EC Questions

Francesco Corda

August 10, 2020

Contents

Chapter 1

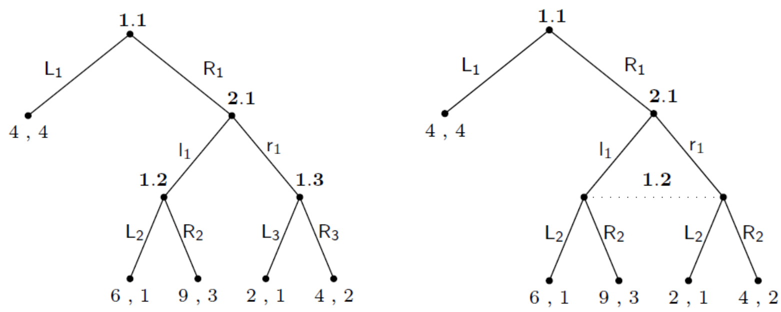
EC1

1.1 EC 1.1

Exercise 1.0.1 (Information set). *What is an information set?*

An information set h of player i is a subset of V_i such that, for all $w, w' \in h$ the property $\rho(w) = \rho(w')$ holds (with an abuse of notation, we denote $\rho(h)$ as $\rho(w)$ with $w \in h$).

Exercise 1.0.2 (Perfect vs. imperfect information). *Report an example of 2-player game in extensive-form representation with perfect information and an example of 2-player game with imperfect information.*



(a) With perfect information (b) Without perfect information

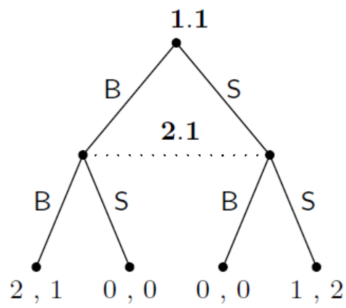
Exercise 1.0.3 (Perfect recall). *Report the definition of perfect-recall games in terms of constraints over the information sets of the players.*

Player i has perfect recall in an imperfect-information extensive-form game if for any two decision nodes w, w' that are in the same information set h for player i, for any path $\langle w_0, a_0, w_1, a_1, w_2, \ldots, w_k, a_k, w \rangle$ from the root $\bar{w}_0$ of the game to w (where w_j are decision nodes of player i and a_j are actions played at w_j by player i) and for any path $\langle w'_0, a'_0, w'_1, a'_1, w'_2, \ldots, w'_l, a'_l, w' \rangle$ from the root of the game to w' it must be the case that:

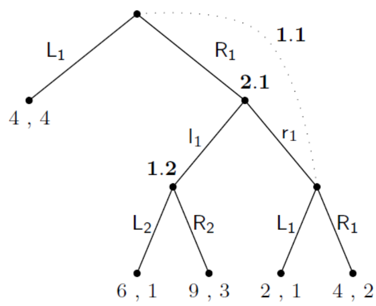
- $k = l$;
- for all $0 \leqslant j \leqslant k$, w_j and w'_j are in the same information set for player i;
- for all $0 \leqslant j \leqslant k$, it holds $a_j = a'_j$;

A game is with perfect recall if every player has perfect recall in it.

Exercise 1.0.4 (Perfect vs. imperfect recall). *Report an example of 2-player game in extensive-form representation with perfect recall and an example of 2-player game with imperfect recall.*



(a) With perfect recall

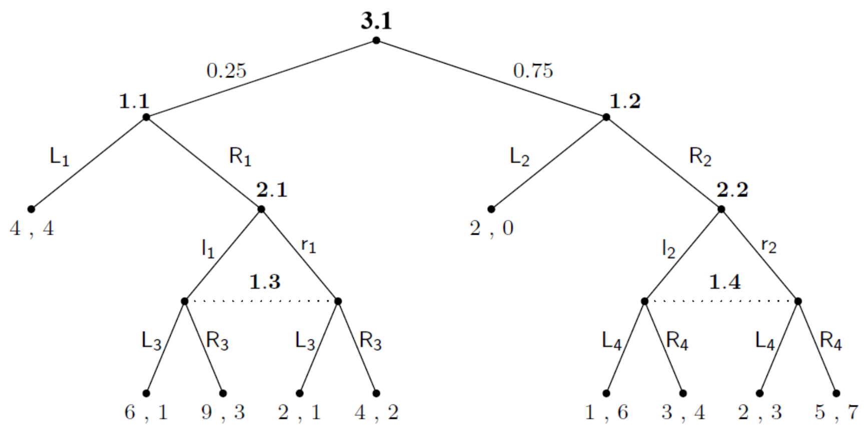


(b) Without imperfect recall

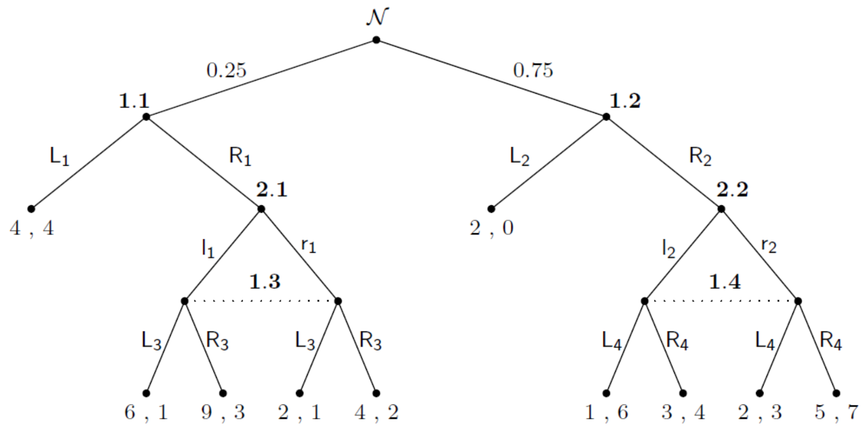
Exercise 1.0.5 (Timeability). *Report the definition of timeable extensive-form game.*

A game is timeable, in the sense that all the players have the sense of time, if and only if all the information sets are chronologically ordered.

Exercise 1.0.6 (Timeable vs. non-timeable). *Report an example of 3-player game in extensive-form representation with perfect recall in which some player has not the sense of time.*



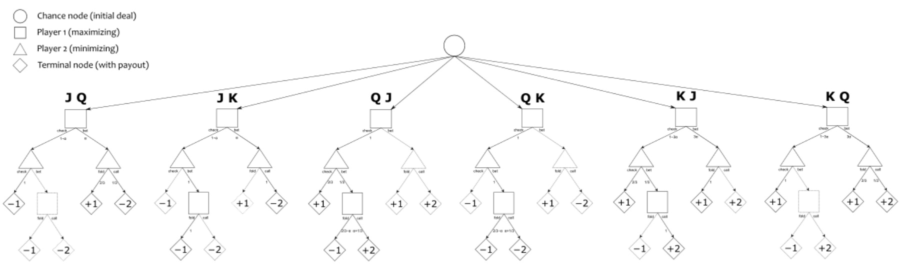
Exercise 1.0.7 (Game with Nature). *Report an example of 2-player game in extensive-form representation with Nature.*



Exercise 1.0.8 (Kuhn Poker). *Consider the Kuhn Poker game:*

- *there are two players {1, 2};*
- *there are three cards {K, Q, J};*
- *each player antes 1;*
- *each player is dealt one of the three cards, and the third is put aside unseen (no player can observe the cad of the opponent);*
- *player 1 can check or bet 1;*
  - *if player 1 checks then player 2 can check or bet 1;*
    - *if player 2 checks there is a showdown for the pot of 2;*
    - *if player 2 bets then player 1 can fold or call;*
      - *if player 1 folds then player 2 takes the pot of 3;*
      - *if player 1 calls there is a showdown for the pot of 4;*
  - *if player 1 bets then player 2 can fold or call;*
    - *if player 2 folds then player 1 takes the pot of 3;*
    - *if player 2 calls there is a showdown for the pot of 4;*
- *in the showdown, the player with the highest card wins the pot entirely.*

Provide the extensive-form representation of the game.



Exercise 1.0.9 (Simplified bargaining game). *Consider the following simplified bargaining game:*

- *there are two players, one buyer b and one seller s;*
- *the seller player is of two types:* $\theta_{s.1}$ *with a probability of 0.33 and* $\theta_{s.2}$ *with a probability of 0.67;*
- *the first player can offer either 0.33 or 0.66 and the action is perfectly observable;*
- *the second player can accept the offer x of the first player, counteroffer* $x \pm 0.20$*, and this action is perfectly observable;*
  - *if the second player counteroffers, then the first player can accept the offer x of the second player, counteroffer* $x \pm 0.10$*, and this action is perfectly observable;*
    - *if the first player counteroffers, then the second player can accept or reject;*
      - *if the second player accepts, the game concludes with an agreement over the last offer x;*
      - *if the second player rejects, the game concludes with a disagreement;*
    - *if the first player accepts, the game concludes with an agreement over the last offer x;*
  - *if the second player accepts, then the game concludes with an agreement over the last offer x;*
- *the utility of all the players from a disagreement is 0, while the utility of buyer from an agreement (x, t) where t is the time at which the agreement is achieved, is* $(1-x)(\delta_b)^t$*, while the utility of seller s.i is* $(\delta_{s.i})^t$*. Assume that* $\delta_b = 0.5, \delta_{s.1} = 0.4, \delta_{s.2} = 0.9$*. Each action requires a unitary temporal cost.*

Provide the extensive-form representation of the game in both cases the first player is the seller (and the second is the buyer) and the first player is the buyer (and the second is he seller).

TODO

Exercise 1.0.10 (Simplified patrolling game). *Consider the following simplified patrolling game:*

- *there are two players, one attacker a and one defender d;*
- *there is a fully connected graph with four vertices labeled* $\{v_1, v_2, v_3, v_4\}$*;*
- *the defender is initially at* v_1 *and spends one time point to move from any vertex to any another vertex;*
- *at time 1, the attacker decides which vertex to attack among* $\{v_2, v_3, v_4\}$ *and moves to it, simultaneously the defender decides the next vertex to visit among* $\{v_2, v_3, v_4\}$*;*

- *at time 2, the defender covers the vertex which it got and decides the next vertex to visit among all the vertices not visited yet;*
- *at time 3, the defender covers he vertex which it got;*
- *if the defender covers the vertex under attack, then the utility of the defender is 1, while the utility of the attacker is 0, else the utility of the defender is $1 - \pi(v_i)$ where v_i is the attacked target, while the utility of the attacker is $\pi(v_i)$.*

Provide the extensive-form representation of the game.

TODO

1.2 EC 1.2

Exercise 2.0.1 (Normal-form representation of an extensive-form game). *Provide the definition of the normal-form representation of an extensive-form game (even in the case games with Nature).*

Given an extensive-form game $(N, A, V, T, \iota, \rho, \chi, U, H)$, the corresponding normal-form representation is a triplet (N, P, U'), in which:

- $P = \{P_1, P_2, \ldots, P_n\}$ is the set of actions (said plans) of the players and each plan $p \in P_i$ is a tuple specifying one action $a \in A_i$ per information set $h \in H_i$ of player i such that $a \in \rho(h)$
- $U' = \{U_1, U_2, \ldots, U_n\}$ is the set of utility functions of the players with $U_i' : P_1 \times P_2 \times \ldots P_n \to \mathbb{R}$ such that $U_i'(p_1, p_2, \ldots, p_n) = U_i(w)$ where w is the terminal node reached by applying plan profile $(p_1, p_2, \ldots, p_n)$.

Exercise 2.0.2 (Normal-form size). *Given an extensive-form game with 2 players, h information sets per player and 2 actions available to each player at each information set, what is, in the worst case, the asymptotical size of the normal-form representation in h?*

The normal-form representation of a dynamic game may have a size that is exponentially large w.r.t. the size of the game tree. The size of the normal-form representation, in terms of number of outcomes, is 2^h (?).

Exercise 2.0.3 (Strategy and strategy profile). *Provide the definition of strategy and strategy profile for a game in normal-form representation.*

Strategy $\sigma_i : A_i \to [0, 1]$ with $\sigma_i \in \Delta(A_i)$ is a function returning the probability with which each action $a_i \in A_i$ is played by player i (we denote with $\Delta(\cdot)$ the simplex over $\cdot$).
Strategy profile σ is a tuple $(\sigma_1, \sigma_2, \ldots, \sigma_n)$, containing one strategy per player.
Strategy profile σ_{-i} is a tuple $(\sigma_1, \sigma_2, \ldots, \sigma_{i-1}, \sigma_{i+1}, \ldots, \sigma_n)$, containing one strategy per player except for player i.

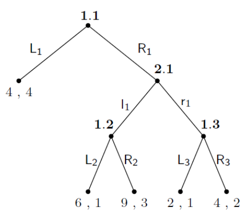
Exercise 2.0.4 (Reduced normal-form representation of an extensive-form game). *Provide the definition of the reduced normal-form representation of an extensive-form game.*

Given the normal-form representation with plans $P_1, P_2, \ldots, P_n$ of an extensive-form game, the reduced normal-form representation is composed of a subset of plans $P'_1, P'_2, \ldots, P'_n$ with $P'_i \subseteq P_i$ such that:

- no $p_i, p'_i \in P'_i$ with $p_i \neq p'_i$ are realization equivalent;
- every $p_i \in P_i \backslash P'_i$ is realization equivalent to some $p'_i \in P'_i$

Exercise 2.0.5 (Reduced normal-form size). *Given an extensive-form game with 2 players, h information sets per player and 2 actions available to each player at each information set, what is, in the worst case, the asymptotical size of the reduced normal-form representation in h?*

Exercise 2.0.6 (Translation). *Given an extensive-form game (even with Nature), provide the corresponding normal-form representation and the corresponding reduced normal-form representation.*



(a) Extensive-form

1 \ 2	l_1	r_1
$L_1L_2L_3$	$4,4$	$4,4$
$L_1L_2R_3$	$4,4$	$4,4$
$L_1R_2L_3$	$4,4$	$4,4$
$L_1R_2L_3$	$4,4$	$4,4$
$R_1L_2L_3$	$6,1$	$2,1$
$R_1L_2R_3$	$6,1$	$4,2$
$R_1R_2L_3$	$9,3$	$2,1$
$R_1R_2R_3$	$9,3$	$4,2$

(b) Normal-form

1 \ 2	l_1	r_1
$L_1 * *$	$4,4$	$4,4$
$R_1L_2L_3$	$6,1$	$2,1$
$R_1L_2R_3$	$6,1$	$4,2$
$R_1R_2L_3$	$9,3$	$2,1$
$R_1R_2R_3$	$9,3$	$4,2$

(c) Reduced normal-form

Exercise 2.0.7 (Expected utility). *Given a game in normal-form with n players, provide the formula of the expected utility for a player i.*

Expected utility $\mathbb{E}_{\mathbf{a}\sim\sigma}[U_{i(\mathbf{a})}]$ returns the expected value of the utility of player i given strategy profile σ. The formula $\mathbb{E}_{\mathbf{a}\sim\sigma}[U_i(\mathbf{a})]$ can be written as:

$$\mathbb{E}_{\mathbf{a}\sim\sigma}[U_i(\mathbf{a})] = \sum_{a_1\in A_1}\sum_{a_2\in A_2}\cdots\sum_{a_n\in A_n}\sigma_1(a_1)\sigma_2(a_2)\ldots\sigma_n(a_n)U_i(a_1, a_2, \ldots, a_n)$$

The degree of the polynomial is n and, given player i, the expected utility is linear in player i's strategy.

1.3 EC 1.4

Exercise 4.0.1 (Sequence-form representation of an extensive-form game). *Provide the definition of the sequence-form representation of an extensive-form* *game (even in the case games with Nature).*

Given an extensive-form game $(N, A, V, T, \iota, \rho, \chi, U, H)$, the corresponding sequence-form representation is a tuple (N, Q, U', C), where:

- N is the set of agents;
- $Q = \{Q_1, Q_2, \ldots, Q_n\}$ is the set of sequences of all the players and Q_i is the set of sequences of player i
- $U' = \{U'_1, U'_2, \ldots, U'_n\}$ is the set of utility functions of all the players where $U'_i : Q_1 \times Q_2 \times \ldots \times Q_n \to \mathbb{R}$ returns the utility $U_i(w)$ of the terminal node w reached by a profile of terminal sequences, while it is not defined if the profile of sequences contains at least a non-terminal sequence;
- $C = \{(F_1, f_1), (F_2, f_2), \ldots, (F_n, f_n)\}$ is the set of the constraints over the sequence-form strategies of all the players.

Exercise 4.0.2 (Sequence-form size). *Given an extensive-form game with 2 players, h information sets per player and 2 actions available to each player at each information set, what is, in the worst case, the asymptotical size of the sequence-form representation in h?*

TODO

Exercise 4.0.3 (Strategy and strategy profile). *Provide the definition of strategy and strategy profile for a game in sequence-form representation.*

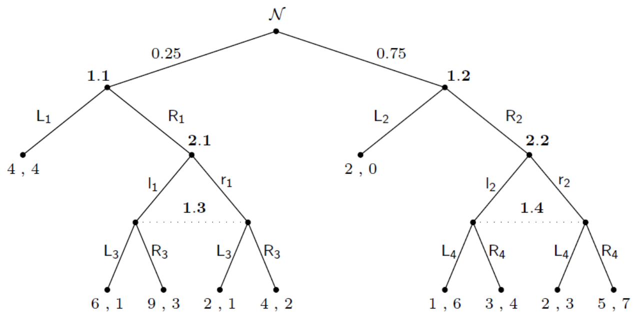
A sequence-form strategy (said realization plan) $r_i : Q_i \to [0, 1]$, with the constraints that $r_i(q_\varnothing) = 1$ and that $r_i(q) = \sum_{a \in \rho(h)} \mathsf{extend}(q, a)$ for each $h \in \mathsf{lead}(q)$ and for every $q \in Q_i$ that is not terminal, is a function returning the probability with which each sequence $q \in Q_i$ is played by player i.

Strategy profile $\mathbf{r}$ is a tuple $(r_1, r_2, \ldots, r_n)$, containing one strategy per player. Strategy profile $\mathbf{r}_{-i}$ is a tuple $(r_1, r_2, \ldots, r_{i-1}, r_{i+1}, \ldots, r_n)$, containing one strategy per player except for player i. Differently from strategies in the normal-form representation, in the sequence-form representation, strategies may not satisfy the condition that $\sum_{g \in Q_t} r_i(q) = 1$ (that is, a realization plan may be not a probability distribution). Indeed, the sequence-form representation requires different constraints.

The constraints over the strategies are linear in the strategies and, by using matrix-based notation, such constraints can be formulated as $F_i r_i = f_i$, where F_i is a matrix $\mathcal{M}$ with size $(|H_i| + 1) \times |Q_i|$, r_i is here intended as a column vector, and f_i is a column vector of $|H_i| + 1$ positions.

Exercise 4.0.4 (Translation). *Given an extensive-form game (even with Nature), provide the corresponding sequence-form representation.*

(Sequence-form representation with Nature). An example of extensive-form game with Nature and its sequence-form representation is reported below (we use symbol '$\times$' to denote the cases in which the utility functions are not defined).



$$U_1 = \begin{bmatrix} \times & \times & \times & \times & \times \\ 1.00 & \times & \times & \times & \times \\ 1.50 & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & 1.50 & 0.50 & \times & \times \\ \times & 2.25 & 0.75 & \times & \times \\ \times & \times & \times & 0.75 & 1.50 \\ \times & \times & \times & 2.25 & 3.75 \end{bmatrix}, \quad U_2 = \begin{bmatrix} \times & \times & \times & \times & \times \\ 1.00 & \times & \times & \times & \times \\ 0.00 & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & 0.25 & 1.50 & \times & \times \\ \times & 0.75 & 1.00 & \times & \times \\ \times & \times & \times & 4.50 & 2.25 \\ \times & \times & \times & 3.00 & 5.25 \end{bmatrix}.$$

Notice that the sequence-form representation has a size, in terms of variables and constraints, that is linear in the size, in terms of terminal nodes, of the game tree.

Exercise 4.0.5 (Expected utility). *Given a game in sequence-form with n players, provide the formula of the expected utility for a player i.*

Expected utility $\mathbb{E}_{\mathbf{q}\sim\mathbf{r}}[U_i(\mathbf{q})]$ returns the expected value of the utility of player i given strategy profile $\mathbf{r}$. The formula $\mathbb{E}_{\mathbf{q}\sim\mathbf{r}}[U_i(\mathbf{q})]$ can be written as:

$$\mathbb{E}_{\mathbf{q}\sim\mathbf{r}}[U_i(\mathbf{q})] = \sum_{q_1 \in Q_1} \sum_{q_2 \in Q_2} \dots \sum_{q_n \in Q_n} r_1(q_1)\, r_2(q_2) \dots r_n(q_n)\, U_i(q_1, q_2, \dots, q_n)$$

once assigned $U_i(\mathbf{q}) = 0$ for all the profile $\mathbf{q}$ containing at least one non-terminal sequence. The degree of the polynomial is n and, given player i, the expected utility is linear in player i's strategy.

1.4 EC 1.5

Exercise 5.0.1 (Kuhn theorem). *Provide the statement of the Kuhn theorem.*

Given every game with perfect recall and every normal-form strategy, there always exists at least a realization-equivalent agent-form strategy and vice versa.

1.5 EC 1.6

Exercise 6.0.1 (Realization equivalence). *Provide the conditions under which two strategies (even defined over different representations of the same game) are realization equivalent.*

(Realization equivalence). Given an extensive-form game, two strategies, even defined on different representations, are realization equivalent if, for every strategy of the opponents, they induce the same probability distribution over the outcomes of the game.

(Sequence/normal-form strategies realization equivalence). Given a game and a player i, a sequence-form strategy r_i and a normal-form strategy σ_i are realization equivalent if and only if the following property holds for every $q \in Q_i$:

$$r_i(q) = \sum_{p \in P_i : \mathsf{action}(q) \in p} \sigma_i(p)$$

The above relation provides a direct way to find a sequence-form strategy given a normal-form strategy. The reverse is more involved, since every sequence-form strategy may be realization equivalent to many normal-form strategies and therefore it is necessary the resolution of a linear equation system.

(Sequence/reduced-normal-form strategies realization equivalence). Every pure sequence-form strategy corresponds to a single plan in the reduced normal form.

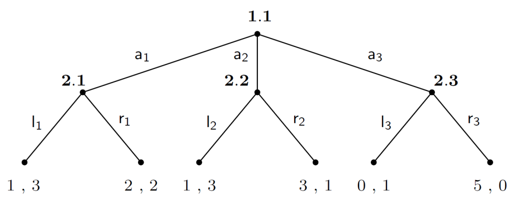
(Normal-form strategy derivation from a sequence-form strategy). Given a game and a sequence-form strategy r_i, the reduced normal-form strategy σ_i defined as

$$\sigma_i(p) = \prod_{q \in \bar{Q}_i : \mathsf{action}(q) \in p} r_i(q)$$

for every $p \in P_i$, is realization equivalent to r_i^1.

Exercise 6.0.2 (From sequence-form to normal-form). *Given an extensive-form game, its normal-form and sequence-form representations, and a strategy profile defined over the sequence-form, provide a strategy defined over the normal-form that is realization equivalent. Is the latter unique?*

Consider the following game in extensive form



its corresponding normal-form representation (the reduced normal form is the same):

2

1		$\mathsf{l}_1\mathsf{l}_2\mathsf{l}_3$	$\mathsf{l}_1\mathsf{l}_2\mathsf{r}_3$	$\mathsf{l}_1\mathsf{r}_2\mathsf{l}_3$	$\mathsf{l}_1\mathsf{r}_2\mathsf{r}_3$	$\mathsf{r}_1\mathsf{l}_2\mathsf{l}_3$	$\mathsf{r}_1\mathsf{l}_2\mathsf{r}_3$	$\mathsf{r}_1\mathsf{r}_2\mathsf{l}_3$	$\mathsf{r}_1\mathsf{r}_2\mathsf{r}_3$
	a_1	$1,3$	$1,3$	$1,3$	$1,3$	$2,2$	$2,2$	$2,2$	$2,2$
	a_2	$1,3$	$3,1$	$3,1$	$3,1$	$1,3$	$1,3$	$3,1$	$3,1$
	a_3	$0,1$	$5,0$	$0,1$	$5,0$	$0,1$	$5,0$	$0,1$	$5,0$

6-1

and its corresponding sequence-form representation:

2

1		l_1	r_1	l_2	r_2	l_3	r_3
	a_1	$1,3$	$2,2$	$\times,\times$	$\times,\times$	$\times,\times$	$\times,\times$
	a_2	$\times,\times$	$\times,\times$	$1,3$	$3,1$	$\times,\times$	$\times,\times$
	a_3	$\times,\times$	$\times,\times$	$\times,\times$	$\times,\times$	$0,1$	$5,0$

(Normal-form strategy derivation from a sequence-form strategy). Consider the following sequence-form strategy:

$$r_2 = \begin{bmatrix} 1.00 \\ 0.50 \\ 0.50 \\ 0.30 \\ 0.70 \\ 0.25 \\ 0.75 \end{bmatrix}$$

In the realization-equivalent strategy σ_2, we have, e.g., plan $p = \mathsf{l}_1\mathsf{l}_2\mathsf{l}_3$ played with probability $\sigma_2(\mathsf{l}_1\mathsf{l}_2\mathsf{l}_3) = 0.50 \cdot 0.30 \cdot 0.25$

Exercise 6.0.3 (From normal-form to sequence-form). *Given an extensive-form game, its normal-form and sequence-form representations, and a strategy profile defined over the normal-form, provide a strategy defined over the sequence-form that is realization equivalent. Is the latter unique?*

(Sequence/normal-form strategies realization equivalence). Consider the example above. Strategies r_2 and σ_2 are realization equivalent if the following equations are satisfied:

$$\begin{aligned}
r_2(\mathrm{l}_1) &= \sigma_2(\mathrm{l}_1\mathrm{l}_2\mathrm{l}_3) &&+ \sigma_2(\mathrm{l}_1\mathrm{l}_2\mathrm{r}_3) &&+ \sigma_2(\mathrm{l}_1\mathrm{r}_2\mathrm{l}_3) &&+ \sigma_2(\mathrm{l}_1\mathrm{r}_2\mathrm{r}_3) \\
r_2(r_1) &= \sigma_2(r_1 l_2 \mathrm{l}_3) &&+ \sigma_2(r_1 l_2 r_3) &&+ \sigma_2(r_1 r_2 \mathrm{l}_3) &&+ \sigma_2(r_1 r_2 r_3) \\
r_2(\mathrm{l}_2) &= \sigma_2(\mathrm{l}_1\mathrm{l}_2\mathrm{l}_3) &&+ \sigma_2(\mathrm{l}_1\mathrm{l}_2 r_3) &&+ \sigma_2(r_1\mathrm{l}_2\mathrm{l}_3) &&+ \sigma_2(r_1\mathrm{l}_2 r_3) \\
r_2(r_2) &= \sigma_2(\mathrm{l}_1 r_2\mathrm{l}_3) &&+ \sigma_2(\mathrm{l}_1 r_2 r_3) &&+ \sigma_2(r_1 r_2\mathrm{l}_3) &&+ \sigma_2(r_1 r_2 r_3) \\
r_2(\mathrm{l}_3) &= \sigma_2(\mathrm{l}_1\mathrm{l}_2\mathrm{l}_3) &&+ \sigma_2(\mathrm{r}_1\mathrm{l}_2\mathrm{l}_3) &&+ \sigma_2(\mathrm{r}_1\mathrm{r}_2\mathrm{l}_3) &&+ \sigma_2(\mathrm{l}_1\mathrm{r}_2\mathrm{l}_3) \\
r_2(\mathrm{r}_3) &= \sigma_2(\mathrm{l}_1\mathrm{l}_2\mathrm{r}_3) &&+ \sigma_2(\mathrm{l}_1\mathrm{r}_2\mathrm{r}_3) &&+ \sigma_2(\mathrm{r}_1\mathrm{r}_2\mathrm{r}_3) &&+ \sigma_2(\mathrm{r}_1\mathrm{l}_2\mathrm{r}_3)
\end{aligned}$$

Notice that $r_2(q) = \frac{1}{2}$ for every q is realization equivalent to $\sigma_2(p) = \frac{1}{8}$ for every p, but it is realization equivalent also to $\sigma_2(l_1 l_2 l_3) = \sigma_2(r_1 r_2 r_3) = \frac{1}{2}$ and $\sigma_2(p) = 0$ for all the other p.

Exercise 6.0.4 (Sequence-form and reduced normal-form). *What is the relation between sequence-form representation and reduced normal-form representation?*

(Sequence/reduced-normal-form strategies realization equivalence). Every pure sequence-form strategy corresponds to a single plan in the reduced normal form.

The observation above shows that the number of pure realization plans is exponentially large in the size of the game tree, the number of plans of the reduced normal-form being exponentially large. When using the reduced normal-form representation, we have a direct way to derive a strategy that is realization equivalent to a given realization plan.

1.6 EC 1.7

Exercise 7.0.1 (Zero-sum/constant-sum/strictly competitive games). *Provide the definition of:*

- *2-player zero-sum game;*
- *2-player constant-sum game;*
- *2-player strictly-competitive game.*

Furthermore, describe their relationships, and their properties in the players utility space.

(Zero-sum game). A zero-sum game is a game in which, for each terminal node w in the game tree (i.e., each outcome), the following property holds: $\sum_{i \in N} U_i(w) = 0$.

(Zero-sum game and utility space). Given the space of players' utilities $(U_1, U_2, \ldots, U_n)$ in $\mathbb{R}^n$, each terminal node can be mapped as a point in such a space and, in a zero-sum game, all these points lie on a hyperplane $\sum_{i \in N} U_i = 0$.

(Constant-sum game). A constant-sum game is a game in which, for each terminal node w in the game tree (i.e., each outcome), the following property holds: $\sum_{i \in N} U_i(w) = constant$.

(Constant-sum game and utility space). Given the space of players' utilities $(U_1, U_2, \ldots, U_n)$ in $\mathbb{R}^n$, each terminal node can be mapped as a point in such a space and, in a constant-sum game, all these points lie on a hyperplane $\sum_{i \in N} U_i = constant$.

(Strictly competitive game). A strictly competitive game is a 2 -player game in which, for any pair of strategy profiles $\sigma, \sigma', \mathbb{E}_{\mathbf{a} \sim \sigma}[U_1(\mathbf{a})] < \mathbb{E}_{\mathbf{a} \sim \sigma'}[U_1(\mathbf{a})]$ if and only if $\mathbb{E}_{\mathbf{a} \sim \sigma}[U_2(\mathbf{a})] > \mathbb{E}_{\mathbf{a} \sim \sigma'}[U_2(\mathbf{a})]$.
Strictly-competitive games capture the situation in which, for every given strategy profile and for any perturbation of it, a player increases her expected utility if and only if the opponent reduces her expected utility. It is known that a strictly-competitive game is just a zero-sum game once an affine transformation is applied.

(Strictly competitive game and utility space). Given the space of players' utilities (U_1, U_2) in $\mathbb{R}^2$, each terminal node can be mapped as a point in such a space and, in a strictly competitive game, all these points lie on a hyperplane $\sum_{i \in N} \alpha_i U_i = constant$, where $\alpha_i \in (0, 1]$.

(Constant-sum game as zero-sum game). A constant-sum game is equivalent to a zero-sum game, once the constant has been subtracted from the utility of a player.

(Strictly competitive and zero-sum games). Any strictly-competitive game is equivalent to a zero-sum game once an affine transformation, in principle different for each player, is applied.

Exercise 7.0.2 (Polymatrix games). *Provide the definition of Polymatrix games and the formula of players expected utility. Furthermore, provide the conditions under which a Polymatrix game is zero sum.*

A Polymatrix game in normal-form representation is a tuple (N, A, U) where:

- the definitions of N and A are standard as in normal-form games;
- $U = \{U_1, U_2, \ldots, U_n\}$ and each utility function U_i is defined as $\sum_{j \in N \setminus \{i\}} U_{i,j}$ where each $U_{i,j}$ is defined as $U_{i,j} : A_i \times A_j \to \mathbb{R}$ and therefore each $U_{i,j}$ can be described as a matrix.

(Polymatrix game and expected utility). The formula $\mathbb{E}_{\mathbf{a} \sim \mathbf{s}}[U_i(\mathbf{a})]$ in a Polymatrix game can be written as:

$$\mathbb{E}_{\mathbf{a} \sim \sigma}[U_i(\mathbf{a})] = \sum_{j \in N \setminus \{i\}} \sum_{a_i \in A_i} \sum_{a_j \in A_j} \sigma_i(a_i) \sigma_j(a_j) U_{i,j}(a_i, a_j)$$

The degree of the polynomial is 2 (independently from n) and, given player i, the expected utility is linear in player i's strategy.

(Zero-sum Polymatrix games). A Polymatrix game with players N is zero sum if for every $a \in A_1 \times A_2 \times \ldots \times A_n$ we have $\sum_{i \in N} U_i(a) = 0$.

(Zero-sum and zero-sum pairwise Polymatrix games). Any zero-sum Polymatrix game is equivalent to a Polymatrix game where $U_{i,j} + U_{i,j}^T = 0$. These last games are called zero-sum pairwise Polymatrix games.

1.7 EC 1.8

Exercise 8.0.1 (Bayesian games). *Given a Bayesian game in epistemic-form, provide the representation in extensive-form.*

A simultaneous-moves Bayesian game in epistemic-form representation is a tuple $(N, A, \Theta, \Omega, U)$ where:

- $N = \{1, 2, \ldots, n\}$ is the set of players;

- $A = \{A_1, A_2, \ldots, A_n\}$ is the set of the actions of all the players and $A_i = \{a_1, a_2, \ldots, a_{m_i}\}$ is the set of player i's actions;
- $\Theta = \Theta_1 \times \Theta_2 \times \ldots \times \Theta_n$ is the set of the types of all the players and $\Theta_i = \{\theta_{i,1}, \theta_{i,2}, \ldots, \theta_{i,k_i}\}$ is the set of the types of player i
- $\Omega : \Theta \to \Delta(\Theta)$ returns the probability associated with each $(\theta_1, \theta_2, \ldots, \theta_n)$ where $\theta_i \in \Theta_i$
- $U = \{U_1, U_2, \ldots, U_n\}$ is the set of the utility functions of all the players and $U_i : A_1 \times A_2 \times \ldots \times A_n \times \Theta \to \mathbb{R}$ is the utility function of player i.

Epistemic-form:
(Bayesian Battle of sexes). Consider a Bayesian game composed of:

- $N = \{1, 2\}$
- $A = \{A_1, A_2\}$ with $A_1 = \{\mathrm{B}, \mathrm{S}\}, A_2 = \{\mathrm{B}, \mathrm{S}\}$
- $\Theta = \{\Theta_1, \Theta_2\}$ with $\Theta_1 = \{\theta_{1,1}, \theta_{1,2}\}, \Theta_2 = \{\theta_{2,1}, \theta_{2,2}\}$
- $\Omega = \{\Omega_1, \Omega_2\}$ with:

$$\Omega_1 = \left\{ \begin{array}{ll} 0.2 & \theta_{1.1} \\ 0.8 & \theta_{1.2} \end{array} \right. , \quad \Omega_2 = \left\{ \begin{array}{ll} 0.3 & \theta_{2.1} \\ 0.7 & \theta_{2.2} \end{array} \right.$$

- utility functions are represented by means of the following bi-matrices:

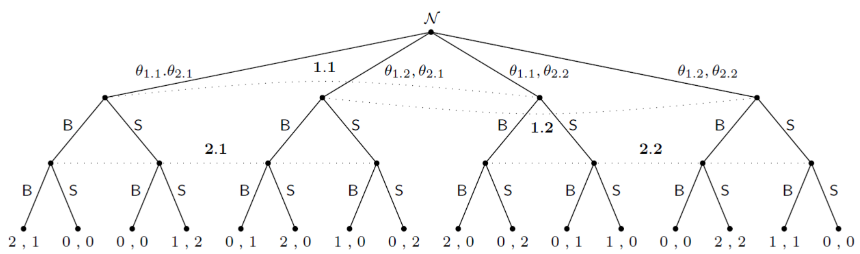
$\theta_{1.1}$ \ $\theta_{2.1}$	B	S
B	$2, 1$	$0, 0$
S	$0, 0$	$1, 2$

$\theta_{1.1}$ \ $\theta_{2.2}$	B	S
B	$2, 0$	$0, 2$
S	$0, 1$	$1, 0$

$\theta_{1.2}$ \ $\theta_{2.1}$	B	S
B	$0, 1$	$2, 0$
S	$1, 0$	$0, 2$

$\theta_{1.2}$ \ $\theta_{2.2}$	B	S
B	$0, 0$	$2, 2$
S	$1, 1$	$0, 0$

Extensive-form:



Chapter 2

EC2

2.1 EC 2.7

Exercise 7.0.1 (Maxmin strategy definition). *Provide the definition of Maxmin strategy in 2–player normal–form games.*

The problem of finding a Maxmin strategy of player i against player $-i$ is formulated as:

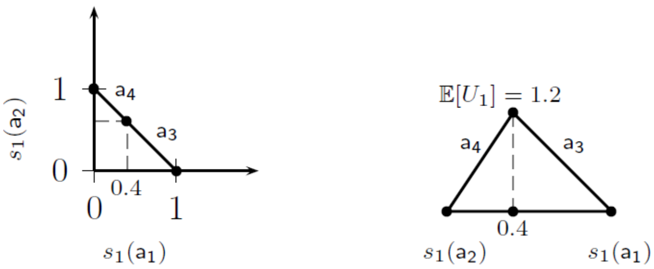
$$
\begin{aligned}
\arg\max_{\sigma_i} \quad & \min_{\sigma_{-i}} \quad \sum_{a_i \in A_i} \sum_{a_{-i} \in A_{-i}} \left[U_i\left(a_i, a_{-i}\right) \sigma_i\left(a_i\right) \sigma_{-i}\left(a_{-i}\right) \right] \\
& \quad s.t. \quad \sum_{a_{-i} \in A_{-i}} \sigma_{-i}\left(a_{-i}\right) = 1 \\
& \qquad\qquad \sigma_{-i}\left(a_{-i}\right) \geqslant 0 \quad \forall a_{-i} \in A_{-i} \\
s.t. \quad & \sum_{a_i \in A_i} \sigma_i\left(a_i\right) = 1 \\
& \sigma_i\left(a_i\right) \geqslant 0 \quad \forall a_i \in A_i
\end{aligned}
$$

Exercise 7.0.2 (Maxmin strategy example). *Given a 2–player normal–form games with 2 actions per player, find a Maxmin strategy.*

Consider the following 2-player matrix game in which player 1 plays as max player and player 2 plays as min player:

1 \ 2	a_3	a_4
a_1	0	3
a_2	2	0

We report the strategy simplex of player 1 partitioned in subareas with the corresponding action player 2 would play and subsequently how the expected utility of player 1 varies in the simplex given the strategy of player 2. The Maxmin value of player 1 is 1.2.



Exercise 7.0.3 (Complexity). *What is the computational complexity of finding the Maxmin/Minmax strategy with 2–player normal–form games?*

The Maxmin strategy and the corresponding Maxmin value of player i against player $-i$ can be found in polynomial time in the size of the game, where the size is given by the number of actions of the players. Therefore the Maxmin finding problem is in FP class.

Exercise 7.0.4 (Mathematical programming formulation). *Given a 2–player normal–form game, provide the mathematical programming formulation for finding a Maxmin/Minmax strategy and show its derivation.*

The Maxmin strategy and the corresponding value of player i against player $-i$ can be found by solving the following Linear Program (LP):

$$\begin{aligned}
\arg\max_{\sigma_i, v_j} \quad & v_i \\
\text{s.t.} \quad & v_i - \sum_{a_i \in A_i} \left[U_i\left(a_i, a_{-i}\right) \sigma_i\left(a_i\right) \right] && \leqslant 0 \quad \forall a_{-i} \in A_{-i} \\
& \sum_{a_i \in A_i} \sigma_i\left(a_i\right) && = 1 \\
& \sigma_i\left(a_i\right) && \geqslant 0 \quad \forall a_i \in A_i
\end{aligned}$$

Proof. Given the definition of the Maxmin optimization problem:

$$
\begin{array}{lllll}
\arg\max\limits_{\sigma_i} & & \min\limits_{\sigma_{-i}} & \sum\limits_{a_i \in A_i} \sum\limits_{a_{-i} \in A_{-i}} \Big[U_i(a_i, a_{-i})\, \sigma_i(a_i)\, \sigma_{-i}(a_{-i}) \Big] & \\
 & & \text{s.t.} & \sum\limits_{a_{-i} \in A_{-i}} \sigma_{-i}(a_{-i}) & = 1 \\
 & & & \sigma_{-i}(a_{-i}) & \geqslant 0 \quad \forall a_{-i} \in A_{-i} \\
\text{s.t.} & \sum\limits_{a_i \in A_i} \sigma_i(a_i) & & & = 1 \\
 & \sigma_i(a_i) & & & \geqslant 0 \quad \forall a_i \in A_i
\end{array}
$$

we substitute the inner maximization problem with its dual problem that, by strong duality, returns the same optimal value of objective function, obtaining:

$$
\begin{array}{lllll}
\arg\max\limits_{\sigma_i} & & \max\limits_{v_i} & v_i & \\
 & & \text{s.t.} & v_i - \sum\limits_{a_i \in A_i} \Big[U_i(a_i, a_{-i})\, \sigma_i(a_i) \Big] & \leqslant 0 \quad \forall a_{-i} \in A_{-i} \\
\text{s.t.} & \sum\limits_{a_i \in A_i} \sigma_i(a_i) & & & = 1 \\
 & \sigma_i(a_i) & & & \geqslant 0 \quad \forall a_i \in A_i
\end{array}
$$

Finally, joining the two max operators we obtain:

$$
\begin{array}{lll}
\arg\max\limits_{\sigma_i, v} & v_i & \\
\text{s.t.} & v_i - \sum\limits_{a_i \in A_i} \Big[U_i(a_i, a_{-i})\, \sigma_i(a_i) \Big] & \leqslant 0 \quad \forall a_{-i} \in A_{-i} \\
 & \sum\limits_{a_i \in A_i} \sigma_i(a_i) & = 1 \\
 & \sigma_i(a_i) & \geqslant 0 \quad \forall a_i \in A_i
\end{array}
$$

□

The Minmax strategy player i against player $-i$ and the corresponding value of player $-i$ can be found by solving the following Linear Program:

$$
\begin{array}{lll}
\arg\min\limits_{\sigma_i, v_j} & v_{-i} & \\
\text{s.t.} & v_{-i} - \sum\limits_{a_i \in A_i} \left[U_{-i}\left(a_i, a_{-i}\right) \sigma_i\left(a_i\right) \right] & \geqslant 0 \quad \forall a_{-i} \in A_{-i} \\
 & \sum\limits_{a_i \in A_i} \sigma_i\left(a_i\right) & = 1 \\
 & \sigma_i\left(a_i\right) & \geqslant 0 \quad \forall a_i \in A_i
\end{array}
$$

Proof. Given the definition of the Minmax optimization problem:

$$\begin{aligned}
\arg\min_{\sigma_i} \quad & \begin{aligned} \max_{\sigma_{-i}} \quad & \sum_{a_i \in A_i} \sum_{a_{-i} \in A_{-i}} \left[U_{-i}(a_i, a_{-i})\, \sigma_i(a_i)\, \sigma_{-i}(a_{-i}) \right] \\ \text{s.t.} \quad & \sum_{a_{-i} \in A_{-i}} \sigma_{-i}(a_{-i}) = 1 \\ & \sigma_{-i}(a_{-i}) \geqslant 0 \quad \forall a_{-i} \in A_{-i} \end{aligned} \\
\text{s.t.} \quad & \sum_{a_i \in A_i} \sigma_i(a_i) = 1 \\
& \sigma_i(a_i) \geqslant 0 \quad \forall a_i \in A_i
\end{aligned}$$

we substitute the inner maximization problem with its dual problem that, by strong duality, returns the same optimal value of objective function, obtaining:

$$\begin{aligned}
\arg\min_{\sigma_i} \quad & \begin{aligned} \min_{v} \quad & v_{-i} \\ \text{s.t.} \quad & v_{-i} - \sum_{a_i \in A_i} \Big[U_{-i}(a_i, a_{-i})\, \sigma_i(a_i) \Big] \geqslant 0 \quad \forall a_{-i} \in A_{-i} \end{aligned} \\
\text{s.t.} \quad & \sum_{a_i \in A_i} \sigma_i(a_i) = 1 \\
& \sigma_i(a_i) \geqslant 0 \quad \forall a_i \in A_i
\end{aligned}$$

Finally, we obtain:

$$\begin{aligned}
\arg\min_{s_i, v_{-i}} \quad & v_{-i} \\
\text{s.t.} \quad & v_{-i} - \sum_{a_i \in A_i} \Big[U_{-i}(a_i, a_{-i})\, \sigma_i(a_i) \Big] \geqslant 0 \quad \forall a_{-i} \in A_{-i} \\
& \sum_{a_i \in A_i} \sigma_i(a_i) = 1 \\
& \sigma_i(a_i) \geqslant 0 \quad \forall a_i \in A_i
\end{aligned}$$

□

Exercise 7.0.5 (Solving a game). *Given a normal–form game with 2 players, find a Maxmin strategy by means of AMPL + GUROBI.*

TODO

2.2 EC 2.8

Exercise 8.0.1 (Maxmin strategy definition). *Provide the definition of Maxmin strategy in 2–player sequence– form games.*

The Maxmin strategy of player i against player $-i$ in sequence form can be found by solving the following linear mathematical programming problem:

$$\begin{aligned}
\arg\max_{r_i, v_i} \quad & \sum_{h_{-i} \in H_{-i} \cup \{\mathsf{h}_\varnothing\}} f_{-i}(h_{-i})\, v_i(h_{-i}) \\
s.t. \quad & \sum_{h_{-i} \in H_{-i} \cup \{\mathsf{h}_\varnothing\}} F_{-i}(h_{-i}, q_{-i})\, v_i(h_{-i}) - \sum_{q_i \in Q_i} \Big[U_i(q_i, q_{-i})\, r_i(q_i) \Big] \leqslant 0 \quad && \forall q_{-i} \in Q_{-i} \\
& \sum_{q_i \in Q_i} F_i(h_i, q_i)\, r_i(q_i) = f_i(h_i) && \forall h_i \in H_i \cup \{\mathsf{h}_\varnothing\} \\
& r_i(q_i) \geqslant 0 && \forall q_i \in Q_i
\end{aligned}$$

Exercise 8.0.2 (Complexity). *What is the computational complexity of finding the Maxmin/Minmax strategy with 2–player sequence–form games?*

The Maxmin strategy and the corresponding Maxmin value of player i against player $-i$ in an extensive-form game can be found, by means of the sequence-form representation, in polynomial time in the size of the game, where the size is given by the number of nodes of the game tree. Therefore the Maxmin finding problem is in FP class.

Exercise 8.0.3 (Mathematical programming formulation). *Given a 2–player sequence–form game, provide the mathematical programming formulation for finding a Maxmin/Minmax strategy and show its derivation.*

The Minmax strategy of player i against player $-i$ in sequence form can be found by solving the following linear mathematical programming problem:

$$\begin{array}{llll}
\arg\min\limits_{r_i, v_{-i}} & \sum\limits_{h_{-i} \in H_{-i} \cup \{\mathsf{h}_\varnothing\}} f_{-i}(h)\, v_{-i}(h_{-i}) & & \\
s.t. & \sum\limits_{h_{-i} \in H_{-i} \cup \{\mathsf{h}_\varnothing\}} F_{-i}(h_{-i}, q_{-i})\, v_{-i}(h) - \sum\limits_{q_i \in Q_i} \Big[U_{-i}(q_i, q_{-i})\, r_i(q_i) \Big] & \geqslant 0 & \forall q_{-i} \in Q_{-i} \\
 & \sum\limits_{q_i \in Q_i} F_i(h_i, q_i)\, r_i(q_i) & = f_i(h_i) & \forall h_i \in H_i \cup \{\mathsf{h}_\varnothing\} \\
 & r_i(q_i) & \geqslant 0 & \forall q_i \in Q_i
\end{array}$$

Exercise 8.0.4 (Solving a game). *Given a sequence–form game with 2 players, find a Maxmin strategy by means of AMPL + GUROBI.*

TODO

2.3 EC 2.9

Exercise 9.0.1 (Team Maxmin strategy definition). *Provide the definition of the Maxmin strategy when there are multiple max players and a single min player.*

Many-max player/single-min players. Now, we focus on the case in which there are multiple max players and only one min player. Initially, we remark that this situation makes sense only when the utility functions of all the max players are the same, otherwise it would not be clear which objective function the max players should maximize. Therefore, in this case, the max players are forming a team and searching for the Maxmin strategy is equivalent to searching for a Team-Maxmin strategy. Also here, we can distinguish the case in which the team plays in correlated strategies from the case in which the team plays in mixed strategies. We initially focus on the first case.
(Maxmin optimization problem (with correlated min players)). The problem of finding a Maxmin strategy of a team of $n-1$ players, playing in correlated fashion, against player i is formulated as:

$$\begin{array}{lllll}
\arg\max\limits_{\boldsymbol{\sigma}_{-i}} & \min\limits_{\sigma_i} & \sum\limits_{\mathbf{a}_{-i} \in A_{-i}} \sum\limits_{a_i \in A_i} \Big[U_{-i}(a_i, \mathbf{a}_{-i})\, \sigma_i(a_i)\, \boldsymbol{\sigma}_{-i}(\mathbf{a}_{-i}) \Big] & & \\
 & s.t. & \sum\limits_{a_i \in A_i} \sigma_i(a_i) & = 1 & \\
 & & \sigma_i(a_i) & \geqslant 0 & \forall a_i \in A_i \\
s.t. & & & &
\end{array}$$

(Maxmin optimization problem (with non-correlated min players)). The problem of finding a Maxmin strategy of a team of $n-1$ players, playing in mixed strategies, against player i is formulated as:

$$\begin{array}{llllll}
\arg\max_{\boldsymbol{\sigma}_{-i}} & \min_{\sigma_i} & \sum_{a_i \in A_i} \sum_{\mathbf{a}_{-i} \in A_{-i}} \left[U_{-i}(a_i, \mathbf{a}_{-i}) \prod_{j \in N} \sigma_j(a_j) \right] & & \\
& s.t. & \sum_{a_i \in A_i} \sigma_i(a_i) & = 1 & \\
& & \sigma_i(a_i) & \geqslant 0 & \forall a_i \in A_i \\
s.t. & \sum_{a_j \in A_j} \sigma_j(a_j) & & = 1 & \forall j \in N \backslash \{i\} \\
& \sigma_j(a_j) & & \geqslant 0 & \forall j \in N \backslash \{i\}, \forall a_j \in A_j
\end{array}$$

Exercise 9.0.2 (Complexity). *What is the computational complexity of finding the Maxmin strategy with n–max players and 1 min player in normal–form games (equivalently finding a Minmax strategy with n–min player and 1 max player)?*

Exercise 9.0.3 (Mathematical programming formulation). *Given an n–player normal–form game, provide the mathematical programming formulation for finding a Maxmin strategy and show its derivation when there are* $n-1$ *max players and* 1 *min players.*

(Maxmin strategy formulation (with correlated min players)). The Maxmin strategy and the corresponding value of a team of $n-1$ players, playing in correlated fashion, against player i can be found by solving the following Linear Program:

$$\begin{array}{lllll}
\arg\max_{\boldsymbol{\sigma}_{-i}, v_{-i}} & v_{-i} & & & \\
s.t. & v_{-i} - \sum_{\mathbf{a}_{-i} \in A_{-i}} \left[U_{-i}(a_i, \mathbf{a}_{-i}) \, \boldsymbol{\sigma}_{-i}(\mathbf{a}_{-i}) \right] & \leqslant 0 & \forall a_i \in A_i \\
& \sum_{\mathbf{a}_{-i} \in A_{-i}} \boldsymbol{\sigma}_{-i}(\mathbf{a}_{-i}) & = 1 & \\
& \boldsymbol{\sigma}_{-i}(\mathbf{a}_{-i}) & \geqslant 0 & \forall \mathbf{a}_{-i} \in A_{-i}
\end{array}$$

Exercise 9.0.4 (Solving a game). *Given a normal–form game with n players among which* $n-1$ *are maximizers and* 1 *is a minimizer, find a Maxmin strategy by means of AMPL + BARON.*

Exercise 9.0.5 (L'ho aggiunta io). *Provide the definition and the mathematical programming formulation of the Maxmin strategy when there is a single max player and multiple min players.*

(Maxmin optimization problem (with correlated min players)). The problem of finding a Maxmin strategy of player i against $n-1$ players, playing in correlated fashion, is formulated as:

$$
\begin{array}{llll}
\arg\max\limits_{\sigma_i} & \min\limits_{\boldsymbol{\sigma}_{-i}} \sum\limits_{a_i \in A_i} \sum\limits_{\mathbf{a}_{-i} \in A_{-i}} \left[U_i(a_i, \mathbf{a}_{-i})\, \sigma_i(a_i)\, \boldsymbol{\sigma}_{-i}(\mathbf{a}_{-i}) \right] & & \\
& \quad s.t. \sum\limits_{\mathbf{a}_{-i} \in A_{-i}} \boldsymbol{\sigma}_{-i}(\mathbf{a}_{-i}) & = 1 & \\
& \quad \boldsymbol{\sigma}_{-i}(\mathbf{a}_{-i}) & \geqslant 0 & \forall \mathbf{a}_{-i} \in A_{-i} \\
s.t. & \sum\limits_{a_i \in A_i} \sigma_i(a_i) & = 1 & \\
& \sigma_i(a_i) & \geqslant 0 & \forall a_i \in A_i
\end{array}
$$

We can easily derive the mathematical program solving the above problem. (Maxmin strategy formulation (with correlated min players)). The Maxmin strategy and the corresponding value of player i against $n-1$ correlated players can be found by solving the following Linear Program:

$$
\begin{array}{llll}
\arg\max\limits_{\sigma_i, v_i} & v_i & & \\
s.t. & v_i - \sum\limits_{a_i \in A_i} \left[U_i(a_i, \mathbf{a}_{-i})\, \sigma_i(a_i) \right] & \leqslant 0 & \forall \mathbf{a}_{-i} \in A_{-i} \\
& \sum\limits_{a_i \in A_i} \sigma_i(a_i) & = 1 & \\
& \sigma_i(a_i) & \geqslant 0 & \forall a_i \in A_i
\end{array}
$$

(Maxmin strategy formulation (with correlated min players)). The problem of finding a Maxmin strategy with multiple min players playing correlated strategies is equivalent to the problem of finding a Maxmin strategy with a single min players, in which the actions available to this player are all the possible joint actions of all the min players.
Let us consider the case in which the min players play mixed strategies and therefore they cannot correlate.
(Maxmin optimization problem (with non-correlated min players)). The problem of finding a Maxmin strategy of player i against $n-1$ players, each playing independently from the others, is formulated as:

$$
\begin{array}{llll}
\arg\max\limits_{\sigma_i} & \min\limits_{\boldsymbol{\sigma}_{-i}} \sum\limits_{a_i \in A_i} \sum\limits_{\mathbf{a}_{-i} \in A_{-i}} \left[U_i(a_i, \mathbf{a}_{-i}) \prod\limits_{j \in N} \sigma_j(a_j) \right] & & \\
& \quad s.t. \sum\limits_{a_j \in A_j} \sigma_j(a_j) & = 1 & \forall j \in N \backslash \{i\} \\
& \quad \sigma_j(a_j) & \geqslant 0 & \forall j \in N \backslash \{i\}, \forall a_j \in A_j \\
s.t. & \sum\limits_{a_i \in A_i} \sigma_i(a_i) & = 1 & \\
& \sigma_i(a_i) & \geqslant 0 & \forall a_i \in A_i
\end{array}
$$

In this case, the problem of deriving a finite (in terms of variables and constraints) mathematical program is open. The crucial issue concerns the impossibility to reformulate the minimization problem since it is not linear and therefore strong duality cannot be applied.

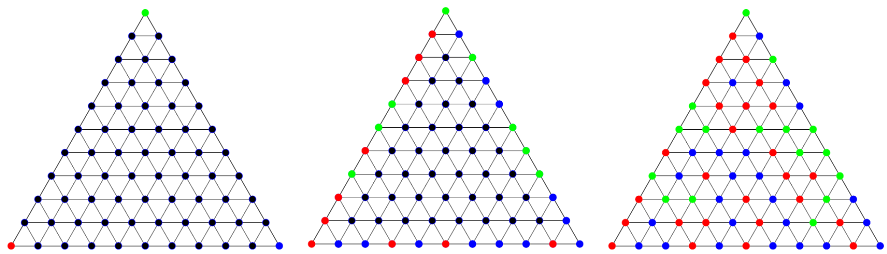
2.4 EC 2.10

Exercise 10.0.1 (Sperner's labeling). *Provide the definition of Sperner's labeling.*

(Sperner's labeling). Given a n-dimensional simplex and a simplicial subdivion $\mathcal{D}$, a Sperner's labeling is a labeling of the vertices V of the subsimplices with labels $L = \{0, 1, \ldots, n\}$ is a function $l : V \to L$ satisfying the following conditions:

- each one of the $n+1$ vertices of the simplex is labeled with a label different from the label of the others;
- the vertices of a subsimplex that lie on the facet of the simplex whose vertices have labels in $\{1, \ldots, n\}$ have label in $\{1, \ldots, n\}$ (a similar condition holds for the vertices on the other facets);
- there is no restriction on the labeling of the vertices in the interior of the simplex.

An example of Sperner's labeling of the above simplex Δ^2 is as follows:



Labeling simplex vertices — *Labeling vertices on the simplex facets* — *Labeling inner vertices*

Exercise 10.0.2 (Panchromatic and quasi–panchromatic subsimplex). *Provide the definitions of panchromatic and quasi–panchromatic n–dimension subsimplex.*

(Panchromatic (completely labeled) subsimplex). A subsimplex is called completely labeled or panchromatic if each vertex has a label different from the one of all the other vertices of the subsimplex. Therefore, a panchromatic n-dimensional subsimplex has $n + 1$ different labels.

(Quasi-panchromatic (almost completely labeled) subsimplex). An n-dimensional subsimplex is called almost completely labeled or quasi panchromatic if each vertex has a label different from the one of all the other vertices of the subsimplex except one. Therefore, a panchromatic n-dimensional subsimplex has n different labels among which one label is present twice.

Exercise 10.0.3 (Sperner's Lemma). *Provide the statement of the Sperner's Lemma.*

(Sperner's Lemma). Given a n-dimensional simplex, a simplicial subdivion $\mathcal{D}$, and a Sperner's labeling l, there exists an odd number of panchromatic n-dimensional subsimplices. Therefore, there exists at least one panchromatic subsimplex.

Exercise 10.0.4 (Sperner's walk). *Provide the definition of Spener's walk and its characterization.*

(Sperner's walk). Given a n-dimensional simplex Δ^n, a simplicial subdivion $\mathcal{D}$, and a Sperner's labeling l, a Sperner's walk is a sequence of adjacent subsimplices such that each common facet traversed to move from a subsimplex to the adjacent subsimplex has the same n different labels, that is $L \setminus \{s\}$ for some $s \in L$. Therefore, each subsimplex of the walk is either quasi panchromatic or panchromatic.

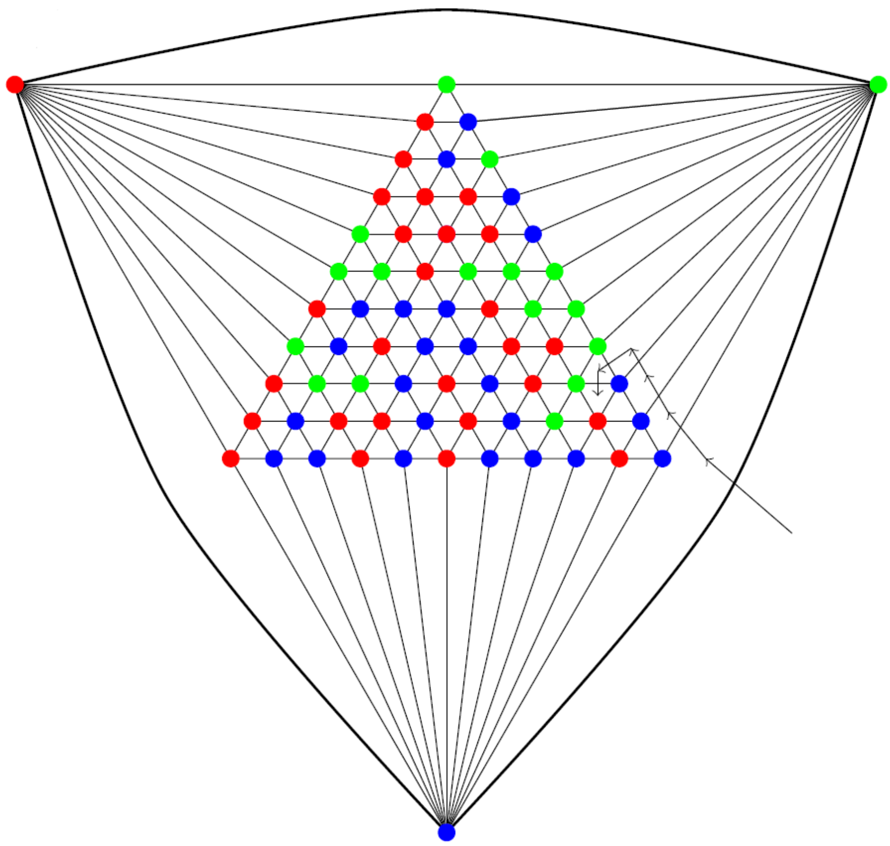
(Sperner's walk characterization). There are four main classes of Sperner's walks:

- a sequence of subsimplices starting from a subsimplex with a facet on a facet of the simplex and ending in a subsimplex with a face on the same facet of the simplex without traversing any panchromatic subsimplex or any subsimplex with a facet on of the other facets of the simplex;
- a sequence of subsimplices starting from a subsimplex with a facet on a facet of simplex and ending in a panchromatic subsimplex;
- a cycle of subsimplices without any panchromatic subsimplex;
- a sequence of subsimplices starting from a panchromatic subsimplex and ending in a panchromatic subsimplex.

Except walks strictly contained in the walks of the above four classes, no other Sperner's walks are possible.

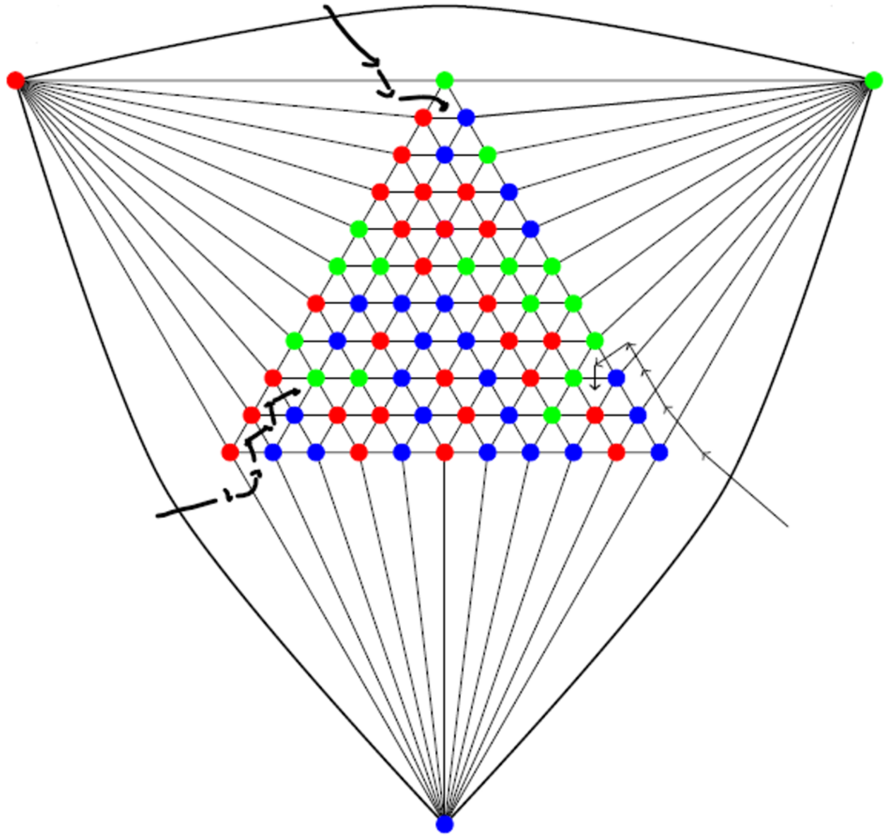
Exercise 10.0.5 (Scarf's algorithm). *Describe the Scarf's algorithm and its properties.*

(Scarf's algorithm). We describe the algorithm for the 2-dimensional case. Given a 2-dimensional simplex Δ^2 and a simplicial subdivion $\mathcal{D}$, we embed Δ^2 in a larger simplex $\bar{\Delta}^2$ as follows. Each vertex v of $\bar{\Delta}^2$ is assigned a different label and it is connected to all the vertices of the subsimplices of Δ^2 of a given facet such that one of the two labels appearing in the facet is the label of v. Two different vertices v, v' of $\bar{\Delta}^2$ are connected to the vertices of different facets of $\bar{\Delta}^2$ except for the vertices of the simplex Delta ^{2} that belong to different facets. Furthermore, the three vertices of $\bar{\Delta}^2$ are connected each other. Thus, there is a fictitious simplex composed of the whole space outer than $\bar{\Delta}^2$ that is panchromatic. This simplex is the starting point of the Scarf's algorithm.
The algorithm follows a Sperner's walk leaving this simplex (notice that three different paths can follow). Each path will end in a panchromatic subsimplex.
We report below the construction needed for the application of the Scarf's algorithm and a path generated by an execution of the algorithm, specifically when the first traversed segment has colors blue and green.



Exercise 10.0.6 (Solving 2–dimension Sperner). *Given a 2–dimension simplex with a simplicial subdivision and a Sperner's labeling, find a panchromatic subsimplex by means of Scarf's algorithm (three examples of exercises follow). Do that for all three possible initializations of the Scarf's algorithm.*

Credo...



2.5 EC 2.11

Exercise 1 (END–OF–LINE). *Provide the definition of the END–OF–LINE problem.*

(END-OF-LINE problem). END-OF-LINE is a problem characterized as follows:

- a graph composed of a potentially exponential number $O(2^n)$ of vertices labeled by a sequence of n bits;
- a function $s : \{0,1\}^n \to \{0,1\} \cup \{N\}$ polynomially computable that, given a vertex, returns the unique successor, if this exists, and N otherwise;
- a function $p : \{0,1\}^n \to \{0,1\} \cup \{N\}$ polynomially computable that, given a vertex, returns the unique successor, if this exists, and N otherwise;
- a node without predecessor;
- the question is the finding of a leaf vertex (i.e., a vertex without either a predecessor or a successor).

Exercise 2 (PPAD class). *Provide the definition of PPAD computational class and its relationship with TFNP, FNP, and FNP–completeness.*

(PPAD complexity class). The PPAD complexity class is composed of all the problems P such that there is a polynomial-time reduction from P to END-OF-LINE problem, i.e., each instance of P can be formulated as an instance of END-OF-LINE whose size is bounded by a polynomial in the size of P.
In other words, a searching problem is in PPAD whether it can be solved by means of a path-following algorithm that starts from a source and, following a path, achieves a sink corresponding to a solution of the problem itself. We focus on the problems that are the hardest ones in the PPAD class.

(PPAD completeness). The set of PPAD-complete problems is a subset of PPAD problems containing all the problems P such that there is a polynomial-time reduction from END-OF-LINE to P.
Easily, a problem is PPAD-complete when it can be reduced from the END-OF-LINE problem. Furthermore, it is commonly believed the following.

(PPAD completeness and FP). It is commonly conjectured that the class of PPAD-complete problems is not in FP.
Therefore, the common conjecture is that FP $\subset$ PPAD $\subset$ FNP. A number of fixed-point problems are PPAD complete. We cite a few of these problems.
Manca relazione con TFNP e FNP-completeness.

Exercise 3 (Nash complexity). *Describe the computational complexity of finding a Nash equilibrium.*

(PPAD-completeness of Nash equilibrium finding in 2-player games). Finding a Nash equilibrium in 2-player games is PPAD complete.
(PPAD completeness of ϵ-Nash equilibrium finding in games with 3 or more players). Finding an ϵ-Nash equilibrium in games with 3 or more players is PPAD complete.
(Pure-strategy Nash equilibrium complexity). Once n is fixed, computing a pure-strategy Nash equilibrium is in P class.

Exercise 4 (Sperner complexity). *Describe the computational complexity of finding a panchromatic subsimplex in k–dimension Sperner's problem.*

(k-dimensional Sperner's Lemma PPAD completeness). Given $k \geqslant 2$, the problem of finding a panchromatic subsimplex of Sperner's Lemma is PPAD complete.
This leads us to the following observation. That is, in the worst case, the Scarf's algorithm is the best algorithm to find a panchromatic subsimplex.
(Scarf's algorithm and Sperner's Lemma complexity). No algorithm more efficient than the Scarf's algorithm exists to find a panchromatic subsimplex of the Sperner's Lemma unless FP = PPAD.

2.6 EC 2.12

Exercise 12.0.1 (Nash equilibrium). *Provide the definition of Nash equilibrium.*

(Nash equilibrium finding). The problem of finding a Nash equilibrium can be formulated as a mathematical programming problem

$$\begin{aligned}
&\sigma_i(a_i)\left(v_i - \sum_{\mathbf{a}_{-i}\in A_{-i}}\left[U_i(a_i,\mathbf{a}_{-i})\prod_{j\in N\setminus\{i\}}\sigma_j(a_j)\right]\right) && = 0 \quad \forall i \in N, a_i \in A_i\\
&v_i - \sum_{\mathbf{a}_{-i}\in A_{-i}}\left[U_i(a_i,\mathbf{a}_{-i})\prod_{j\in N\setminus\{i\}}\sigma_j(a_j)\right] && \geqslant 0 \quad \forall i \in N, a_i \in A_i\\
&\sum_{a_i\in A_i}\sigma_i(a_i) && = 1 \quad \forall i \in N\\
&\sigma_i(a_i) && \geqslant 0 \quad \forall i \in N, a_i \in A_i
\end{aligned}$$

whose nature is Non-Linear Complementarity Constraint (NLCP).
In the special case of 2 players, we have that the program reduced to a Linear Complementarity Program (LCP) as follows.

(Nash equilibrium finding with 2 players (LCP)). The problem of finding a Nash equilibrium with 2 players can be formulated as a mathematical programming problem

$$\begin{aligned}
&\sigma_i(a_i)\left(v_i - \sum_{a_{-i}\in A_{-i}}\left[U_i(a_i,a_{-i})\,\sigma_{-i}(a_{-i})\right]\right) && = 0 \quad \forall i \in N, a_i \in A_i\\
&v_i - \sum_{a_{-i}\in A_{-i}}\left[U_i(a_i,a_{-i})\,\sigma_{-i}(a_{-i})\right] && \geqslant 0 \quad \forall i \in N, a_i \in A_i\\
&\sum_{a_i\in A_i}\sigma_i(a_i) && = 1 \quad \forall i \in N\\
&\sigma_i(a_i) && \geqslant 0 \quad \forall i \in N, a_i \in A_i
\end{aligned}$$

whose nature is Linear Complementarity (LCP).
Let us remark that the program keeps to be nonlinear, variables $v_i and \sigma_i$ being multiplied together. However, an LCP have better properties than a generic NLCP. In particular, the solutions of an LCP have rational values, while the solution sof NLCP may have irrational values.

Exercise 12.0.2 (Nash equilibrium example). *Given a 2–player normal–form game with 2 actions per player, find all the Nash equilibria.*

TODO

Exercise 12.0.3 (Mathematical programming formulation for normal–form games). *Given an 2–player normal–form game, provide the MILP/MINLP program for finding a Nash equilibrium and show its derivation.*

(Nash equilibrium finding with 2 players (MILP)). The problem of finding a Nash equilibrium with 2 players can be formulated as a mathematical programming problem

$$\begin{aligned}
&\sigma_i(a_i)\left(v_i - \sum_{a_{-i}\in A_{-i}}\Big[U_i(a_i,a_{-i})\,\sigma_{-i}(a_{-i})\Big]\right) && = 0 \quad \forall i \in N, a_i \in A_i \\
&v_i - \sum_{a_{-i}\in A_{-i}}\Big[U_i(a_i,a_{-i})\,\sigma_{-i}(a_{-i})\Big] && \geqslant 0 \quad \forall i \in N, a_i \in A_i \\
&\sum_{a_i\in A_i}\sigma_i(a_i) && = 1 \quad \forall i \in N \\
&\sigma_i(a_i) && \geqslant 0 \quad \forall i \in N, a_i \in A_i
\end{aligned}$$

whose nature is Mixed-Integer Linear (MILP); the constant M_i can be set as $M_i = \max_a\{U_i(\mathbf{a})\} - \min_a\{U_i(\mathbf{a})\}$

Proof. We just need to show that this reformulation is equivalent to the previous LCP one. The novelty resides in the adoption of binary variables b_i. The rationale is that an action a_i can be played with strictly positive probability only if $b_i\,(a_i)$ is equal to 1. This latter condition holds only the expected utility given by playing action a_i equals v_i. Indeed, if $b_i\,(a_i) = 1$, we have that the term $M_i\,(1 - b_i\,(a_i)) = 0$ and therefore $v_i = \sum_{a_{-i}\in A_{-i}} [U_i\,(a_i, a_{-i})\,\sigma_{-i}\,(a_{-i})]$. If instead $b_i\,(a_i) = 0$, the expected utility given by action a_i may be strictly smaller than v_i, since we that $v_i - \sum_{a_{-i}\in A_{-i}} [U_i\,(a_i, a_{-i})\,\sigma_{-i}\,(a_{-i})] \leqslant M_i$ that is always true by construction of M_i. In other words, the adoption of a binary $b_i\,(a_i)$ allows one to enable or disable a set of constraints.

Exercise 12.0.4 (Solving a normal–form game). *Given a normal–form game with 2 players, find a Nash equilibrium by means of AMPL + GUROBI.*

TODO

2.7 EC 2.13

Exercise 13.0.1 (Lemke–Howson's labeling). *Provide the definition of the Lemke–Howson's labeling.*

The Lemke-Howson algorithm is a path-following algorithm working in 2-player normal-form games, returning an exact Nash equilibrium. Differently from the Scarf's algorithm, the Lemke-Howson does not discretize the simplex space. Nevertheless, the Lemke-Howson algorithm is a combinatorial algorithm moving over a finite set of points. Initially, we introduce the set of labels.

(Lemke-Howson's labeling). The Lemke-Howson's labeling l is defined as follows:

$$a_i \in l(\sigma) \Longleftrightarrow \left((\sigma_i(a_i) = 0) \vee \left(v_i = \sum_{a_{-i} \in A_{-i}} U_i(a_i, a_{-i}) \sigma_{-i}(a_{-i}) \right) \right)$$

Exercise 13.0.2 (Lemke–Howson's algorithm definition). *Provide the sketch of the Lemke–Howson's algorithm.*

(Lemke-Howson (rationale)). The algorithm is structured as follows.

- Partition the simplex of player i in terms of best response of player $-i$. Each strategy profile σ corresponds to a pair of points, one in the simplex of player 1 and one in the simplex of player 2. Each strategy profile σ is associated with a number of labels according to the Lemke-Howson labeling.

- Focus on the nodes obtained by the intersections between the best response conditions and simplex border. Each node has three different labels (in degenerate games the labels may be more than three). The algorithm moves only between nodes.

- Focus on completely labeled solutions and almost completely labeled solutions. An almost completely labeled solution σ presents a label, say a, twice. One is given in the simplex of player 1, while the other is given by the simplex of player 2. Therefore, given an almost completely labeled solution σ, there are two ways to remove the label that appears twice: moving in the simplex of player 1 and moving in the simplex of player 2. This means that moving along almost completely labeled solutions the algorithm moves along paths. These paths can be either cycles or non-cycles in which the starting solution and the ending solution are completely labeled solutions.

- Start from an artificial completely labeled solution: $\sigma_i(a_i) = 0$ for every $i \in N$ and for every $a_i \in A_i$. And move along almost completely labeled solutions until find a completely labeled solution.

Exercise 13.0.3 (Lemke–Howson's algorithm application). *Given a 2–player game with 3 actions per player and the partitioning of the simplices based on the opponent's best responses, provide the labels associated with each area and apply the Lemke–Howson algorithm with three different initializations, corresponding to the three actions of player 1 (two examples of exercises follow).*

Consider the following 2-player game:

		2		
		a_4	a_5	a_6
1	a_1	$2,3$	$2,0$	$0,0$
	a_2	$0,0$	$3,3$	$2,0$
	a_3	$3,0$	$0,0$	$1,3$

and the following strategy profile:

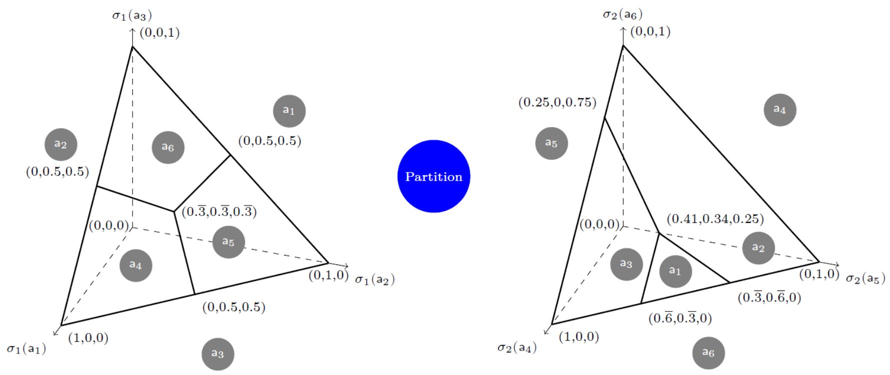
$$\sigma_1(a_1) = \begin{cases} 0.5 & \mathrm{a}_1 \\ 0.5 & \mathrm{a}_2 \\ 0.0 & \mathrm{a}_3 \end{cases} \qquad \sigma_2(a_2) = \begin{cases} 0.5 & \mathrm{a}_4 \\ 0.0 & \mathrm{a}_5 \\ 0.5 & \mathrm{a}_6 \end{cases}$$

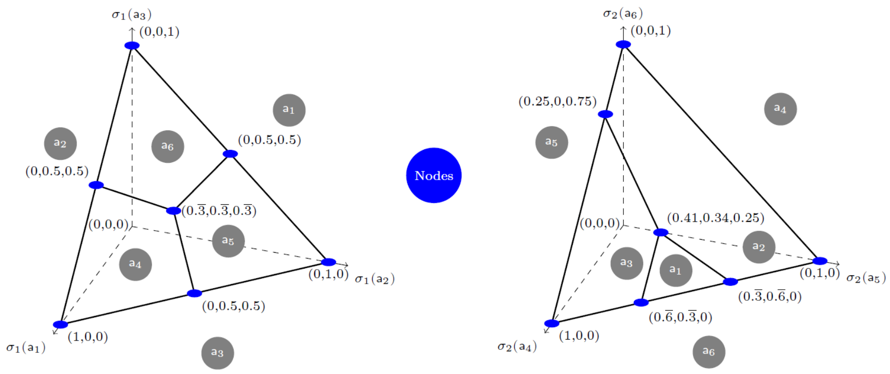
We have that $l(\sigma) = \{a_3, a_4, a_5\}$. Indeed, a_3 is the only best response of player 1 and therefore the labels of a_1 and a_2 that are played with strictly positive probability cannot be in $l(\sigma)$, while both a_4 and a_5 are best response of player 2 and therefore the label of a_6 that is played with strictly positive probability cannot be in $l(\sigma)$.

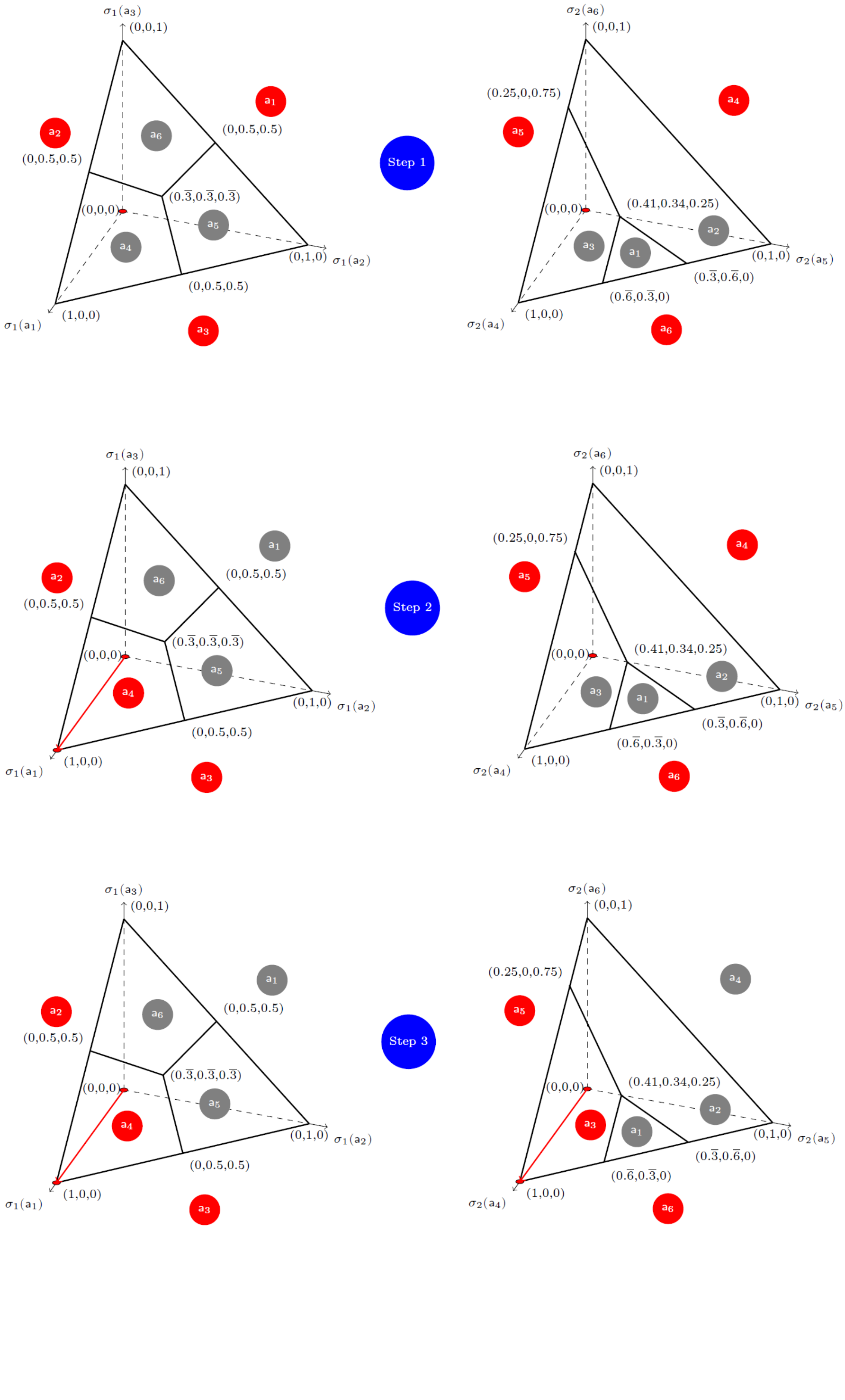
(Lemke-Howson). Consider the following 2-player game:

		2		
		a_4	a_5	a_6
1	a_1	$2,3$	$2,0$	$0,0$
	a_2	$0,0$	$3,3$	$2,0$
	a_3	$3,0$	$0,0$	$1,3$

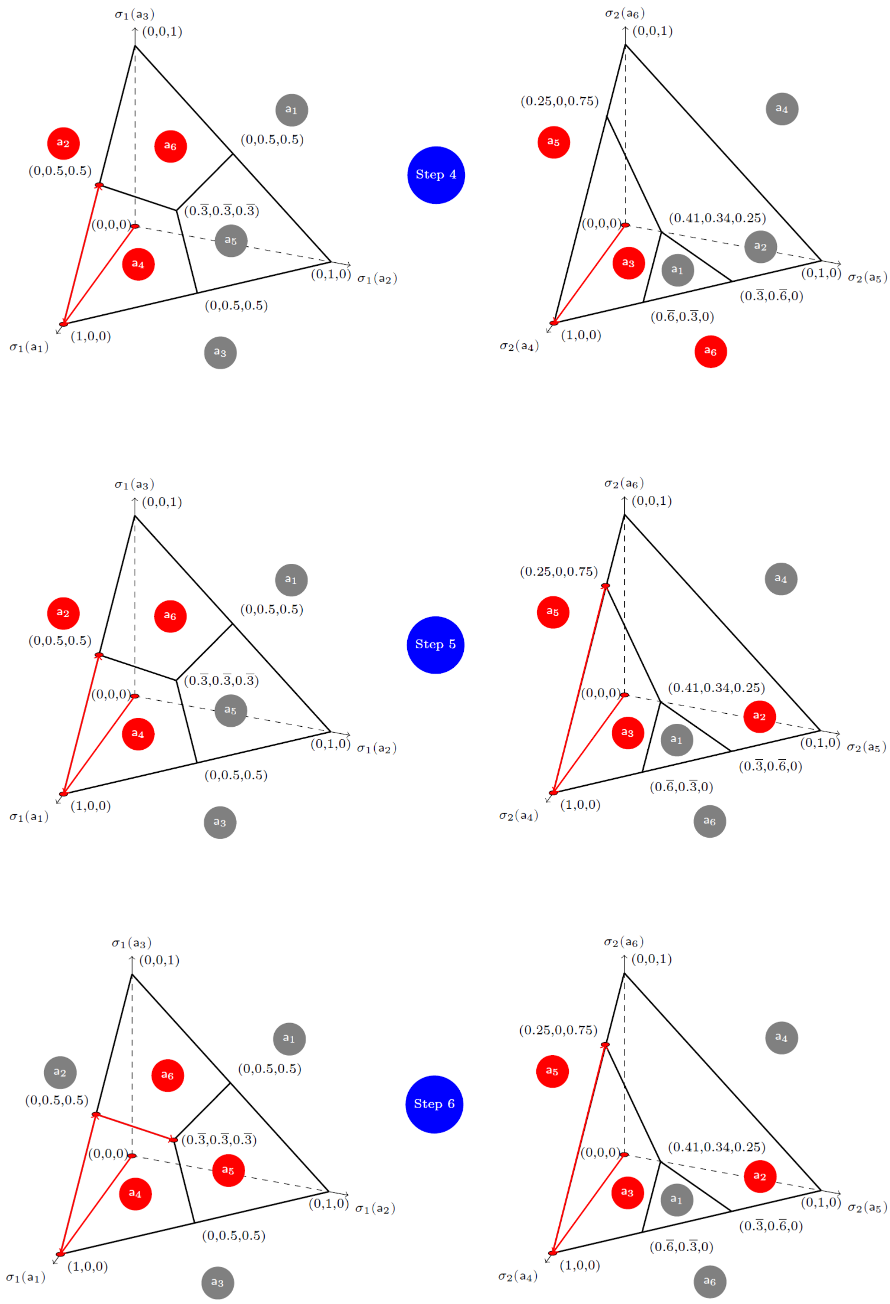
We report graphically the functioning of the algorithm.



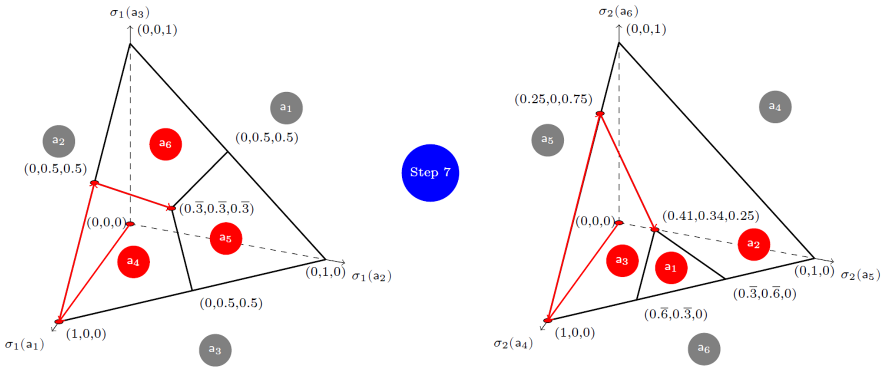




σ1(a3)
(0,0,1)
a1
a2
a6
(0,0.5,0.5)
(0,0.5,0.5)
(0.3̄,0.3̄,0.3̄)
(0,0,0)
a5
a4
(0,1,0)
σ1(a2)
(0,0.5,0.5)
σ1(a1)
(1,0,0)
a3
Step 1
σ2(a6)
(0,0,1)
(0.25,0,0.75)
a4
a5
(0,0,0)
(0.41,0.34,0.25)
a2
a3
a1
(0,1,0)
σ2(a5)
(0.3̄,0.6̄,0)
(0.6̄,0.3̄,0)
σ2(a4)
(1,0,0)
a6
Step 2
Step 3



Step 4
Step 5
Step 6
$\sigma_1(a_3)$
(0,0,1)
(0,0.5,0.5)
$(0.\overline{3},0.\overline{3},0.\overline{3})$
(0,0,0)
(0,1,0)
$\sigma_1(a_2)$
(1,0,0)
$\sigma_1(a_1)$
a_1
a_2
a_3
a_4
a_5
a_6
$\sigma_2(a_6)$
(0.25,0,0.75)
(0.41,0.34,0.25)
$(0.\overline{3},0.\overline{6},0)$
$(0.\overline{6},0.\overline{3},0)$
$\sigma_2(a_5)$
$\sigma_2(a_4)$



$\sigma_1(a_3)$
(0,0,1)
a_1
a_2
a_6
(0,0.5,0.5)
(0,0.5,0.5)
$(0.\overline{3},0.\overline{3},0.\overline{3})$
(0,0,0)
a_5
a_4
(0,1,0)
$\sigma_1(a_2)$
(0,0.5,0.5)
(1,0,0)
$\sigma_1(a_1)$
a_3
Step 7
$\sigma_2(a_6)$
(0,0,1)
(0.25,0,0.75)
a_4
a_5
(0,0,0)
(0.41,0.34,0.25)
a_2
a_3
a_1
(0,1,0)
$\sigma_2(a_5)$
$(0.\overline{3},0.\overline{6},0)$
$(0.\overline{6},0.\overline{3},0)$
(1,0,0)
$\sigma_2(a_4)$
a_6

Chapter 3

EC3

TODO

Chapter 4

EC4

TODO