COSC-6590/GSCS-6390

Games: Theory and Applications

Lecture 07 - Games in Extensive Form

Luis Rodolfo Garcia Carrillo

School of Engineering and Computing Sciences
Texas A&M University - Corpus Christi, USA

Table of contents

Motivation

Motivation

Consider a zero-sum matrix game for which

$$A = \underbrace{\begin{bmatrix} 1 & 3 & 0 \\ 6 & 2 & 7 \end{bmatrix}}_{\substack{P_2 \text{ choices} \\ \text{(Left, Middle, Right)}}} \left.\vphantom{\begin{bmatrix} 1 \\ 6 \end{bmatrix}}\right\} \begin{array}{c} P_1 \text{ choices} \\ \text{(Top, Bottom)} \end{array}$$

For this game we have

$$\underline{V}(A) = \max_j \min_i a_{ij} = 2$$

$$\bar{V}(A) = \min_i \max_j a_{ij} = 3$$

$$V_m(A) = \min_y \max_z y' A z = \max_z \min_y y' A z = \frac{8}{3} \approx 2.667$$

Motivation

Each of these values (and policies) are meaningful for a particular information structure of the game:

- $\underline{V}(A)$: outcome when the maximizer P_2 plays first (and the minimizer P_1 knows P_2's choice before selecting an action).
- $\bar{V}(A)$: outcome when the minimizer P_1 plays first (and the maximizer P_2 knows P_1's choice before selecting an action).
- $V_m(A)$: expected value of the outcome when both players play simultaneously (none knowing the others choice before selecting their actions).

However, the matrix description does not capture the information structure of the game

- in fact, **other information structures are possible**.

Extensive Form

Extensive Form Representation

An EFR of a zero-sum two-person game is a decision tree.

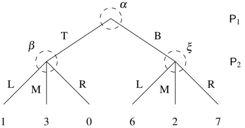
- Each **node** must be associated with one player (P_1 or P_2).
- The **links** emanating from one node correspond to decisions made by the player.
- All nodes must be enclosed in **dashed boxes**, called **information sets** (IS).
  Each IS may contain one or more nodes of the same player.
  All nodes in the same IS are indistinguishable for the corresponding player and must have the same alternatives.
- The different **leafs** of the tree are the final **outcomes** of the game and should be labeled with a cost/reward.

Extensive Form Representation

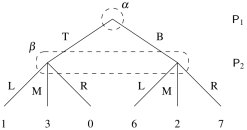
For A there is a total of 6 possible information structures

$$\underbrace{1}_{P_1 \text{ plays first}} + \underbrace{1}_{\text{simultaneous}} + \underbrace{1}_{\substack{P_2 \text{ plays first} \\ \text{with full information}}} + \underbrace{3}_{\substack{P_2 \text{ plays first} \\ \text{with partial information}}} = 6$$

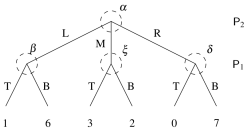
Alternative games in extensive form represented by A



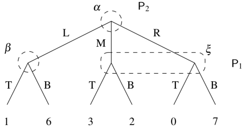
(a) $P_1 - P_2$: P_1 plays first, and P_2 plays knowing P_1's decision.



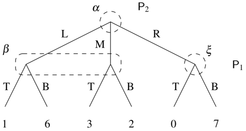
(b) Simultaneous: Both players play at the same time without knowing each others choice.



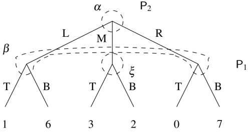
(c) $P_2 - P_1$: P_2 plays first, and P_1 plays knowing P_2's decision.



(d) Mixed: P_2 plays first, if P_2 plays L then P_1 knows about this choice, but if P_2 plays M or R, then P_1 does not know which was played.



(e) Mixed: P_2 plays first, if P_2 plays R then P_1 knows about this choice, but if P_2 plays L or M, then P_1 does not know which was played.



(f) Mixed: P_2 plays first, if P_2 plays M then P_1 knows about this choice, but if P_2 plays L or R, then P_1 does not know which was played.

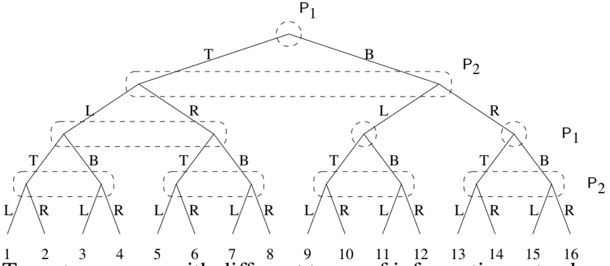
Multi-Stage Games

Multi-Stage Games

A sequence of rounds (or stages). In each stage the players have the opportunity to select one action.

EFR: tree can have more than two levels

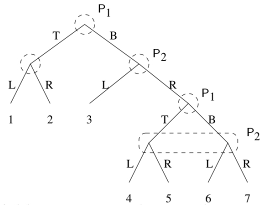
- generally twice as many levels as the number of stages.



(a) Two-stage game with different types of information sets, depending on the players' actions.

Multi-Stage Games

Some multi-stage games have a **variable number of stages**, depending on the actions of the players



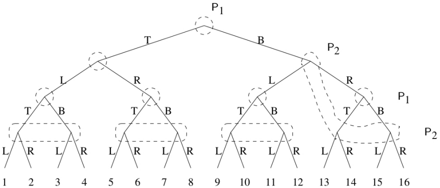
(b) Variable-stage game that may have one or two stages, depending on the actions of the players.

Example: Chess

- some actions of the players will lead to the termination of the game with very few stages, whereas other sets of actions may require a large number of stages before the game is over.

Multi-Stage Games

Information Sets may span over different stages as long as they only contain nodes of the same player and all nodes within the set exhibit exactly the same possible actions for that player.



(c) Two stage game with an information set for P_2 that spans across stages: If P_1 initially chooses B, then P_2 must make a decision without knowing the stage of play (this would be a very "memory-constrained" player).

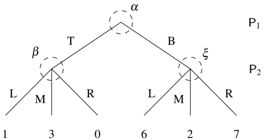
When an IS spans several stages, that particular player does not know at which stage play is taking place.

Pure Policies and Saddle-Point Equilibria

Pure Policies and Saddle-Point Equilibria

Pure policy for the player P_i: a decision rule (i.e., a function) that associates one action to each IS of this player.

Example: for the $P_1 - P_2$ game



(a) $P_1 - P_2$: P_1 plays first, and P_2 plays knowing P_1's decision.

Possible policies π_1, π_2 for P_1 and P_2, respectively

$$\pi_1(IS) = \left\{ \begin{array}{ll} T & \text{if } IS = \alpha \end{array} \right. \qquad \pi_2(IS) = \left\{ \begin{array}{ll} M & \text{if } IS = \beta \\ R & \text{if } IS = \xi \end{array} \right.$$

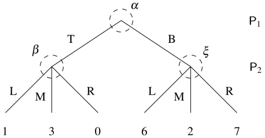
Total number of distinct pure policies for a given player is

$$\underbrace{(\#\text{ actions of 1st IS}) \times (\#\text{ actions of 2nd IS}) \times \cdots \times (\#\text{ actions of last IS})}_{\text{product over all information sets for that player}}$$

Pure Policies and Saddle-Point Equilibria

For the alternate play $P_1 - P_2$ game, we have a total of

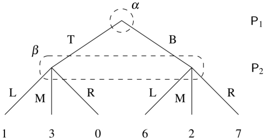
- 2 pure policies for P_1, and
- $3 \times 3 = 9$ pure policies for P_2.



(a) $P_1 - P_2$: P_1 plays first, and P_2 plays knowing P_1's decision.

For the simultaneous play game we have a total of

- 2 pure policies for P_1, and
- 3 pure policies for P_2.



(b) Simultaneous: Both players play at the same time without knowing each others choice.

Pure Policies and Saddle-Point Equilibria

Use the concepts of security levels, security policies, and saddle-point equilibria with the understanding that:

1. **action spaces:** the sets Γ_1 and Γ_2 of all pure policies for players P_1 and P_2, respectively
2. for a particular pair of pure policies $\gamma \in \Gamma_1$, $\sigma \in \Gamma_2$ we denote by $J(\gamma, \sigma)$ the **outcome of the game** when P_1 uses policy γ and P_2 uses policy σ.

Definition 7.1 (Pure saddle-point equlibrium). A pair of policies $(\gamma^*, \sigma^*) \in \Gamma_1 \times \Gamma_2$ is a pure saddle-point equilibrium if

$$J(\gamma^*, \sigma^*) \leq J(\gamma, \sigma^*) \quad \forall \gamma \in \Gamma_1$$
$$J(\gamma^*, \sigma^*) \geq J(\gamma^*, \sigma) \quad \forall \sigma \in \Gamma_2$$

and $J(\gamma^*, \sigma^*)$ is called the saddle-point value

Matrix Form for Games in Extensive Form

Matrix Form for Games in Extensive Form

A game in **extensive form** can be converted into an equivalent game in **matrix form** by regarding each policy of the game in extensive form as a possible action in a game in matrix form.

Game in extensive form, with sets as pure policies for P_1 and P_2:

$$\Gamma_1 = \{\gamma_1, \gamma_2, \ldots, \gamma_m\}, \qquad \Gamma_2 = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$$

Define an $m \times n$ matrix A_{ext}, where

1. each row is one pure policy for P_1, and m is the number of distinct pure policies in Γ_1,
2. each col is one pure policy for P_2, and n is the number of distinct pure policies in Γ_2,
3. entry a_{ij} is equal to the outcome of the game $J(\gamma_i, \sigma_j)$ when P_1 selects the pure policy γ_i (the ith row) and P_2 selects the pure policy σ_j (the jth column).

Matrix Form for Games in Extensive Form

Interpretation of matrix game A_{ext}: a game being played under **simultaneous play**

- players choose their policies before the game starts.

Recall: each row/column of A_{ext} represents the choice of a policy, not an action.

Then, we can re-write the pure saddle-point equlibrium as

$$a_{i^*j^*} \leq a_{ij^*}, \quad \forall i$$
$$a_{i^*j^*} \geq a_{i^*j}, \quad \forall j$$

- i^* is the row of A_{ext} corresponding to the policy γ^*
- j^* is the column of A_{ext} corresponding to the policy σ^*.

Conclusion: (γ^*, σ^*) is a pure saddle-point equilibrium if and only if (i^*j^*) is a pure saddle-point equilibrium for A_{ext}.

Matrix Form for Games in Extensive Form

Proposition 7.1.

The game in extensive form has a pure saddle-point equilibrium if and only if the matrix game defined by A_{ext} has a pure saddle-point equilibrium

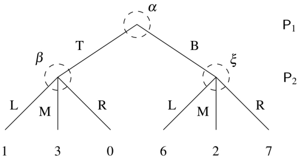
$$\underline{V}(A_{\text{ext}}) = \bar{V}(A_{\text{ext}})$$

If such equilibria exist, they have the same saddle-point value.

Note. The size of the matrix A_{ext} grows exponentially with the number of actions and information sets, which limits the use of this procedure to fairly small games.

Matrix Form for Games in Extensive Form

Example 7.1. For the $P_1 - P_2$ game in extensive form



(a) $P_1 - P_2$: P_1 plays first, and P_2 plays knowing P_1's decision.

suppose we enumerate the policies for P_1 and P_2 as

P_1 :

IS	γ_1	γ_2
α	T	B

P_2 :

IS	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	σ_7	σ_8	σ_9
β	L	L	L	M	M	M	R	R	R
ξ	L	M	R	L	M	R	L	M	R

Matrix Form for Games in Extensive Form

In this case, we obtain

$$A_{\text{ext}} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 3 & 3 & 3 & 0 & 0 & 0 \\ 6 & 2 & 7 & 6 & 2 & 7 & 6 & 2 & 7 \end{bmatrix}}_{\substack{P_2 \text{ policies} \\ (\sigma_1, \sigma_2, \ldots, \sigma_9)}} \left.\vphantom{\begin{bmatrix} 1 \\ 6 \end{bmatrix}}\right\} \begin{array}{c} P_1 \text{ policies} \\ (\gamma_1, \gamma_2) \end{array}$$

For this matrix:

- $\underline{V}(A_{\text{ext}}) = \bar{V}(A_{\text{ext}}) = 3$.
- two pure saddle-point equilibria: $(1, 4)$ and $(1, 6)$.

Consequently, the $P_1 - P_2$ game in extensive form also has two pure saddle-point equilibria: (γ_1, σ_4) and (γ_1, σ_6).

- but (γ_1, σ_6) is the dominating saddle-point equilibrium.

Matrix Form for Games in Extensive Form

Indeed

$$A_{\text{ext}} = \xrightarrow[\text{(col. 3 dominates over 1 \& 2, col. 6 dominates over 4 \& 5)}]{\text{col. 9 dominates over 7 \& 8}} A^{\dagger}_{\text{ext}} := \begin{bmatrix} 1 & 3 & 0 \\ 7 & 7 & 7 \end{bmatrix}$$

$$\xrightarrow{\text{row 1 dominates over 2}} A^{\ddagger}_{\text{ext}} := [1 \;\; 3 \;\; 0] \xrightarrow{\text{col. 2 dominates over 1 \& 3}} A^{\flat}_{\text{ext}} := [3]$$

This corresponds to the pure saddle-point equilibrium (γ_1, σ_6)

$$\gamma_1(IS) = \left\{ \; T \quad \text{if } IS = \alpha \right. \qquad \sigma_6(IS) = \begin{cases} M & \text{if } IS = \beta \;\; \text{(i.e., if } P_1 \text{ chooses T)} \\ R & \text{if } IS = \xi \;\; \text{(i.e., if } P_1 \text{ chooses B)} \end{cases}$$

However, this does not invalidate the fact that

$$\gamma_1(IS) = \left\{ \; T \quad \text{if } IS = \alpha \right. \qquad \sigma_4(IS) = \begin{cases} M & \text{if } IS = \beta \;\; \text{(i.e., if } P_1 \text{ chooses T)} \\ L & \text{if } IS = \xi \;\; \text{(i.e., if } P_1 \text{ chooses B)} \end{cases}$$

is also a (non-dominating) saddle-point equilibrium.

Recursive Computation of Equilibria for Single-Stage Games

Recursive Computation of Eq. for Single-Stage Games

Procedure starts at the bottom of the tree and moves upwards

- this one is computationally much more attractive.

Procedure for single-stage games (assume P_1 is **first-acting**)

Step 1. Construct a matrix game corresponding to each information set of P_2. Each matrix game will have

- one action of P_1 for each edge entering the information set
- one action of P_2 for each edge leaving the information set.

Step 2. Compute pure saddle-point equilibria for each of the matrix games constructed.

- method fails if none has a pure saddle-point equilibrium.

Otherwise, P_2's pure security/saddle-point policy should map to each information set the action corresponding to P_2's pure security policy in the corresponding matrix game.

Recursive Computation of Eq. for Single-Stage Games

Step 3. Replace each information set by the value of the corresponding pure saddle-point equilibrium.

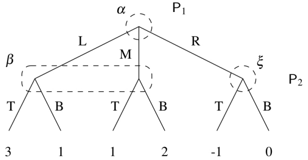
- link leading to this value should be labeled with P_1's pure policy in the corresponding matrix game.

Step 4. Player P_1 chooses the policy corresponding to the pure saddle-point value that is most favorable for P_1.

This procedure is guaranteed to result in a pure saddle-point equilibrium.

Examples illustrating how to apply this procedure

1.- Initial Game



2. Construction of matrix games for the different information sets

$$A_\beta = \underbrace{\begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}}_{\substack{P_2 \text{ choices} \\ (T,B)}} \left.\vphantom{\begin{bmatrix} 3 \\ 1 \end{bmatrix}}\right\} \begin{matrix} P_1 \text{ choices} \\ (L,M) \end{matrix} \qquad A_\xi = \underbrace{\begin{bmatrix} -1 & 0 \end{bmatrix}}_{\substack{P_2 \text{ choices} \\ (T,B)}} \Big\} \begin{matrix} P_1 \text{ choices} \\ (R) \end{matrix}$$

Examples illustrating how to apply this procedure

3. Solution of matrix games for the different information sets.

$$\underline{V}(A_\beta) = 1, \quad \bar{V}(A_\beta) = 2$$

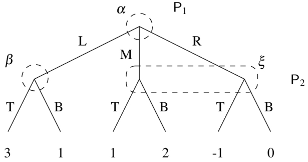
$$\underline{V}(A_\xi) = \bar{V}(A_\xi) = 0$$

The method fails

- there is no pure saddle-point equilibrium for A_β.

Examples illustrating how to apply this procedure

1.- Initial Game



2. Construction of matrix games for the different information sets

$$A_\beta = \underbrace{\begin{bmatrix} 3 & 1 \end{bmatrix}}_{\substack{P_2 \text{ choices} \\ (T,B)}} \Big\} \begin{matrix} P_1 \text{ choices} \\ (L) \end{matrix} \qquad A_\xi = \underbrace{\begin{bmatrix} 1 & 2 \\ -1 & 0 \end{bmatrix}}_{\substack{P_2 \text{ choices} \\ (T,B)}} \Bigg\} \begin{matrix} P_1 \text{ choices} \\ (M,R) \end{matrix}$$

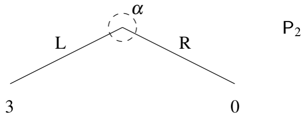
Examples illustrating how to apply this procedure

3. Solution of matrix games for the different information sets.

$$\underline{V}(A_\beta) = \bar{V}(A_\beta) = 3, \quad (i^*, j^*) = (L, T)$$
$$\underline{V}(A_\xi) = \bar{V}(A_\xi) = 0, \quad (i^*, j^*) = (R, B)$$

$$P_2 \text{ policy} = \begin{cases} T & \text{if IS} = \beta \\ B & \text{if IS} = \xi \end{cases}$$

4. Replacement of information sets by the values of the games.



$$P_1 \text{ policy} = \{R \quad \text{if IS} = \alpha$$

Recursive Computation of Eq. for Single-Stage Games

Proposition 7.2. For any zero-sum single-stage game in extensive form, if every matrix game corresponding to the information sets of the second-acting player has a pure saddle-point equilibrium, then the game in extensive form has a pure saddle-point equilibrium.

When all the information sets of a game have a single element, the matrix games corresponding to these sets necessarily have a pure saddle-point equilibrium because one of the players only has a single action.

Corollary 7.1. Every zero-sum single-stage game in extensive form for which all information sets have a single element has a pure saddle-point equilibrium.

Recursive Computation of Eq. for Single-Stage Games

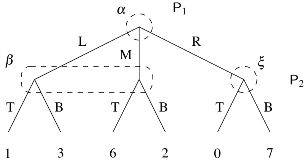
Attention! Policies constructed with this procedure are guaranteed to be saddle-point equilibria

- procedure may not bring up all possible pure saddle-point equilibria
- it will only produce dominating saddle-point equilibria.

Wen the procedure fails, there may or there may not be pure saddle-point equilibria.

Recursive Computation of Eq. for Single-Stage Games

1.- Initial Game



2. Construction of matrix games for the different information sets

$$A_\beta = \underbrace{\begin{bmatrix} 1 & 3 \\ 6 & 2 \end{bmatrix}}_{\substack{P_2 \text{ choices} \\ (T,B)}} \Big\} \begin{matrix} P_1 \text{ choices} \\ (L,M) \end{matrix} \qquad A_\xi = \underbrace{\begin{bmatrix} 0 & 7 \end{bmatrix}}_{\substack{P_2 \text{ choices} \\ (T,B)}} \Big\} \begin{matrix} P_1 \text{ choices} \\ (R) \end{matrix}$$

Recursive Computation of Eq. for Single-Stage Games

3.- Solution of matrix games for the different information sets.

$$\underline{V}(A_\beta) = 2, \qquad \bar{V}(A_\beta) = 3,$$
$$\underline{V}(A_\xi) = \bar{V}(A_\xi) = 7$$

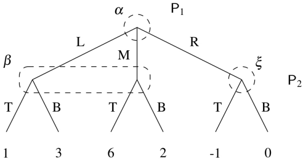
The method fails

- there is no pure saddle-point equilibrium for A_β.

This game has no pure saddle-point equilibria.

Recursive Computation of Eq. for Single-Stage Games

1.- Initial Game



2. Construction of matrix games for the different information sets

$$A_\beta = \underbrace{\begin{bmatrix} 1 & 3 \\ 6 & 2 \end{bmatrix}}_{\substack{P_2 \text{ choices} \\ (T,B)}} \left.\vphantom{\begin{bmatrix} 1 \\ 6 \end{bmatrix}}\right\} \begin{array}{c} P_1 \text{ choices} \\ (L, M) \end{array} \qquad A_\xi = \underbrace{\begin{bmatrix} -1 & 0 \end{bmatrix}}_{\substack{P_2 \text{ choices} \\ (T,B)}} \Big\} \begin{array}{c} P_1 \text{ choices} \\ (R) \end{array}$$

Recursive Computation of Eq. for Single-Stage Games

3.- Solution of matrix games for the different information sets.

$$\underline{V}(A_\beta) = 2, \qquad \bar{V}(A_\beta) = 3,$$
$$\underline{V}(A_\xi) = \bar{V}(A_\xi) = 0$$

The method fails

- there is no pure saddle-point equilibrium for A_β.

However, this game has several pure saddle-point equilibria, including

$$P_2 \text{ policy} = \begin{cases} T & \text{if } IS = \beta \\ B & \text{if } IS = \xi \end{cases} \quad P_1 \text{ policy} = \{ \; R \quad \text{if } IS = \alpha$$

P_1 never choose L or M since R strictly dominates both choices.

Feedback Games

Feedback Games

Definition 7.2 (Feedback games).
A multi-stage game in extensive form is a **feedback game (in extensive form)** if the following conditions hold:

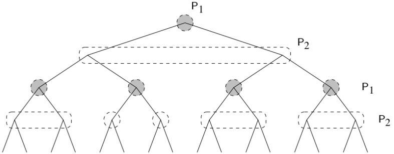
C1: no information set spans over multiple stages,

- when a player must select an action, they know the current stage of the game.

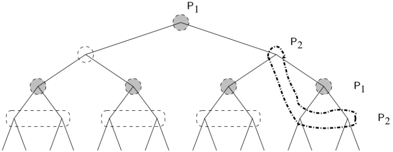
C2: the nodes that correspond to the start of each stage are the roots of sub-trees that do not share information sets with each other (including at the level of the root).

- both players have full information about what both players did in past stages of the game.

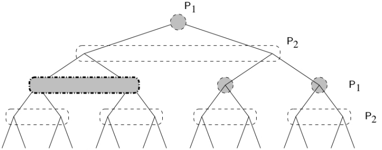
Feedback Games - Examples



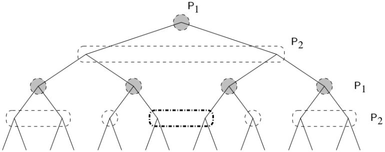
(a) Feedback game.



(b) Not a feedback game because the dash-dot information set spans over multiple stages, violating condition C7.1.



(c) Not a feedback game because at the start of the second stage P_1 does not necessarily know the action selected by P_2 in the first stage. The dash-dot information set violates condition C7.2.



(d) Not a feedback game because at the second stage, P_2 does not necessarily know the action selected by P_1 at the first stage. The dash-dot information set violates condition C7.2.

Feedback Saddle-Point for Multi-Stage Games

Feedback Saddle-Point for Multi-Stage Games

In feedback games IS do not span over multiple stages

- then, we can decompose a policy for a particular player into several sub-policies, one for each stage.

Consider a feedback game with K stages, and denote by $\mathcal{I}$ the set of all information sets for player P_i.

$\mathcal{I}$ can be partitioned into K disjoint subsets

$$\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_K, \qquad \mathcal{I} = \bigcup_{i=1}^{K} \mathcal{I}_i$$

each one containing all the Information Sets for a specific stage.

Feedback Saddle-Point for Multi-Stage Games

If we are then given a policy

$$\gamma : \mathcal{I} \to \mathcal{A}, \qquad \alpha \mapsto \gamma(\alpha)$$

that maps each IS in $\mathcal{I}$ to some action in the action space $\mathcal{A}$, we can decompose γ into K sub-policies $\gamma := \{\gamma_1, \gamma_2, \ldots, \gamma_K\}$, where each

$$\gamma_i : \mathcal{I}_i \to \mathcal{A}, \qquad \alpha \mapsto \gamma(\alpha)$$

only maps the ith stage information sets to actions in $\mathcal{A}$.

This decomposition motivates a **new** definition of saddle-point equilibria for feedback games.

Feedback Saddle-Point for Multi-Stage Games

Definition 7.3 (Feedback pure saddle-point equilibrium).

For a feedback game with K stages, a pair of policies (γ^*, σ^*)

$$\gamma^* := \{\gamma_1^*, \gamma_2^*, \dots, \gamma_K^*\}, \quad \sigma^* := \{\sigma_1^*, \sigma_2^*, \dots, \sigma_K^*\}$$

is a **feedback pure saddle-point equilibrium** if for every stage k and every policy $\gamma_1, \gamma_2, \dots, \gamma_{k-1}$ and $\sigma_1, \sigma_2, \dots, \sigma_{k-1}$ for the stages prior to k we have that

$$J\left(\left\{\gamma_{1\dots k-1}, \boxed{\gamma_k^*}, \gamma_{k+1\dots K}^*\right\}, \left\{\sigma_{1\dots k-1}, \boxed{\sigma_k^*}, \sigma_{k+1\dots K}^*\right\}\right)$$
$$\leq J\left(\left\{\gamma_{1\dots k-1}, \boxed{\gamma_k}, \gamma_{k+1\dots K}^*\right\}, \left\{\sigma_{1\dots k-1}, \boxed{\sigma_k^*}, \sigma_{k+1\dots K}^*\right\}\right) \quad \forall \gamma_k$$
$$J\left(\left\{\gamma_{1\dots k-1}, \boxed{\gamma_k^*}, \gamma_{k+1\dots K}^*\right\}, \left\{\sigma_{1\dots k-1}, \boxed{\sigma_k^*}, \sigma_{k+1\dots K}^*\right\}\right)$$
$$\geq J\left(\left\{\gamma_{1\dots k-1}, \boxed{\gamma_k^*}, \gamma_{k+1\dots K}^*\right\}, \left\{\sigma_{1\dots k-1}, \boxed{\sigma_k}, \sigma_{k+1\dots K}^*\right\}\right) \quad \forall \sigma_k$$

Universal quantifications refer to all k stage pure sub-policies.

Feedback Saddle-Point for Multi-Stage Games

$$J\left(\left\{\gamma_{1\ldots k-1}, \boxed{\gamma_k^*}, \gamma_{k+1\ldots K}^*\right\}, \left\{\sigma_{1\ldots k-1}, \boxed{\sigma_k^*}, \sigma_{k+1\ldots K}^*\right\}\right)$$
$$\leq J\left(\left\{\gamma_{1\ldots k-1}, \boxed{\gamma_k}, \gamma_{k+1\ldots K}^*\right\}, \left\{\sigma_{1\ldots k-1}, \boxed{\sigma_k^*}, \sigma_{k+1\ldots K}^*\right\}\right) \quad \forall \gamma_k$$
$$J\left(\left\{\gamma_{1\ldots k-1}, \boxed{\gamma_k^*}, \gamma_{k+1\ldots K}^*\right\}, \left\{\sigma_{1\ldots k-1}, \boxed{\sigma_k^*}, \sigma_{k+1\ldots K}^*\right\}\right)$$
$$\geq J\left(\left\{\gamma_{1\ldots k-1}, \boxed{\gamma_k^*}, \gamma_{k+1\ldots K}^*\right\}, \left\{\sigma_{1\ldots k-1}, \boxed{\sigma_k}, \sigma_{k+1\ldots K}^*\right\}\right) \quad \forall \sigma_k$$

In words:

No matter what the players do **before stage** k (rational or not) and assuming that both players will play at the equilibrium **after stage** k, the stage k sub-policies (γ_k^*, σ_k^*) must be a pure saddle-point equilibrium.

Feedback Saddle-Point for Multi-Stage Games

Lemma 7.1. For every feedback game in extensive form, a **feedback pure saddle-point equilibrium** in the sense of **Definition 7.3** is always a **pure saddle-point equilibrium** in the sense of **Definition 7.1**.

Proof of Lemma 7.1. To prove this result, show that

$$J\left(\left\{\gamma_{1\ldots k-1}, \boxed{\gamma_k^*}, \gamma^*_{k+1\ldots K}\right\}, \left\{\sigma_{1\ldots k-1}, \boxed{\sigma_k^*}, \sigma^*_{k+1\ldots K}\right\}\right)$$
$$\leq J\left(\left\{\gamma_{1\ldots k-1}, \boxed{\gamma_k}, \gamma^*_{k+1\ldots K}\right\}, \left\{\sigma_{1\ldots k-1}, \boxed{\sigma_k^*}, \sigma^*_{k+1\ldots K}\right\}\right) \quad \forall \gamma_k$$

$$J\left(\left\{\gamma_{1\ldots k-1}, \boxed{\gamma_k^*}, \gamma^*_{k+1\ldots K}\right\}, \left\{\sigma_{1\ldots k-1}, \boxed{\sigma_k^*}, \sigma^*_{k+1\ldots K}\right\}\right)$$
$$\geq J\left(\left\{\gamma_{1\ldots k-1}, \boxed{\gamma_k^*}, \gamma^*_{k+1\ldots K}\right\}, \left\{\sigma_{1\ldots k-1}, \boxed{\sigma_k}, \sigma^*_{k+1\ldots K}\right\}\right) \quad \forall \sigma_k$$

actually imply that

$$J(\gamma^*, \sigma^*) \leq J(\sigma, \gamma^*), \forall \gamma \qquad J(\gamma^*, \sigma^*) \geq J(\gamma^*, \sigma), \forall \sigma$$

Feedback Saddle-Point for Multi-Stage Games

Previous equations can be re-written in terms of sub-policies as:

$$J\Big(\underbrace{\{\gamma^*_{1\ldots K}\}}_{\gamma^*},\underbrace{\{\sigma^*_{1\ldots K}\}}_{\sigma^*}\Big) \leq J\Big(\underbrace{\{\gamma_{1\ldots K}\}}_{\gamma},\underbrace{\{\sigma^*_{1\ldots K}\}}_{\sigma^*}\Big), \quad \forall \gamma_1, \gamma_2, \ldots, \gamma_K$$

$$J\Big(\underbrace{\{\gamma^*_{1\ldots K}\}}_{\gamma^*},\underbrace{\{\sigma^*_{1\ldots K}\}}_{\sigma^*}\Big) \leq J\Big(\underbrace{\{\gamma^*_{1\ldots K}\}}_{\gamma^*},\underbrace{\{\sigma_{1\ldots K}\}}_{\sigma}\Big), \quad \forall \sigma_1, \sigma_2, \ldots, \sigma_K$$

To accomplish this, use the first eq., for the first stage $k = 1$

$$J\Big(\{\boxed{\gamma^*_1}, \gamma^*_{2\ldots K}\}, \underbrace{\{\boxed{\sigma^*_1}, \sigma^*_{2\ldots K}\},}_{\sigma^*}\Big) \leq J\Big(\{\boxed{\gamma_1}, \gamma^*_{2\ldots K}\}, \underbrace{\{\boxed{\sigma^*_1}, \sigma^*_{2\ldots K}\},}_{\sigma^*}\Big), \quad \forall \gamma_1$$

and now for the stage $k = 2$, with $\sigma_1 := \sigma^*_1$ in the first stage

$$J\Big(\{\gamma_1, \boxed{\gamma^*_2}, \gamma^*_{3\ldots K}\}, \underbrace{\{\sigma^*_1, \boxed{\sigma^*_2}, \sigma^*_{3\ldots K}\}}_{\sigma^*}\Big) \leq J\Big(\{\gamma_1, \boxed{\gamma_2}, \gamma^*_{3\ldots K}\}, \underbrace{\{\sigma^*_1, \boxed{\sigma^*_2}, \sigma^*_{3\ldots K}\}}_{\sigma^*}\Big), \quad \forall \gamma$$

Feedback Saddle-Point for Multi-Stage Games

Combining these two inequalities we obtain

$$J\big(\{\gamma^*_{1\ldots K}\}, \underbrace{\{\sigma^*_{1\ldots K}\}}_{\sigma^*}\big) \leq J\big(\{\gamma_1, \gamma_2, \gamma^*_{3\ldots K}\}, \underbrace{\{\sigma^*_{1\ldots K}\}}_{\sigma^*} x\big), \quad \forall \gamma_1, \gamma_2$$

As we combine this with the first eq. in **Lemma 7.1**, for each subsequent stage $k \in \{3, 4, \ldots, K\}$ and with $\sigma_i = \sigma_i^*$, $\forall i \in \{1, 2, \ldots, k-1\}$ for the previous stages, we eventually obtain

$$J\big(\{\gamma^*_{1\ldots K}\}, \underbrace{\{\sigma^*_{1\ldots K}\}}_{\sigma^*}\big) \leq J\big(\{\gamma_{1\ldots K}\}, \underbrace{\{\sigma^*_{1\ldots K}\}}_{\sigma^*} x\big), \quad \forall \gamma_1, \gamma_2, \ldots, \gamma_K$$

which proves that $J(\gamma^*, \sigma^*) \leq J(\gamma, \sigma^*)$, $\forall \gamma$.

The other saddle-point inequality presented, can be proved similarly using the second eq. in **Lemma 7.1**.

Feedback Saddle-Point for Multi-Stage Games

Note 8 (Feedback pure saddle-point equilibria).

Why introduce a new notion of saddle-point equilibria when we already had a very reasonable one?

There are several reasons:

1. Finding feedback saddle-point equilibria is easier than finding saddle-point equilibria that are not feedback.
2. Feedback saddle-point equilibria are **better**:
   - they provide optimal security levels, even if the player did not play rationally in the past.
   - these security levels may not be as good as the ones that could have been obtained if the player had been playing rationally since the beginning of the game.

Recursive Computation of Equilibria for Multi-Stage Games

Feedback Saddle-Point for Multi-Stage Games

Procedure: (assume P_1 is first-acting, game has K stages)

Step 1. Construct a matrix game corresponding to each IS of P_2 at the Kth stage. Each matrix game will have one action of P_1 for each edge entering the IS and one action of P_2 for each edge leaving the IS.

Step 2. Compute pure saddle-point equilibria for each of the matrix games constructed.

- If any of these matrix games does not have a pure saddle-point equilibrium, this method fails.
- Otherwise, P_2's pure security/saddle-point sub-policy σ_K^* for the Kth stage should map to each IS the action corresponding to P_2's security policy/saddle-point in the corresponding matrix game.

Feedback Saddle-Point for Multi-Stage Games

Step 3. Replace each IS for P_2 at the Kth stage by the value of the corresponding pure saddle-point equilibrium.

- link leading to this value should be labeled with P_1's pure security/saddle-point in the corresponding matrix game.

Step 4. The pure security/saddle-point sub-policy γ_K^* for P_1 at Kth stage maps to each IS of P_1, the action in the link corresp. to the most favorable pure saddle-point value for P_1.

The policies (γ_K^*, σ_K^*) have the property that, for every policy $\gamma_1, \gamma_2, \ldots, \gamma_{K-1}$ and $\sigma_1, \sigma_2, \ldots, \sigma_{K-1}$ for the stages prior to K,

$$J\left((\{\gamma_{1\ldots K-1}, \boxed{\gamma_K^*}\}, \{\sigma_{1\ldots K-1}, \boxed{\sigma_K^*}\})\right) \leq J\left((\{\gamma_{1\ldots K-1}, \boxed{\gamma_K}\}, \{\sigma_{1\ldots K-1}, \boxed{\sigma_K^*}\})\right),$$

$$J\left((\{\gamma_{1\ldots K-1}, \boxed{\gamma_K^*}\}, \{\sigma_{1\ldots K-1}, \boxed{\sigma_K^*}\})\right) \geq J\left((\{\gamma_{1\ldots K-1}, \boxed{\gamma_K^*}\}, \{\sigma_{1\ldots K-1}, \boxed{\sigma_K}\})\right),$$

as required for a feedback pure saddle-point equilibrium

Feedback Saddle-Point for Multi-Stage Games

Step 5. Replace each IS for P_1 at the Kth stage by the value corresponding to the link selected by P_1.

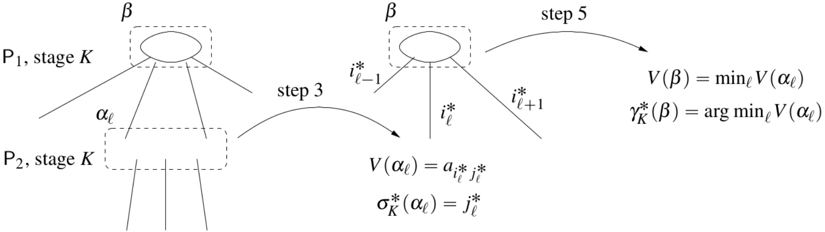
At this point we have a game with $K-1$ stages, and we compute the sub-policies $(\gamma^*_{K-1}, \sigma^*_{K-1})$ using the exact same procedure described above.

This algorithm is iterated until all sub-policies have been obtained.

At every stage we guarantee by construction that the conditions in the equations, for a feedback pure saddle-point equilibrium hold.

Feedback Saddle-Point for Multi-Stage Games

Illustration of steps 3-5 in the recursive computation of equilibria for multi-stage games.



α: information state corresponding to P_2 at the Kth stage
$V(\alpha)$: value of the matrix game associated with that information state
(i^*, j^*): the corresponding saddle-point equilibrium point.

Feedback Saddle-Point for Multi-Stage Games

Proposition 7.3. For any zero-sum game in extensive form with a finite number of stages, if every matrix game constructed using the above procedure has a pure saddle-point equilibrium, then the overall game in extensive form has a feedback pure saddle-point equilibrium.

When all the IS of a game have a single element, the matrix games corresponding to these sets necessarily have a saddle-point equilibrium because one of the players only has a single action.

This observation leads to the following consequence:

Corollary 7.2. Every zero-sum game in extensive form with a finite number of stages for which all information sets have a single element has a feedback pure saddle-point equilibrium.

Practice Exercise

Practice Exercise

7.1 (Number of games in extensive form).

How many different extensive games are possible for a given $m \times n$ matrix of outcomes?

Solution to Exercise 7.1.

Assuming P_1 plays first, the number of distinct games is given by the **Bell number** b_m: the number of partitions of the set $\{1, 2, \ldots, m\}$. b_m is computed recursively by

$$b_{k+1} = \sum_{i=1}^{k} \binom{k}{i} \qquad \text{(cf. Table below).}$$

m	0	1	2	3	4	5	6	7	8	9	$\cdots$	15
b_m	1	1	2	5	15	52	203	877	4149	21147	$\cdots$	$\approx 1.4 \times 10^9$

Practice Exercise

On the other hand, assuming that player P_2 plays first, the number of distinct games is given by the **Bell number** b_n. Consequently, the total number of distinct games is equal to

$$b_m - b_n - 1$$

subtraction by 1 is to avoid double counting the simultaneous play, which otherwise would be counted twice.
For example for the 2×3 matrix

$$A = \underbrace{\begin{bmatrix} 1 & 3 & 0 \\ 6 & 2 & 7 \end{bmatrix}}_{\substack{P_2 \text{ choices} \\ \text{Left, Middle, Right}}} \left.\vphantom{\begin{bmatrix} 1 \\ 6 \end{bmatrix}}\right\} \begin{array}{c} P_1 \text{ choices} \\ \text{Top, Bottom} \end{array}$$

we have $b_2 + b_3 - 1 = 2 + 5 - 1 = 6$ games.

End of Lecture

07 - Games in Extensive Form

Questions?