y)) \tag{38}$$

Since $\Delta^{m_1} \times \Delta^{m_2}$ is a convex compact set, Brouwer's Theorem implies that f has a fixed point. That is, there is an $x^* \in \Delta^{m_1}$ and a $y^* \in \Delta^{m_2}$ such that $f(x^*, y^*) = (x^*, y^*)$. Now we just need to show that (x^*, y^*) is a Nash Equilibrium.

Claim. (x^*, y^*) is a Nash Equilibrium.

Proof. We must show that for all $i \in [m_1], j \in [m_2], k_i(x^*, y^*) = k'_j(x^*, y^*) = 0$. Assume there is an $i \in [m_1]$ such that $k_i(x^*, y^*) > 0$. Then there must be a $j \in [m_1]$ such that $x_j \neq 0$ and $k_j(x^*, y^*) = 0$. If we suppose that this is not true (i.e. that every $k_j(x^*, y^*) > 0$), then we have:

$$\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} = \sum_{\ell \in [m_1]} x_\ell (\boldsymbol{A}\boldsymbol{y})_\ell > \sum_{\ell \in [m_1]} x_\ell (\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}) = \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} \sum_{\ell \in [m_1]} x_\ell = \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}$$

A contradiction. ■

Since (x^*, y^*) is a fixed point of f, $g(x^*, y^*) = x^*$. Therefore,

$$\frac{x_i^* + k_i(x^*, y^*)}{1 + \sum_{\ell \in [m_1]} k_\ell(x^*, y^*)} = x^* \qquad (\neq 0 \text{ as } x_i^* + k_i(x^*, y^*) \neq 0)$$

Rearranging, we have $\sum_{\ell \in [m_1]} k_l(x^*, y^*) = \dfrac{k_i(x^*, y^*)}{x_i^*} > 0$, but then:

$$\begin{aligned} \frac{x_j^* + k_j(x^*, y^*)}{1 + \sum_{\ell \in [m_1]} k_\ell(x^*, y^*)} &= \frac{x_j^*}{1 + \sum_{\ell \in [m_1]} k_\ell(x^*, y^*)} \\ &= \frac{x_j^*}{1 + k_i(x^*, y^*)/x_i^*} \neq x_j^* \end{aligned}$$

This contradicts the fact that (x^*, y^*) is a fixed point, and hence there is no i such that $k_i(x^*, y^*) = 0$. Similar reasoning shows that $k'_j(x^*, y^*) = 0$ for all $j \in [m_2]$. Therefore no player can improve their expected payoff by switching strategy, so no player the incentive to unilaterally deviate. Thus (x^*, y^*) is a Nash Equilibrium. □

10 Complexity of computing equilibria

We can compute Nash Equilibria in two-player zero-sum games in polynomial time using linear programming as in Section 7.1. However, in general it is not possible to compute exact Nash Equilibria unless some artificial encoding of the output is used, as Nash Equilibria may contain irrational numbers.

Definition 10.1 (ε-Nash Equilibrium)**.** A strategy profile σ^* is an ε-Nash Equilibrium if for all players i and for all $\sigma_i \in \Delta^{|S_i|}$:

$$u_i(\sigma^*_{-i}, \sigma^*_i) \geq u_i(\sigma^*_{-i}, \sigma_i) - \varepsilon \tag{39}$$

Definition 10.2 (ε-approximate fixed point)**.** Let S be a normed space (i.e. a space on which a norm is defined). The point x^* is an ε-approximate fixed point of $f : S \to S$ if $|f(x^*) - x^*| \leq \varepsilon$.

Let $M = \max_i |S_i|$ and R be the difference between the largest and smallest payoff in the game. Let

$$\alpha = \frac{\varepsilon}{M^2 R^2} \tag{40}$$

Any α-approximate fixed point of the function constructed in the proof of Nash's Theorem corresponds to an ε-Nash Equilibrium. The function in the proof of Nash's Theorem is sufficiently "smooth", so that by dividing the domain into a fine grid and applying Sperner's Lemma, we can find an approximate fixed point of the function. The following fact formalises the intuition of smoothness.

Fact. The function constructed in Nash's Theorem is $O(nM^2R)$-Lipschitz, i.e., for all x, y:

$$||f(x) - f(y)||_\infty \leq O(nM^2R) \cdot ||x - y||_\infty \tag{41}$$

10.1 Scarf's Theorem

The following theorem by Scarf connects Lipschitz-continuity of a function and the granularity of the subdivision of our domain required to identify α-approximate fixed points.

Theorem 10.3 (Scarf's). *Let S be a d-dimensional simplex that is subdivided into subsimplices of diameter at most $\delta > 0$. Let $f : S \to S$ be a function from S to itself. Colour every vertex v of every subsimplex according to the rules from Brouwer's Theorem (i.e., if v receives a colour, then $f(v)_i \leq v_i$. If we choose δ such that:*

- $\delta \leq \frac{\alpha}{2d}$
- $||x - y||_\infty \leq \delta \Rightarrow ||f(x) - f(y)||_\infty \leq \frac{\alpha}{2d}$ *for all x, y*

Then any point in a fully-coloured subsimplex is an α-approximate fixed point.

Proof 1. Consider point $x = (x_1, \ldots, x_{d+1})$ in a fully-coloured simplex. Either x is already a fixed point, or one of the coordinates of x decreases. WLOG assume x_1 decreases, so $f(x)_1 < x_1$. We show that no coordinate can increase by more than $\frac{\alpha}{d}$, i.e. for all $j \in \{1, \ldots, d+1\}$, $f(x)_j \leq x_j + \frac{\alpha}{d}$. This is certainly true for $j = 1$; assume for contradiction that it is not true for some other j, i.e. that there is a $j \in \{2, \ldots, d+1\}$ such that:

$$f(x)_j > x_j + \frac{\alpha}{d}$$

Let y be the corner of the subsimplex with colour j – y is a corner such that $f(y)_j \leq y_j$. Note that both x and y are in the subsimplex of diameter δ, hence

$$f(y)_j \leq y_j \leq x_j + \delta$$

Combining the two inequalities, we get

$$f(x)_j - f(y)_j > \frac{\alpha}{d} - \delta \geq \frac{\alpha}{2d} \tag{42}$$

As x and y are both in the subsimplex we have have $||x - y||_\infty \leq \delta$, and by the smoothness of the subdivision we have $||f(x) - f(y)||_\infty \leq \frac{\alpha}{2d}$. Hence, $f(x)_j - f(y)_j \leq \frac{\alpha}{2d}$ (since $||f(x) - f(y)||_\infty \leq k$ means that $f(x)_j - f(y)_j \leq k$ for all j). However this contradicts (42) – that $f(x)_j - f(y)_j > \frac{\alpha}{2d}$ – so we have that $f(x)_j \leq x_j + \frac{\alpha}{d}$ for all $j \in \{1, \ldots, d+1\}$.

Since we are working with Barycentric coordinates, all coordinates must sum to 1. If $f(x)_j \leq x_j + \frac{\alpha}{d}$ for all $j \in \{1, \ldots, d+1\}$, then $f(x)_j \geq x_j - \alpha$ for all $j \in \{1, \ldots, d+1\}$. Hence x is an α-approximate fixed point of f. ☐

Proof 2. Consider the point $x = (x_1, \ldots, x_{d+1})$ is a fully-coloured subsimplex. We prove that x is an α-approximate fixed point of f by showing that for every $j \in \{1, \ldots, d+1\}$:

1) $f(x)_j - x_j \leq \frac{\alpha}{d}$
2) $f(x)_j - x_j \geq -\alpha$

We begin by showing (1). Let y be the corner of the same subsimplex that has colour j, i.e. $f(y)_j < y_j$. Since x and y are in the same subsimplex of diameter δ, we have $y_j \leq x_j + \delta$ (since $||x - y||_\infty \leq \delta$). By the smoothness of f, we have $||f(x) - f(y)||_\infty \leq \frac{\alpha}{2d}$.

Now:

$$\begin{aligned} f(x)_j - x_j = [f(x)_j - f(y)_j] + [f(y)_j - x_j] &\leq \frac{\alpha}{2d} + [y_j - x_j] \\ &\leq \frac{\alpha}{2d} + \delta \\ &\leq \frac{\alpha}{2d} + \frac{\alpha}{2} = \frac{\alpha}{d} \end{aligned}$$

Hence we have shown that $f(x)_j - x_j \le \frac{\alpha}{d}$. Now we show (2): that $x_j - f(x)_j \le \alpha$. We have:

$$\begin{aligned} x_j - f(x)_j &= [1 - \sum_{i:i\neq j} x_i] - [1 - \sum_{i:i\neq j} f(x)_i] \\ &= \sum_{i:i\neq j} (f(x)_i - x_i) \le d \cdot \frac{\alpha}{d} \end{aligned}$$

Therefore every coordinate is at most α away from the coordinate's image, hence x is an α-approximate fixed point. $\square$

10.2 The number of subsimplices

Notice that the function from the proof of Nash's Theorem has dimension $d = \sum_{i\in[n]} |S_i| = \sum_{i\in[n]} m_i$ (the sum of the number of pure strategies for all players). Using Fact 10 and Scarf's Theorem, we conclude that if we divide the domain into subsimplices of diameter $O(\frac{\alpha}{nd^3 R}$, then any point inside a fully-coloured simplex will be an α-approximate fixed point of the function. Hence $O([\frac{(nM)^2 R}{\alpha}]^d)$ many subsimplices are sufficient. We can find a fully-coloured subsimplex (by searching all of them), and this will correspond to an α-approximate fixed point, hence an approximate Nash Equilibrium. Note that the total number of subsimplices is polynomial in ε but exponential in the number of pure strategies.

Corollary. There exists a constant c' such that if $\delta = c' \cdot \frac{\alpha}{ndM^2 R}$, then every point in a fully-coloured simplex is an ε-Nash Equilibrium.

Proof. We use Scarf's Theorem and Fact 10. We divide S into subsimplices of degree $\delta = c' \cdot \frac{\alpha}{ndM^2 R} = c' \cdot \frac{\varepsilon}{ndM^4 R^2}$. The number of subsimplices in each subdivision is $\frac{1}{d} = \frac{ndM^4 R^2}{c' \varepsilon}$. This is $O(\frac{1}{\delta}^d)$. $\square$

We can compute such a c by simply enumerating all subsimplices and checking which correspond to Nash Equilibria. However, note that that while the total number of subsimplices is polynomial in ε, it is exponential in the number of pure strategies.

11 PPAD

Definition 11.1 (Search problem)**.** An algorithmic problem is a total search problem if each instance x has a search space $S_x \subseteq \{0,1\}*$ of bitstrings of length polynomial in $|x|$, as well as a non-empty subset Q_x of valid solutions.

Given the input x to some search problem, we wish to find a $y \in Q_x$. For example, the computational problem Nash is, given a finite game in strategic form and parameter $\varepsilon > 0$, find an ε-Nash Equilibrium. By Nash's Theorem, every finite game has a Nash Equilibrium, hence every finite game also has an ε-Nash Equilibrium.

The problem Sperner is, given a valid colouring of a subdivision of a simplex, compute a fully-coloured vertex in such a colouring.

The problem α-Brouwer is, given a continuous function $f : S \to S$ from a compact convex set to itself and parameter $\alpha > 0$, find an α-approximate fixed point of f.

11.1 End-of-a-Line

The problem (Explicit) End-of-a-Line is, given a directed graph $G = (V, \vec{E})$ such that every vertex has at most one predecessor and at most one successor, and an initial vertex v_0 (the *standard source*) with in-degree 0 and out-degree 1, find a vertex of out-degree 1 that is not equal to v_0. Recall that the graph we constructed when computing the number of panchromatic triangles is of such a structure. To solve Explicit End-of-a-Line, we can simply follow the edges of the graph until we reach a sink.

Another formulation of the problem is as follows: we are given a graph $G = (V, \vec{E})$ in which vertices are k-bit strings (so $|V| \le 2^k$). The graph is encoded by two functions $P, S : \{0,1\}^k \to \{0,1\}^k$ that compute a node's predecessor and successor, respectively. That is, there is an edge (u, v) if and only if $S(u) = v$,

$P(v) = u$, and $v \neq (0, \ldots, 0)$. The End-of-a-Line problem is to find a node other than $(0, \ldots, 0)$ that either has a successor but no predecessor or has a predecessor but no successor.

Due to a simple parity argument, we know such a node exists – the problem is in finding one. We could just test all nodes in the graph, but since there are 2^k of them this cannot be done in time polynomial in the size of the input. It is not known whether End-of-a-Line can always be solved in polynomial time.

Definition 11.2 (Polynomial-time reduction)**.** A polynomial-time reduction from a search problem X to search problem Y is a pair (f, g) of polynomial-time computable functions such that f maps inputs of X to inputs of Y, and g maps valid solutions of Y to valid solutions of X. We write $X \leq_m Y$.

For $x \in X$ we have $f(x) \in Y$, and for any solution y of Y for input $f(x)$ we have $g(x, y)$ as a solution to X for input x.

Informally, the class PPAD (Polynomial Parity Argument (Directed case)) is the set of all problems whose solution space can be formulated as the set of all sinks and all nonstandard sources in a directed graph with the properties as above. Equivalently, PPAD is the class of all search problems π such that $\pi \leq_p$ End-of-a-Line. While PPAD is a good metric for intractability, it is a subset of NP and hence "not as hard".

Theorem 11.3. Nash $\leq_p$ Brouwer $\leq_p$ Sperner $\leq_p$ End-of-a-Line

Theorem 11.4 (Goldberg, 2006)**.** Nash *is PPAD-complete.*

12 Correlated Equilibria

Recall the Battle of the Sexes game:

Alice, Bob	**Costa**	**Starbucks**
Costa	3,1	0,0
Starbucks	0,0	1,3

There are two pure Nash Equilibria: (Costa, Costa) and (Starbucks, Starbucks). The first is unfair to Bob, as he would rather go to Starbucks, while the second is unfair to Alice as she prefers Costa. There is one mixed Nash Equilibrium: $((\frac{3}{4}, \frac{1}{4}), (\frac{1}{4}, \frac{3}{4}))$. In this case Alice and Bob both have expected utility $\frac{3}{4}$ – this is good in that it is fair, but bad in that $\frac{3}{4} < 1$, hence they receive lower expected utility than the minimum they would receive had they just played a pure strategy. Furthermore, there is a $\frac{3}{4} \cdot \frac{3}{4} + \frac{1}{4} \cdot \frac{1}{4} = \frac{5}{8}$ probability that they do not meet. So neither of these solutions are particularly attractive.

Now recall the Game of Chicken:

I, II	**Swerve**	**Straight**
Swerve	0,0	-1,1
Straight	1,-1	-10,-10

The pure Nash Equilibria are (Straight, Swerve) and (Swerve, Straight), and the mixed Nash Equilibrium is $((\frac{9}{10}, \frac{1}{10}), (\frac{9}{10}, \frac{1}{10}))$. These are more fair, but still, there is a $\frac{1}{10} \cdot \frac{1}{10} = 1\%$ chance of a crash, and maybe we aren't in the mood for such risk. A natural solution is to let each player win with probability, which avoids all ties and crashes. We want:

I, II	**Swerve**	**Straight**
Swerve	0%	50%
Straight	50%	0%

12.1 Product Distributions

Suppose player I plays strategy $s_1 \in S_{\mathrm{I}}$ with probability x_1 and strategy $s_2 \in S_{\mathrm{I}}$ with probability x_2, and player II plays strategy $t_1 \in S_{\mathrm{II}}$ with probability y_1 and strategy $t_2 \in S_{\mathrm{II}}$ with probability y_2. The product distribution is as follows:

I, II	y_1	y_2
x_1	$x_1 \cdot y_1$	$x_1 \cdot y_2$
x_2	$x_2 \cdot y_1$	$x_2 \cdot y_2$

To be a valid product distribution, we must have that $x_1, x_2, y_1, y_2 \geq 0$, $x_1 + x_2 = 1$, and $y_1 + y_2 = 1$. However, there is no $(x_1, x_2), (y_1, y_2)$ that satisfies both these conditions and $x_1 y_2 = \frac{1}{2}$, $x_2 y_1 = \frac{1}{2}$.

We want to design a mechanism/trusted mediator/traffic light that will advise the players on what to do.

12.1.1 Example

Consider the probability distribution on strategy profiles $\sigma = (z_1, z_2, z_3, z_4)$ such that $\sum_i z_i = 1$ and $z_i \geq 0$:

I, II	y_1	y_2
x_1	z_1	z_2
x_2	z_3	z_4

The traffic light works by showing each player a strategy to follow, and the players may or may not follow this recommendation. The players know the product distribution (i.e., the likelihood of ending up in each cell in the table), but not what strategy the other has been advised to follow.

Is the solution $\sigma = (10\%, 30\%, 40\%, 20\%)$ stable? For clarity let player I choose between strategies 1 (swerve) and 2 (straight), and player II choose between strategies a (swerve) and b (straight). The game is:

I, II	a	b
1	0,0	-1, 1
2	1, -1	-10, -10

With product distribution:

I, II	a	b
1	10%	30%
2	40%	20%

Suppose player I is shown the signal 2. We have:

$$Pr[a|2] = \frac{Pr[a \text{ and } 2]}{Pr[2]} = \frac{40}{40+20} = \frac{2}{3}$$
$$Pr[b|2] = \frac{Pr[b \text{ and } 2]}{Pr[2]} = \frac{20}{40+20} = \frac{1}{3}$$

The expected utilities for player I are as follows:

$$\mathbb{E}[u_{\mathrm{I}}(2, (\frac{2}{3}, \frac{1}{3}))] = \frac{2}{3} \cdot 1 + \frac{1}{3} \cdot -10 = -\frac{8}{3}$$
$$\mathbb{E}[u_{\mathrm{I}}(1, (\frac{2}{3}, \frac{1}{3}))] = \frac{2}{3} \cdot 0 + \frac{1}{3} \cdot -1 = -\frac{1}{3}$$

Hence $\sigma = (10\%, 30\%, 40\%, 20\%)$ is not stable, as player I will not respect the advice that the traffic light gives it (as strategy 1 has greater expected utility).

12.2 Correlated Equilibria

Definition 12.1 (Correlated Equilibrium)**.** A joint mixed strategy profile $\sigma \in \Delta_{S_1 \times S_n}$ is a correlated equilibrium if for every player i and for all strategies $s_i, t_i \in S_i$, we have:

$$\sum_{\boldsymbol{x} \in S_{-i}} u_i(\boldsymbol{x}, s_i) \cdot \sigma(\boldsymbol{x}, s_i) \geq \sum_{\boldsymbol{x} \in S_{-i}} u_i(\boldsymbol{x}, t_i) \cdot \sigma(\boldsymbol{x}, s_i)$$

In the above definition, $u_i(\boldsymbol{x}, s_i)$ gives the utility to player i if they player s_i, given that everyone else plays $\boldsymbol{x}$, and $\sigma(\boldsymbol{x}, s_i)$ gives the probability that the traffic light tells player i to play strategy s_i.

Fact. If $\sigma^* = (\sigma_1^*, \dots, \sigma_n^*)$ is a Mixed Nash Equilibrium, then $\overline{\sigma}^* \in \Delta^{S_1 \times \dots \times S_n}$ is a Correlated Equilibrium, where

$$\overline{\sigma}^*(s_1, \dots, s_n) := \sigma_1^*(s_1) \cdot \dots \cdot \sigma_n^*(s_n) = \prod_i \sigma_i^*(s_i)$$

The concept of the Correlated Equilibrium is a generalisation of the Nash Equilibrium. Nash's Theorem states that every finite game has a Nash Equilibrium, hence using the above fact every finite game also has a Correlated Equilibrium.

Proof that correlated equilibria always exist. Let $\sigma^* = (\sigma_1^*, \dots, \sigma_n^*)$ be a Nash Equilibrium. Then for every $i \in [n]$, for every $s_i \in S_i$, if $\sigma_i^*(s_i) > 0$ then s_i is a best response to σ_{-i}^*, i.e. for all $t_i \in S_i$, $u_i(\sigma_{-i}^*, s_i) \geq u_i(\sigma_{-i}^*, t_i)$.

Equivalently, for all players $i \in [n]$ and all $s_i, t_i \in S_i$, $\sigma_i^*(s_i) \cdot u_i(\sigma_{-i}^*, s_i) \geq \sigma_i^*(s_i) \cdot u_i(\sigma_{-i}^*, t_i)$. By noting that $u_i(\sigma_{-i}^*, s_i) = \sum_{x \in S_{-i}} \overline{\sigma}_{-i}^*(x) \cdot u_i(x, s_i)$, we thus get:

$$\sigma_i^*(s_i) \cdot \sum_{x \in S_{-i}} \overline{\sigma}_{-i}^*(x) \cdot u_i(x, s_i) \geq \sigma_i^*(s_i) \cdot \sum_{x \in S_{-i}} \overline{\sigma}_{-i}^*(x) \cdot u_i(x, t_i)$$

This is equivalent to:

$$\sum_{x \in S_{-i}} \overline{\sigma}^*(x) \cdot u_i(x, s_i) \geq \sum_{x \in S_{-i}} \overline{\sigma}^*(x) \cdot u_i(x, t_i)$$

This is the definition of a correlated equilibrium, hence every finite game has a correlated equilibrium as a result of Nash's Theorem. □

12.2.1 Example

Recall the game of Chicken:

I, II	**3**	**4**
1	0,0	-1,1
2	1,-1	-10,-10

With product distribution:

I, II	**3**	**4**
1	z_{11}	z_{12}
2	z_{21}	z_{22}

For player I:

$$\begin{aligned} 0z_{11} - z_{12} &\geq z_{11} - 10z_{12} \\ z_{21} - 10z_{22} &\geq 0z_{21} - z_{22} \end{aligned}$$

For player II:

$$\begin{aligned} 0z_{11} - z_{21} &\geq z_{11} - 10z_{21} \\ z_{12} - 10z_{22} &\geq 0z_{12} - z_{22} \end{aligned}$$

A correlated equilibrium can be computed in polynomial time. We can formulate the problem as a linear program – for the game of Chicken above we have the following:

$$\max_{z_{11},z_{12},z_{21},z_{22}} -20z_{22} \text{ subject to } 0z_{11} - 1z_{12} \geq 1z_{11} - 10z_{12}$$
$$1z_{21} - 10z_{22} \geq 0z_{21} - 1z_{22}$$
$$0z_{11} - 1z_{21} \geq 1z_{11} - 10z_{21}$$
$$1z_{12} - 10z_{22} \geq 0z_{12} - 1z_{22}$$
$$z_{11} + z_{12} + z_{21} + z_{22} = 1$$
$$z_{11}, z_{12}, z_{21}, z_{22} \geq 0$$

This simplifies to:

$$\max_{z_{11},z_{12},z_{21},z_{22}} -20z_{22} \text{ subject to } 9z_{12} \geq z_{11}$$
$$z_{21} \geq 9z_{22}$$
$$9z_{21} \geq z_{11}$$
$$z_{12} \geq 9z_{22}$$

The value of -20 arises as a result of the *social utility*, the sum of all utilities. Note that to maximise the social utility, it is enough to set z_{22} to 0, meaning the players never crash.

13 Auctions

13.1 Single-Item Auctions

In a single-item auction there are:

- 1 item
- n bidders interested in acquiring the item
- a private valuation $v_i \in \mathbb{R}$ for obtaining the item

The goal is to give the item to the bidder with the highest valuation for it. As a first attempt at designing such a mechanism, we simply ask each bidder for their bid $b_i \in \mathbb{R}$ and give the item to the bidder who bids the highest. Note that since v_i is private, bidders have the option to act untruthfully. This is a poor mechanism as it can be manipulated – bidders may bid arbitrarily high and are not incentivised to report their true valuations.

For our second attempt, we consider the First-Price Auction: first, ask the bidders to submit a bid b_i, give the item to the bidder with the highest bid and make them pay this bid. Bidders have *quasi-linear utility*, so if the item is awarded to bidder i they receive utility $u_i = v_i - b_i$, while everyone else gets 0. This mechanism stops players bidding arbitrarily high, as their is the risk they actually have to pay that bid. However, the mechanism can still be manipulated – the winner has the incentive to bid $v_j + \varepsilon$, where v_j is the second highest valuation. This would net them utility $v_i - (v_j + \varepsilon) = v_i - v_j - \varepsilon > 0$.

13.1.1 Vickrey's Mechanism

Vickrey's mechanism works as follows:

- Ask for each player's bid b_i
- Make the player i with the highest bid pay the second highest bid B – the winner gets utility $u_i = v_i - B$, while everyone else gets utility 0.

Theorem 13.1. *For all bids $b_1, \ldots, b_n$, let u_i be player i's utility if $b_i = v_i$, and u'_i otherwise. Then $u_i \geq u'_i$ i.e. telling the truth is a dominant strategy.*

Proof. Assume i wins by bidding v_i (i.e. by being truthful) and let B denote the second highest bid. Then $u_i = v_i - B \geq 0$, as either i wins and pays $v_i - B$ where $v_i > B$, or i loses and gets utility 0.

Suppose $b_i > B$. Player i still wins and pays the same: $u'_i = v_i - B$, so there is no incentive to overbid. Now suppose that $b_i < B$. i would lose, so $u'_i = 0 \leq u_i$.

Now assume i loses by bidding $b_i = v_i$, so $u_i = 0$. Let player j be the winner, so $b_j \geq v_i$. For $b_i < b_j$, i still loses, so $u'_i = 0$. For $b_i > b_j$, i wins and pays b_j, so $u'_i = v_i - b_j \leq 0$.

In all cases, telling the truth by setting $b_i = v_i$ is no worse than attempting to manipulate the mechanism, so truth-telling is a dominant strategy. □

Definition 13.2 (Incentive Compatibility)**.** An auction mechanism is incentive-compatible if bidding their true valuation is a weakly dominant strategy for each bidder.

An auction being incentive compatible is a stronger condition than it just having a Nash Equilibrium. Vickrey's mechanism is incentive compatible for single-item auctions.

13.2 Combinatorial Auctions

In a combinatorial auction there is:

- a set S of items to be sold
- n bidders
- for every bidder i and every subset $S' \subseteq S$, $v_i(S') \in \mathbb{R}$ is the valuation of bidder i for set S'

An auction mechanism then:

- receives bids $b_1, \dots, b_n : 2^S \to \mathbb{R}$
- allocates disjoint sets $A_1, \dots, A_n$ of items to bidders $1, \dots, n$ (so $A_1, \dots, A_n \subseteq S, A_i \cap A_j \neq \emptyset$ for $i \neq j$)
- charges bidder i the price p_i

Note that the valuations $v_i : 2^S \to \mathbb{R}$ are objects of size exponential in the size of S, or the number of items on offer. We may succinctly represent the bidders' valuations by a graph in which bidder i wants to buy a path from vertex s_i to vertex t_i. A path is a set of edges S'. Consider the valuation:

$$v_i(S') = \begin{cases} \frac{1}{|S'|} & S' \text{ is a path from } s_i \text{ to } t_i \\ 0 & \text{otherwise} \end{cases}$$

Definition 13.3 (Social welfare maximising auction mechanism)**.** An auction mechanism is social welfare maximising if its allocation of items to bidders maximises (over all possible allocations) the social welfare $\sum_{i \in [n]} v_i(A_i)$

13.2.1 VCG mechanism

The Vickrey-Clarke-Groves mechanism works as follows. It:

- take bids $b_1, \dots, b_n$
- compute an allocation $A_1, \dots, A_n$ that maximises $\sum_i b_i(A_i)$ ("believe" the bidders)
- allocate $A_1, \dots, A_n$ to bidders $1, \dots, n$
- for every bidder k, compute an allocation $A_1^{-k}, \dots, A_n^{-k}$ (an allocation excluding k) that maximises $\sum_{i:i \neq k} b_i(A_i^{-k})$
- charge bidder k the amount $\sum_{i:i \neq k} b_i(A_i^{-k}) - \sum_{i:i \neq k} b_i(A_i)$ (each bidder is charged their EXTERNALITY

An externality is the decline in values of bidders other than k resulting from k entering the auction. The amount each bidder is charged $\sum_{i:i \neq k} b_i(A_i^{-k}) - \sum_{i:i \neq k} b_i(A_i)$ is a generalisation of Vickrey's second price auction.

13.2.2 Example

Consider the following auction with 2 items and 4 bidders, with the following valuations:

	$\{A\}$	$\{B\}$	$\{A,B\}$
b_1	0	0	5
b_2	2	0	0
b_3	0	1	0
b_4	0	1	1

It is easy to see that the allocation $(A_1, A_2, A_3, A_4) = (\{A, B\}, \emptyset, \emptyset, \emptyset)$ maximises $\sum_i b_i(A_i)$ (note that social welfare and fairness are not necessarily aligned).

Now we calculate the amount that each bidder will be charged. Since bidder 1 is the only player receiving any items, they will be the only one charged. Now, $\sum_{i:i\neq 1} b_i(A_i) = b_2(A_2) + b_3(A_3) + b_4(A_4) = 0$. Next we compute an allocation (A_2, A_3, A_4) that maximises $\sum_{i:i\neq 1} b_i(A_i^{-1})$. If we exclude player 1 from the auction, then $(A_2^{-1}, A_3^{-1}, A_4^{-1}) = (\{A\}, \emptyset, \{B\})$, and $b_2(\{A\}) + b_3(\emptyset) + b_4(\{B\}) = 2 + 0 + 1 = 3$, so player 1 must pay $\sum_{i:i\neq 1} b_i(A_i^{-k}) - \sum_{i:i\neq 1} b_i(A_i) = 3 - 0$.

Theorem 13.4. *The VCG mechanism is incentive compatible and social welfare maximising – making bidders pay their externalities aligns their individual incentives with social welfare maximisation.*

Proof. Fix bidder k. The utility of bidder k under allocation A_k is:

$$u_k(A_k) = v_k(A_k) - \left(\sum_{i:i\neq k} b_i(A_i^{-k}) - \sum_{i:i\neq k} b_i(A_i) \right)$$

Observation: the sum $\sum_{i:i\neq k} b_i(A_i^{-k})$ does not depend on bidder k. Hence bidder k wants to maximise $v_k(A_k) + \sum_{i:i\neq k} b_i(A_i)$.

Claim 13.5. *Bidding $b_k = v_k$ is a weakly dominant strategy for k.*

Proof. If k bids $b_k = v_k$ hen the VCG mechanism produces the allocation $(A_1, \ldots, A_n)$ that maximises $\sum_i b_i(A_i) = b_k(A_k) + \sum_{i:i\neq k} b_i(A_i) = v_k(A_k) + \sum_{i:i\neq k} b_i(A_i)$. This is equal to the value that k is trying to maximise, hence the VCG mechanism is working in k's favour, so telling the truth will give k the best outcome. ■

The VCG mechanism maximises social welfare by definition. □

13.3 VCG mechanism's efficiency

13.3.1 Auctions with single-minded bidders

In a combinatorial auction with single-minded bidders, each bidder is interested only in one subset of item. Each bidder therefore needs only to submit (S_i, b_i), where b_i is the bid for $S_i \subseteq S$, and $v_i(S') = 0$ for all $S' \subseteq S$, $S' \neq S_i$. If bidder i gets assigned its desired set S_i, it is *a* winner.

Consider the following example with 5 items and 7 single-minded bidders:

$$\begin{aligned} v_1(\{A, C, D\}) &= 7 \\ v_2(\{B, E\}) &= 7 \\ v_3(\{C\}) &= 3 \\ v_4(\{A, B, C, D\}) &= 9 \\ v_5(\{D\}) &= 4 \\ v_6(\{A, B, C\}) &= 5 \\ v_7(\{B, D\}) &= 5 \end{aligned}$$

We can allocate the items to players 1 and 2, which allocates all items and achieves a social welfare of 14, or to players 2, 3, and 5, which allocates all but item A and achieves the same social welfare.

Theorem 13.6. *Finding an optimal allocation to bidders in the VCG mechanism (even in the case of single-minded bidders) is NP-hard.*

First, recall the problems MaxIndependentSet: given a graph $G = (V, E)$, we wish to compute a subset of maximum cardinality $S \subseteq V$ such that for all $u, v \in V$, $(u, v) \notin E$.

Proof. We will prove it by constructing a polynomial-time reduction from MaxIndependentSet in graphs. To prove that computing an allocation (CAwSMB) is NP-hard, we have to show that, given an instance of MaxIndependentSet, we can transform it into an instance of our combinatorial auction (i.e., we are showing that MaxIndependentSet $\leq^P$ CAwSMB).

Let $G = (V, E)$ be an undirected graph. Construct a combinatorial auction by setting the bidders as the vertices, edges as the items, and bid for player i as $(\{e \in E : e \text{ is incident to } v\}, 1)$.

A set $W \subseteq V$ is a set of winners iff W is an independent set in G. Similarly, a set $W \subseteq V$ is social welfare maximising iff W is a maximum independent set. This reduction can obviously be done in polynomial time. □

The VCG mechanism has good theoretical properties such as dominant strategy incentive compatibility, however (as far as we know so far) it is computationally intractable. The death blow for the VCG mechanism comes in the following theorem.

Theorem 13.7. *For every small constant $\varepsilon > 0$, computing an $|E|^{\frac{1}{2}-\varepsilon}$-approximate (i.e. anything better than $\sqrt{|E|}$) solution of* MaxIndependentSet *cannot be done in polynomial time, under standard complexity-theoretic assumptions (e.g., that NP $\not\subseteq$ BPP).*

The best hope we can have is to design an efficient auction mechanism that produces $\sqrt{|S|}$-approximate allocations of items to bidders. If the sizes of sets that bidders are interested in is at most 2, then the problem is easy: it is simply a matching problem. If they are at least 3 however, hardness kicks in and the problem is intractable.

Fact. If we replace optimal allocations in the VCG mechanism by approximate ones then the resulting mechanism is not incentive compatible.

13.4 A greedy mechanism for auctions with single-minded bidders

In this section we design a greedy mechanism for computing optimal allocations for auctions with single-minded bidders. The advantage of such a mechanism is that it is computationally efficient, incentive compatible. A possible drawback is that it "only" computes a $\sqrt{|S|}$-approximate social welfare maximising allocation.

An auction mechanism is *monotone* if any bidder who bids (S_i, b_i) and wins (i.e. is allocated S_i) also wins if they bid (A, w), with $A \subseteq S_i$ and $w \geq b_i$ (they win if they bid more for less).

Definition 13.8 (Critical value pricing)**.** In a mechanism with critical value pricing, a winner i who bids (S_i, b_i) pays the minimum price needed to win (the infimum of all w such that the bid (S_i, w) would also win).

Lemma 13.9. *A mechanism for auctions with single-minded bidders in which losers pay 0 is incentive compatible iff it is monotone and it uses critical value pricing.*

The greedy mechanism:

- Winner determination:
  - Sort bids and rename such that $\frac{b_1}{\sqrt{|S_1|}} \geq \ldots \geq \frac{b_n}{\sqrt{|S_n|}}$
  - initialise $W \leftarrow \emptyset$
  - for $i = 1, \ldots, n$, if $S_i \cap (\bigcup_{j \in W} S_j) = \emptyset$, then $W = W \cup \{i\}$ (i.e. if all of i's desired items are available)
- Critical value pricing
  - winner i pays $\frac{v_j}{\sqrt{|S_j|/|S_i|}}$, where j is the smallest index greater than i such that for all $k < j$,

either $k = i$ or $S_k \cap S_j = \emptyset$

- winner i pays 0 if no such j exists

The above winner-determination rule is monotone as increasing bid b_i or decreasing S_i will make the ratio $\frac{b_i}{\sqrt{|S_i|}}$ larger. Similarly, the payment rule indeed uses critical-value pricing as i wins iff i appears before j.

Theorem 13.10. *The greedy mechanism is incentive compatible and produces a $\sqrt{|S|}$-approximation allocation w.r.t. social welfare.*

Proof. The mechanism's incentive compatibility follows from Lemma 13.9, which states that a mechanism for auctions with single-minded bidders in which losers pay 0 is DSIC iff it is monotone and it uses critical-value pricing. To prove the approximation bound, we want to show:

$$\sum_{i \in W^*} v_i \leq \sum_{i \in W} v_i$$

where W^* is the set of winners in the welfare-maximising solution, and W is the set of winners produced by the greedy solution.

Let $W_i^* = \{j \in W^*, j \geq i \ : \ S_i \cap S_j \neq \emptyset\}$, i.e. bidders that win in an optimal allocation but cannot in the greedy auction as their desired set conflicts with that of bidder i, who appears before them. For $j \in W_i^*$, we have $b_j \leq \sqrt{|S_j|} \cdot \frac{b_i}{\sqrt{|S_i|}}$. Hence,

$$\sum_{j \in W_i^*} b_j \leq \frac{b_i}{\sqrt{|S_i|}} \cdot \sum_{j \in W_i^*} \sqrt{|S_j|}$$

Claim 13.11. $\sum_{j \in W_i^*} \sqrt{|S_j|} \leq \sqrt{|S_i|} \cdot \sqrt{|S|}$

Proof. The Cauchy-Schwarz inequality states that $(\sum_{k=1} x_k y_k)^2 \leq \sum_k x_k^2 \cdot \sum_k y_k^2$. Choosing $x_k = 1$ and $y_k = \sqrt{|S_k|}$, we get

$$(\sum_{k=1} \sqrt{|S_k|})^2 \leq |W_i^*| \cdot \sum_{j \in W_i^*} |S_j|$$

Taking the square root of both sides gives us

$$\sum_{k=1} \sqrt{|S_k|} \leq \sqrt{|W_i^*|} \cdot \sqrt{\sum_{j \in W_i^*} |S_j|}$$

Observe that $|W_i^*| \leq |S_i|$, as for every $j \in W_i^*$, S_j intersects S_i and all of these intersections are disjoint. We also have $\sum_{j \in W_i^*} |S_j| \leq |S|$. ■

Using this claim and summing over all $i \in W$, we get

$$\begin{aligned}
\sum_{i \in W} \sum_{j \in W_i^*} b_j &\leq \sum_{i \in W} \frac{b_i}{\sqrt{|S_i|}} \cdot \sum_{j \in W_i^*} \sqrt{|S_j|} \\
&\leq \sum_{i \in W} \frac{b_i}{\sqrt{|S_i|}} \cdot \sqrt{|S_i|} \cdot \sqrt{|S|} \\
&\leq \sqrt{|S|} \cdot \sum_{i \in W} b_i
\end{aligned}$$

As $W^* \subseteq \bigcup_{i \in W} W_i^*$, we have that $\sum_{i \in W} \sum_{j \in W_i^*} b_j$ is an upper bound of the optimal solution welfare. □

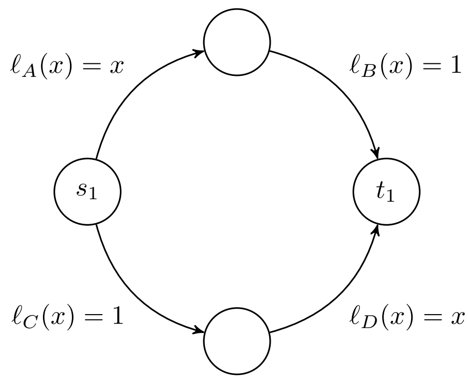
We conclude that the greedy auction mechanism is incentive compatible and approximates the optimal social welfare within a factor of $\sqrt{|S|}$. Since computing an allocation of items to bidders that gives a $|S|^{\frac{1}{2}-\varepsilon}$-approximation in polynomial time under reasonable complexity theoretic assumptions is not possible by Theorem 13.7, this is essentially the best bound we can hope for.

14 Network Routing games

We turn our attention now to network flows, and begin by defining the following model. Let $G = (V, E)$ be a directed graph with:

- k source-destination pairs (s_i, t_i), each with demand $r_i \in \mathbb{R}_{\geq 0}$.
- latency functions $\ell_e : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ for all edges, mapping traffic to latency
- P_i, the set of paths from s_i to t_i for each source-destination pair
- Let $P := \bigcup^k P_i$
- Flow $f : P \to \mathbb{R}_{\geq 0}$ such that $\sum_{p \in P_i} f(p) = r_i$
- Flow on edge e, $f_e := \sum_{p \in P_i : e \in p} f(p)$
- Latency of path p $L_p(f) := \sum_{e \in p} \ell_e(f_e)$
- Social cost of flow $C(f) = \sum_{p \in P} f(p) \cdot L_p(f) = \sum_{e \in E} f_e \cdot \ell_e(f_e)$

Consider the following example:



We have:

- $\ell_A(x) = x$, $\ell_B(x) = 1$, $\ell_C(x) = 1$, $\ell_D(x) = x$
- $P = P_1 = \{(A, B), (C, D)\}$

Call (A, B) the "up" path, and (C, D) the "down" path. Consider the flow f for $\lambda \in [0, 1]$ where $f^\lambda(\text{up}) = \lambda$ and $f^\lambda(\text{down}) = 1 - \lambda$. We have the following flows on edges:

$$f_A^\lambda = f_B^\lambda = \lambda$$
$$f_C^\lambda = f_D^\lambda = 1 - \lambda$$

The latencies of the paths are:

$$L_{\text{up}}(f^\lambda) = \ell_A(f_A^\lambda) + \ell_B(f_B^\lambda) = \ell_A(\lambda) + \ell_B(\lambda) = \lambda + 1$$
$$L_{\text{down}}(f^\lambda) = \ell_C(f_C^\lambda) + \ell_D(f_D^\lambda) = \ell_C(1 - \lambda) + \ell_D(1 - \lambda) = 1 + 1 - \lambda = 2 - \lambda$$

The social cost of flow f^λ is hence given by:

$$\begin{aligned} C(f^\lambda) &= \sum_{p \in P} f^\lambda(p) \cdot L_p(f^\lambda) \\ &= f^\lambda(\text{up}) \cdot L_{\text{up}}(f^\lambda) + f^\lambda(\text{down}) \cdot L_{\text{down}}(f^\lambda) \\ &= \lambda(1 + \lambda) + (1 - \lambda)(2 - \lambda) \\ &= 2(\lambda^2 - \lambda + 1) \end{aligned}$$

The cost of this flow is minimised by setting $\lambda = 1/2$, which gives a social cost of $C(f^{\frac{1}{2}}) = 2(\frac{1}{2}^2 - \frac{1}{2} + 1) = 3/2$.

14.1 Non-atomic flow games

So far there has been no game to play – we could simply compute the cost-minimising flow and route the traffic as such. However, in selfish routing we assume that the agents are responsible for routing the traffic, each trying to minimise his own latency. In this model, there are:

- infinitely many player $i \in [0, r]$, each controlling an infinitesimal amount of traffic
- each player aims to minimise their own latency

Wardrop's First Principle: if a route is used, then it is at least as good as all other routes (otherwise they would simply take another better route).

Definition 14.1 (Wardrop Equilibrium)**.** Flow f is a Wardrop Equilibrium (WE) if for every $i \in [k]$, for every pair of paths $p, p' \in P_i$, if $f(p) > 0$ then $L_p(f) \leq L_{p'}(f)$. In other words, whenever there is some flow on some path p, then all other alternative paths p' have at least the same latency.

Fact. Every flow network has a Wardrop Equilibrium.

14.1.1 Example

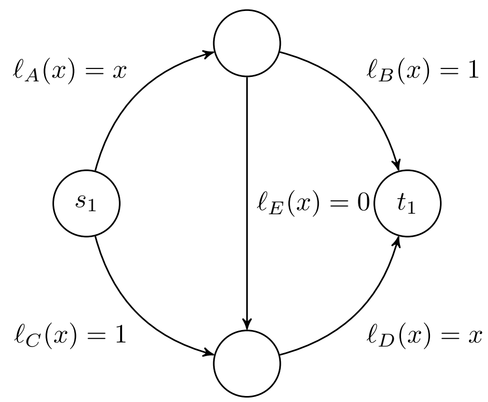
Claim 14.2. *The only Wardrop Equilibrium in the previous game is the flow* $f^{\frac{1}{2}}$

Proof. First observe that setting $\lambda = 1/2$ is indeed an equilibrium: the flow on the "up" path is $\lambda + 1 = 3/2$, equal to the flow on the "down" path, $2 - \lambda = 3/2$. Therefore no player has the incentive to deviate to the other path, as it would not decrease the latency they experience.

If $\lambda > 1/2$ then the flow on the upper path is non-zero. But $L_{\text{up}}(f^\lambda) = 1 + \lambda > 3/2$, while $L_{\text{down}}(f^\lambda) = 2 - \lambda < 3/2$, so $L_{\text{up}}(f^\lambda) > L_{\text{down}}(f^\lambda)$, meaning players on the top path would want to move to the bottom path.

If $\lambda < 1/2$ then the flow on the bottom path is non-zero. Then we have $L_{\text{down}}(f^\lambda) > 3/2$ and $L_{\text{up}}(f^\lambda) < 3/2$, so players on the bottom path would want to switch to the top path. Hence the flow $f^{\frac{1}{2}}$ is the only equilibrium. $\square$

14.1.2 Braess' Paradox



We have installed a bridge E that incurs no latency when travelling over it, that is, $\ell_E(x) = 0$.

We have the paths $P = \{(A, B), (A, E, D), (C, D)\}$ which we will label "up", "bridge", and "down" respectively. We consider the flow $f^{\overline{\lambda}} = f^{\lambda_1, \lambda_2, \lambda_3}$ that sends λ_1 units of flow through the top path, λ_3 units through the bottom path, and the remaining $\lambda_2 = 1 - \lambda_1 - \lambda_3$ units through the bridge path. The flows are:

$$\begin{aligned}
f_A^{\overline{\lambda}} &= \lambda_1 + \lambda_2 \\
f_B^{\overline{\lambda}} &= \lambda_1 \\
f_C^{\overline{\lambda}} &= \lambda_3 \\
f_D^{\overline{\lambda}} &= \lambda_2 + \lambda_3 \\
f_E^{\overline{\lambda}} &= \lambda_2
\end{aligned}$$

The latencies are:

$$\begin{aligned}
L_{\text{up}}(f^{\overline{\lambda}}) &= \lambda_1 + \lambda_2 + 1 = 2 - \lambda_3 \\
L_{\text{down}}(f^{\overline{\lambda}}) &= 1 + \lambda_2 + \lambda_3 = 2 - \lambda_1 \\
L_{\text{bridge}}(f^{\overline{\lambda}}) &= \lambda_1 + 2\lambda_2 + \lambda_3 = 2 - \lambda_1 - \lambda_3
\end{aligned}$$

The minimum social cost is again $3/2$, achieved by flow $f^{\frac{1}{2},0,\frac{1}{2}}$. However, the only Wardrop Equilibrium in this game is the flow $f^{0,1,0}$, where everyone uses the zero-cost bridge.

Claim 14.3. *The only Wardrop Equilibrium in the game is the flow* $f^{0,1,0}$.

Proof. The flow $f^{0,1,0}$ is a Wardrop Equilibrium as $L_{\text{up}}(f^{0,1,0}) = L_{\text{down}}(f^{0,1,0}) = L_{\text{bridge}}(f^{0,1,0}) = 2$.

Now consider the cases where $\lambda_1, \lambda_3 \neq 0$. In either case some flow is sent along the "bridge" path. If only $\lambda_1 > 0$:

- there is some flow sent along the "up" path, and none along the "down" path
- $L_{\text{up}}(f^{\overline{\lambda}}) = 2 > 2 - \lambda_1 = L_{\text{bridge}}(f^{\overline{\lambda}})$
- so players have incentive to deviate to taking the "bridge" path

If only $\lambda_3 > 0$:

- there is some flow sent along the "down" path, and none along the "up" path
- $L_{\text{down}}(f^{\overline{\lambda}}) = 2 > 2 - \lambda_3 = L_{\text{bridge}}(f^{\overline{\lambda}})$
- so players have incentive to deviate to taking the "bridge" path

If both $\lambda_1, \lambda_3 > 0$, then all paths are used, and $L_{\text{bridge}}(f^{\overline{\lambda}}) < L_{\text{up}}(f^{\overline{\lambda}}), L_{\text{down}}(f^{\overline{\lambda}})$.

Hence whenever $\lambda_1, \lambda_3 > 0$ players always have an incentive to deviate to the "bridge" path. Therefore the only Wardrop Equilibrium is the flow $f^{0,1,0}$.

□

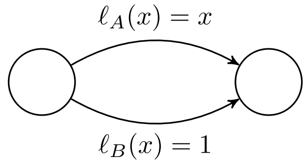
Definition 14.4 (Price of Anarchy)**.** The Price of Anarchy (POA) of a network G is the maximum ratio of the cost of the worst possible Wardrop flow and the cost of the optimal feasible flow. That is,

$$POA(G) = \frac{\max_{f \text{ is WE}} C(f)}{\min_{f^* \text{ is flow}} C(f^*)} \tag{43}$$

Fact. The Price of Anarchy in Braess' Network is $\frac{2}{3/2} = 4/3$.

So the Wardrop Equilibrium flow is 1/3 worse than the optimal in Braess' Network. This example is interesting as it shows that adding links that seem to be beneficial may actually degrade performance if the traffic is controlled by selfish players. We will now cover an even simpler example where the Price of Anarchy can be made arbitrarily bad.

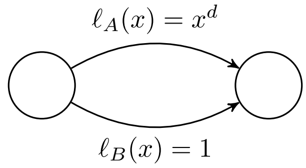
14.1.3 Pigou's Network



The latency functions are $\ell_A(x) = x$, $\ell_B(x) = 1$. Consider the flow f:

$$\begin{aligned} f^\varepsilon : A &\mapsto 1-\varepsilon \\ B &\mapsto \varepsilon \end{aligned}$$

No matter the value of ε, everyone always wants to take the top path: sending ε along edge B incurs a latency of 1, but a latency of $1-\varepsilon$ along edge A. Hence the only Wardrop Equilibrium is f^0, whose social cost is 1. The minimum social cost is achieved by the flow $f^{\frac{1}{2}}$, which incurs a social cost of 3/4. Hence the Price of Anarchy of the above network is $\frac{1}{3/4} = 4/3$.



Now consider the same network with latency functions $\ell_A(x) = x^d$ for $d \in [0,1]$, and $\ell_B(x) = 1$. The flow f^0 is still the only Wardrop Equilibrium with a social cost of $C(f^0) = 1$.

The cost of sending ε units of flow along B and the remainder along A is

$$C(f^\varepsilon) = (1-\varepsilon)(1-\varepsilon)^d + 1\cdot\varepsilon = \varepsilon + (1-\varepsilon)^{d+1}$$

This is minimised when the derivative is 0:

$$\begin{aligned} 1-(d+1)(1-\varepsilon^*)^d &= 0 \\ \varepsilon^* &= 1 - \frac{1}{\sqrt[d]{d+1}} \end{aligned}$$

The cost of this flow is:

$$\begin{aligned} C(f^{\varepsilon^*}) &= \varepsilon^* + (1-\varepsilon^*)^{d+1} \\ &= 1 - \frac{d}{(d+1)\sqrt[d]{d+1}} \end{aligned}$$

As $d \to \infty$, $C(f^{\varepsilon^*}) \to 0$. Therefore the Price of Anarchy ($\frac{\text{stable}}{\text{optimal}}$) is unbounded.

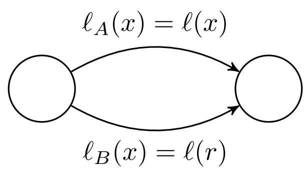
14.1.4 Pigou Bound

The latency functions used in Pigou's example are very steep. Can we derive better bounds on the Price of Anarchy by restricting the kind of latency functions that are allowed in the network?

Definition 14.5 (Pigou Bound)**.** Let $\mathcal{C}$ be a set of latency functions. The Pigou Bound $\pi(\mathcal{C})$ is defined as follows:

$$\pi(\mathcal{C}) = \sup_{\ell\in\mathcal{C}} \sup_{\varepsilon\in[0,r]} \frac{r\cdot\ell(r)}{\varepsilon\cdot\ell(\varepsilon) + (r-\varepsilon)\cdot\ell(r)} \tag{44}$$

This value is the worst Price of Anarchy possible in a network with two parallel edges, one with a constant latency function and the other with a latency function drawn from a set $\mathcal{C}$. The idea is to take the worst possible latency function $\ell \in \mathcal{C}$ and the worst possible demand r. One of the two edges has constant latency equal to $\ell(r)$, and the other's latency is given by the latency function ℓ.



Theorem 14.6. *If a class $\mathcal{C}$ of latency functions contains all constant functions, then $POA(\mathcal{C}) \leq \pi(\mathcal{C})$.*

Recall that the Price of Anarchy can be made arbitrarily large, i.e. $POA(\mathcal{C}) \geq \pi(\mathcal{C})$

Corollary. The worst POA in Pigou's Network is equal to the worst POA in all graphs.

Theorem 14.7 (Variational Inequality Characterisation)**.** *The flow f is a Wardrop Equilibrium iff for every feasible flow f^* we have:*

$$\sum_{e \in E} f_e \cdot \ell_e(f_e) \leq \sum_{e \in E} f_e^* \cdot l_e(f_e) \tag{45}$$

That is, the social cost of f is at most the social cost of f^* in the network where each edge has constant latency function $\ell_e(f_e)$.

Proof. Begin by writing the right hand side of the inequality in terms of paths rather than edges and define $H_f(f^*)$ as:

$$\begin{aligned} H_f(f^*) &= \sum_{e \in E} f_e^* \cdot \ell_e(f_e) \\ &= \sum_{i \in [k]} \sum_{p \in P_i} f^*(p) \cdot L_p(f) \end{aligned}$$

Note that the right hand side is equal to $H_f(f^*)$ while the left hand side is equal to $H_f(f)$. $\square$