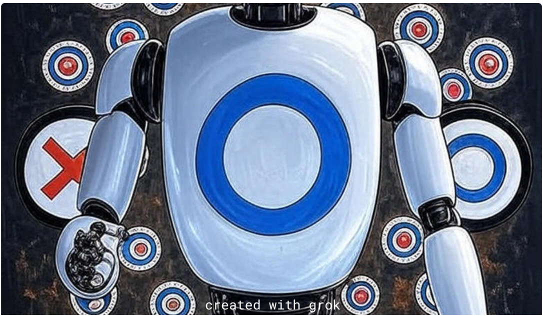
Game Theory and Control

Silvan Stadelmann - 8. Dezember 2025 - v0.0.1

github.com/silvasta/summary-gtc



Contents

1 Static games

1.1 Basic Definitions

Game theory is the study of mathematical models of conflict and cooperation between rational decision-makers.

Several things are needed to characterize a game:

- The **players** are the agents that make decisions

- The **actions** available to each player at each decision point

- The **information structure** specifies what each player knows before making each decision, in particular with respect to other players' decisions

- The **outcome** for each player, which depends on all players' decisions

1.1.1 Matrix representation

1.1.2 Nash Equilibrium

1.1.3 Dominated actions

1.1.4 Reduced game

1.1.5 Security levels and policies

1.2 Multiple Nash Equilibria

1.2.1 Admissible Nash Equilibria

Poset

Hasse diagram

Minimal element

1.3 Mixed strategies

Careful!Refer to deterministic strategies as pure strategies!

1.3.1 Security levels

Mixed security level

Mixed security strategy

Computational complexity

1.3.2 Mixed Nash Equilibrium

1.4 Nash Theorem

2 Zero-sum games

- Zero-sum games model a large number of practical applications

- Nash equilibria in zero-sum games have many useful properties

- Nash equilibria in zero-sum games are much easier to compute

2.1 Two-Person Zero-sum Games

Games in which the two players have opposite payoffs:

$$J_1(\gamma, \sigma) = -J_2(\gamma, \sigma)$$

Static game with $B = -A$ (we only indicate one matrix, A)

2.1.1 Payoff Matrix

- Row player loses aij

- Column player gains aij

- Row player minimizes outcome V

- Column player maximizes outcome V

2.1.2 Rock, Paper, Scissors

Consider only one round of the game.

$$A = \begin{array}{cc} \begin{array}{ccc} \text{Rock} & \text{Paper} & \text{Scissors} \\ \end{array} & \\ \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix} & \begin{array}{c} \text{Rock} \\ \text{Paper} \\ \text{Scissors} \end{array} \end{array}$$

2.2 Security levels and policies

- Security level Player 1,2

- Security policy Player 1,2

2.2.1 Min-Max Property

Security levels in **zero-sum games** have a fundamental property that general static games don't have.

For every finite matrix A, the following properties hold:

(i) Security levels are well defined and unique

(ii) Both players have security policies (not necessarily unique)

(iii) The security levels always satisfy:

2.3 Nash equilibrium in zero-sum games

Intepretation: As in static games: no regret, stable strategy when iterated, etc.

Also known as **saddle-point equilibrium**

$$a_{i^\star j} \leq a_{i^\star j^\star} \leq a_{ij^\star} \quad \forall i \in 1,...,n, \forall j \in 1,...,m$$

Saddle-point value $V^\star := a_{i^\star j^\star}$

2.4 Saddle-point and security levels

Not all zero-sum games have a saddle point (f.e. Rock-Paper-Scissors). We can exactly characterize the zero-sum games that have a saddle point.

Theorem 1 (Saddle-point and security levels)**.** A zero-sum game defined by A has a saddle-point equilibrium **if and only if**

$$\underline{\mathrm{V}} = \bar{V} \quad (= V^\star \text{ saddle-point value})$$

Important consequences follow (only for zero-sum games!)

All saddle-point equilibria (Nash equilibria) of a zero-sum game have the same value $V^\star$, which we denote as the value of the game.

2.5 Mixed strategies

2.5.1 Computing the mixed security level via LP

pseudo, matlab code:

m = 5; n = 10; A = rand (m, n) ; y = sdpvar (m, 1) ; t = sdpvar (1 , 1) ; obj = t ; c o n s t r a i n t s = [A ' ✗ y <= ones (n , 1) ✗ t , sum (y) == 1 , y >= 0] ; optimize (constraints , obj) ; S e c u r i t y L e v e l = double (t) ; S e c u r i t y P o l i c y = double (y) ;

2.5.2 Min-Max Property

2.5.3 Nash Equilibrum

Theorem 2 (Mixed Nash equilibrium for zero-sum games)**.** Policy $(y^\star, z^\star)$ is called mixed-strategy saddle-point equilibrium (or Nash equilibrium) if

$$y^{\star T} A z^\star \leq y^T A z^\star \quad \forall y \quad \text{(minimizer)}$$

$$y^{\star T} A z^\star \geq y^{\star T} A z \quad \forall z \quad \text{(maximizer)}$$

$y^{\star T} A z^\star$ is called **saddle point value**

3 Auctions

- N Players: bidders

- Action: bid x_i≥0

Outcome

- $w(x)$ the winner of the auction

- $p(x)$ the price that the winner has to pay

- t_i the true value of the item for agent i

The outcome (cost) for agent i is:

$$J_i = \begin{cases} p(x) - t_i & \text{if} i = w(x) \\ 0 & \text{otherwise} \end{cases}$$

3.1 First-price auctions

In first-price auctions, the best-response bid depends on the bids of the other agents and your own true value.

- Underbidding $x_i < t_i$

- Truthful bidding $x_i = t_i$

- Overbidding $x_i > t_i$

Proposition 1 (first-price auctions)**.** .

- any **overbidding** strategy is dominated by **truthful bidding**
- **truthful bidding** is not a dominant strategy

3.2 The problem of private information

Until now, we always assumed that the cost functions of the agents is known to the other agents. Auctions are an application of game theory in which this is not true, and we encode all the **private information** in the true value t_i.

Lack of information leads to inefficiency!

3.3 Second-price auctions

Winner selection $w(x) = \text{argmax}_i\, x_i$

(the bidder with the highest bid wins)

Payment rule $p(x) = \max_{i \neq w(x)} x_i$

(the winner pays the second-largest bid)

3.3.1 Dominant Strategy

In second-price auctions, a Nash equilibrium exists and **can be computed by each agent based on their own private information.**

Truthful bidding is a weakly dominant strategy in a second-price auction

Intuition: your bid determines whether you win, not how much you pay.

- Incentive compatibility

3.3.2 Properties of second-price auction

Social efficiency

Does the auctioneer achieve the highest return?

No (cost of eliciting truthful bidding...)

Incentive compatibility and **social efficiency** often go together (the true value needs to be disclosed in order to be used for efficient allocation).

3.4 Generalized auctions

3.4.1 Bids

Each **bid** is represented by a pair $x_j = (b_j, m_j)$

- b_j is the bidded amount

- m_j describes the object of the bid

(Can be extended to allow multiple bids)

Fungible goods $m_j \in \mathbb{R}_{>0}$ parts of a total quantity M

Non-fungible goods $m_j \in 2^{\mathcal{M}}$ with finite set of items $\mathcal{M}$

3.4.2 Choice function

Choice function w maps bids x into N-dimensional binary vector

$$w_j(x) = \begin{cases} 1 & \text{if bid } j \text{ is accepted} \\ 0 & \text{otherwise} \end{cases}$$

- Choice constraints

3.4.3 Payment function

Payment function p maps bids x into N-dimensional vector where $p_j(x)$ is the payment requested from the player that placed the bid j

3.5 VCG auctions

- VCG choice function

- VCG payment function

In order to compute each payment p_j, we need to evaluate the choice function twice: with and without the bid j

3.5.1 Social utility

The social utility is the aggregate utility of all players and the auctioneer

$$U(t, w) = \sum_i t_i w_i$$

The social utility - depends on the true value of the goods according to the player that receives it

- does not depend on the entity of the payments

If players bid truthfully $b_j = t_j$ then VCG choice function achieves maximal social utility $U^\star$

Interpretation of the VCG payment if agents bid truthfully:

3.5.2 Non-negative utility

When an agent bids truthfully, his utility is non-negative.

3.5.3 Dominant bidding strategy

Truthful bidding is a weakly dominant strategy in a VCG auction.

3.6 Open problem of auction design

We saw how to design an auction which guarantees

- incentive compatibility / truthful bidding

- optimal social efficiency

- non-negative payments

Unfortunately, it comes with drawbacks. For example

- it yields low returns

- it can be manipulated by colluding agents

- it is computationally challenging to solve.

4 Potential games

4.1 N-player games

N-player non-zero-sum games

- Player i can choose one among m_i pure actions

$$\Gamma_i = \{\gamma_i^{(1)}, \gamma_i^{(2)}, ... \gamma_i^{(m_i)}\}$$

- The outcome of the game for Player i is given by

$$J_i = (\gamma_1, \gamma_2, ... \gamma_N) = J_i(\gamma_i, \gamma_{-i})$$

Definition 1 (Pure Nash equilibrium in N-player games)**.** A pure strategy profile $\gamma^\star = \{\gamma_i^\star, \gamma_2^\star, ... \gamma_N^\star\}$ is a pure Nash equilibrium if for every player i

$$J_i(\gamma_i^\star, \gamma_{-i}^\star) \leq J_i(\gamma_i', \gamma_{-i}^\star) \quad \gamma_i' \in \Gamma_i$$

- Randomized play $\rightarrow$mixed strategies, mixed Nash equilibria

- Multiple Nash equilibria are possible (non interchangeable, different payoffs)

Definition 2 (Pure Best Response in N-player games)**.** The pure best response of player i is the set $R_i(\gamma_{\text{-}i}) \subseteq \Gamma_i$ such that $\gamma_i \in R_i(\gamma_{\text{-}i})$ if and only if

$$J_i(\gamma_i, \gamma_{-i}) \leq J_i(\gamma_i', \gamma_{-i}) \quad \forall \gamma_i' \in \Gamma_i$$

Equivalent: $R_i(\gamma_{-i}) :=$ argmin$_{\gamma_i \in \Gamma_i} J_i(\gamma_i, \gamma_{-i})$

- $R_i(\gamma_{-i})$ is a set, and it is not necessarily a singleton

- $R_i(\gamma_{-i})$ is never empty.

- $R_i(\gamma_{-i})$ is a function of the strategies of other players.

Proposition 2. A pure strategy profile $\gamma^\star = \{\gamma_i^\star, \gamma_2^\star, ...\gamma_N^\star\}$ is a pure Nash equilibrium if and only if $\gamma_i^\star \in R_i(\gamma_{-i}^\star)$ for every player i.

4.1.1 Best-response dynamics

Consider an initial pure strategy profile $\gamma^0 = \{\gamma_1^0, \gamma_2^0, ...\gamma_N^0\}$

Step $k = 0, ...$:

1 - If γ^k is a pure Nash equilibrium stop

2 - Else there exists a player i for which $\gamma_i^k \notin R(\gamma_{-i})$

3 - Update:$\gamma^{k+1} := (R(\gamma_{-i}), \gamma_{-i})$.

4 - $k = k + 1$, goto step 1.

Clearly, it does not converge if a pure Nash equilibrium does not exist.

Conjecture: It always converges to a pure Nash equilibrium, if that exists.

In this summary:

A class of N-player non-zero sum games for which

- a pure Nash equilibrium is guaranteed to exist

- best-response dynamics converge

- pure Nash equilibria are easy to find

4.2 Potential games

Definition 3 (Potential function)**.** A function $P : \gamma_1 \times \gamma_2 \times \cdots \times \gamma_N \to \mathbb{R}$ is a **potential function** if for every player i and every γ_{-i}

$$J_i(\gamma_i', \gamma_{-i}) - J_i(\gamma_i'', \gamma_{-i}) = P(\gamma_i', \gamma_{-i}) - P(\gamma_i'', \gamma_{-i})$$

for every $\gamma_i', \gamma_i'' \in \Gamma_i$

A game is a **potential game** if it admits a potential function.

Note:

- The potential function P is the same for all players

- The potential function assigns a value to each joint strategy profile

- When player i chooses a best response, the potential decreases.

Proposition 3. Finite games with a potential function have a pure Nash equilibrium. Furthermore, best response dynamics converge.

- Provides a **computation method** and an intuition for **repeated games**

- These iterations converge to a Nash equilbrium that depends on the **initial conditions**

- It does not converge only to admissible Nash equilibria

Proposition 4. In potential games, Nash equilibria correspond to **directionally maxima** (or minima, if players are minimizers) of the potential.

4.2.1 Paths

- A path in Γ is a sequence ...

- closed path

- simple path

4.3 Congestion games

Theorem 3 (Potential function for Congestion games)**.** The following is a potential function for congestion games.

$$P(\gamma) = \sum_{j=1}^{M} \sum_{k=1}^{\ell_j(\gamma)} f_j(k)$$

Consequently, congestion games admit a pure Nash equilibrium.

4.4 Social welfare

Definition 4 (Welfare function). In a N-person game, let $\gamma_i \in \Gamma_i$ be the strategy played by agent i.

Let $\gamma \in \Gamma := \Gamma_1 \times \Gamma_2 \times \cdots \times \Gamma_N$ be the system-wide strategy.

A **welfare cost** $W : \Gamma \to \mathbb{R}$ is a measure of efficiency of each strategy for the social cost of the population of agents.

Let the **individual cost** be $J_i(\gamma)$ that player i wants to minimize.

4.4.1 Price of Anarchy

5 Convex games

5.1 Games with infinite actions

Examples:

- money (e.g., auctions)
- physical control inputs (force, velocity, ...)
- waiting time (e.g., Start-Stop)
- coverage path of a surveillance camera

5.1.1 Problems with infinite actions

Theory Most of the properties and results that we saw don't hold anymore!

- every time there is an argmin or argmax
- Nash Theorem

Algorithms The few algorithms that we have seen are also not suited for infinite actions

- algorithms *by inspection*
- linear programming

5.1.2 Example: Cournot competition

Problem Setup - Cournot competition

Consider two producers competing in a market.
Each player (producer) $i = 1, 2$ decides on the **quantity** to produce denoted by $x_i \geq 0$, and has a production marginal cost of $c > 0$.
The market price $p : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a linearly decreasing function of the total production $x_1 + x_2$

$$p(x_1, x_2) = a\text{-}b(x_1 + x_2)$$

1. Write each producer's losses as a function of the quantities produced.
2. Derive the Nash equilibrium.
3. What is the Nash equilibrium if producers' decisions are restricted to $[0, k]$?

5.1.3 Example: Betrand competition

Problem Setup - Betrand competition

Consider two producers competing in a market.
Each producer decides on the price $x_i \geq 0$ of their product, and has a production marginal cost of $c > 0$.

The total demand is 1 unit and the consumers choose to buy from the producer with the lowest price (if both firms declare the same price, then half of the demand chooses firm 1 and the other half chooses firm 2).

1. Write each producer's losses as a function of the price they charge.

2. Derive the Nash equilibrium.
3. What is the Nash equilibrium if each producer has the capacity to serve maximum 2/3 of the unit demand?

5.1.4 Definitions and background

Convexity

- Convex set - Convex function

Differentiable functions

- Convex function - Characterization: Hessian

5.2 Convex games

N-player game with continuous action spaces

- N player game
- Player i's action $x_i \in K_i \subset \mathbb{R}^n$
- K_i is non-empty, closed and convex
- def: $K = K_1 \times K_2 \times \cdots \times K_N \mathbb{R}^{nN}$
- Player i's outcome $J_i : K \to \mathbb{R}$
- Compact notation: $J_i(x_i, x_{\text{-}i})$

Definition 5 (Nash Equilibrium - Convex Games)**.** mostly the same as usual

Theorem 4 (Existance of Pure NE - Convex Games)**.** Consider an N-player game with continuous action spaces K_i. Suppose

- action spaces $K_i \subset \mathbb{R}^n$ are **compact and convex**
- J_i are **continuous** in $x \in K$
- J_i are **convex** in x_i for fixed $x_{\text{-}i}$.

Then a pure Nash equilibrium $x^\star$ exists.

Games that satisfy those conditions are called **convex games**.

Theorem 5 (Maximum theorem)**.**

Upper hemi-continuity

Cournot and Bertrand models

5.3 Variational inequalities

Definition 6 (Variational inequalitiy)**.**

VI and convex optimization

Definition 7 (First-order optimaliy conditions)**.**

5.3.1 VI and Nash Equilibria

Characterization of Nash Equilibria of convex games

Definition 8 (Nash Equilibria and Variational Inequality)**.**

Two important advantages coming from the connection between **variational inequalities** and **Nash equilibria**.

- Borrow **theoretical results** from VI (e.g. uniqueness)
- Borrow **numerical methods** to solve VI / find NE.

Monotone maps

Definition 9 (Monotonicity)**.**

How to check monotonicity of a map F?

5.3.2 Uniqueness of Nash Equilibria

Definition 10 (SOL(K, F) is a singleton)**.**

Corollary 1.

5.4 Computing the Nash Equilibrium of a convex game

Learning/computing the Nash Equilibrium

Definition 11 (Best-response iteration)**.**

Iterative NE-seeking algorithm

Definition 12 (Iterative update)**.**

5.4.1 Projected game map

Definition 13 (Conjecture)**.**

Interpretation

Analysis

Definition 14 (Assumptions).

5.4.2 Analysis of convergence

Equilibrium

Proposition 5.

Contractive maps

Theorem 6 (Banach Fixed Point).

Contractiveness of the projected-game-map iteration

Projection is non-expansive

Theorem 7 (Projection non-expansive).

Convergence result

Theorem 8 (Convergence of projection).

Example TCP congestion

6 Stackelberg games

6.1 Definition of Stackelberg games

- Γ: pure-strategy space of the **leader** (Player 1)
- Σ: pure-strategy space of the **follower** (Player 2)

Rational reaction set

Stackelberg equilibrium

A pair of strategies $\tilde{y} \in \mathcal{Y}$ and $\tilde{\sigma}(y) : \mathcal{Y} \to \Sigma$ is a Stackelberg Equilibrium if

- Player 1 plays the best response to $\tilde{\sigma}(y)$:
- Player 2 plays the best response to y:

Applications

6.2 Stackelberg zero-sum games

Stackelberg Equilibrium in zero-sum games

6.2.1 Security strategies

Mixed security strategy

Pure security strategy

6.2.2 Stackelberg vs Nash

In zero-sum games, mixed Stackelberg equilibria and Nash equilibria coincide.

6.3 Stackelberg non-zero-sum games

Highest leader cost

Theorem 9 (Upper bound on Stackelberg cost).
Theorem 10 (Nash vs Highest Leader Cost).

Lowest leader cost

Theorem 11 (Nash vs Lowest Leader Cost).

Generic Stackelberg games

6.3.1 Computation of the Stackelberg equilibrium

Divide-and-conquer algorithm

Linear programming

In non-zero-sum games with mixed strategies, computing Stackelberg Equilibria is much easier than computing Nash Equilibria!

6.4 Security games

Randomized defender stategy in security games

Definition 15 (Coverage vector)**.**

Randomized attacker stategy in security games

6.4.1 Stackelberg solution of a security game

6.4.2 Nash equilibria of security games

Auxiliary zero-sum security game

Interpretation

Stackelberg equilibria are Nash equilibria

7 Repeated games

7.1 Tragedy of the commons

N players (companies, countries) have access to a single resource

Players can

- **cooperate** and limit their consumption

- **exploit** the resource, leading to overuse and depletion

2 Players (maximizers)

$$\begin{array}{c c} & \begin{array}{cc} \text{cooperate} & \text{exploit} \end{array} \\ \begin{array}{c} \text{cooperate} \\ \text{exploit} \end{array} & \begin{bmatrix} (a,a) & (0,b) \\ (b,0) & (c,c) \end{bmatrix} \end{array}$$

N Players

$$P_i = \begin{cases} a\frac{N_C-1}{N-1}\text{(cooperate)} \\ (b-c)\frac{N_C}{N-1} + c\text{(exploit)} \end{cases}$$

Best response:

Exploit

- is a dominant strategy, and therefore the pure strategy

- is the only Nash Equilibrium

7.1.1 Application domains

- Human population growth leading to overpopulation
- Atmospheric pollution (ozone depletion, global warming, ...)
- Water use and overirrigation
- Logging of old forests
- Fossil fuel use and consequential global warming
- Ocean overfishing
- Antibiotic use and antibiotic resistance
- Vaccinations and herd immunity
- Wi-Fi channel use and transmission power
- Hoarding of items such as toilet paper during a perceived threat

7.2 Tragedy of the commons: multi-stage version

- **Supergame** $G^{(T)}$, same game G repeatet T times.

- **Actions** $u^t \in$ {cooperate,exploit}^{N} of N players at stage t

- **History of the game** $\mathcal{H}^t$ actions of N players before stage t

$\mathcal{H}^1 = ()$ empty vector, $\mathcal{H}^T = (u^1, \ldots, u^{T-1})$

Assumption: All agents have perfect recall of the past.

7.3 Strategies in a repeated game

- **Pure strategy** $\gamma_i^{(T)}$ of Player i is **sequence of functions**

$$\gamma_i^{(T)} : \mathcal{H}^t \to \mathcal{U} := \{\text{cooperate}, \text{exploit}\}$$

- **Payoff** of repeated game:

$$P_i^{(T)}(\gamma_i^{(T)}, \ldots, \gamma_N^{(T)}) = \sum_{t=1}^{T} P_i(u^t)$$

- P_i payoff function of the game played at every stage
- $u^t = \gamma^t(\mathcal{H}^t)$

NE

Player i cannot improve by unilaterally changing his strategy, while other agents maintain the same **strategy** (not **actions**!)

Nash equilibria of repeated games

Theorem 12. Consider a game G, and its repetition T times G(T) .
- If
- If

7.4 Infinitely repeated games

Infinite repetition $G^(\infty)$ of the same game G.

Discounted payoff

$$P_i^{(\infty)}(\gamma_i^{(\infty)}) = (1-\delta)\sum_{t=1}^{\infty} \delta^{t-1} P_i(u^t)$$

with **discoun factor** $0 < \delta < 1$

- **evil** strategy $\gamma_i^t = \text{exploit}$

Proposition

The strategy evil $\forall i$ is a Nash Equilibrium.

- **trigger** strategy

$$\gamma_i^t = \begin{cases} \text{cooperate} & \text{if} u_j^\tau = \text{cooperate} \forall J \neq i, \forall \tau < t \\ \text{exploit} & \text{else} \end{cases}$$

Proposition

NE if...

Positive message: Cooperation can be achieved via repetition!

Cooperation via repetition

Consider a static game G and its infinite repetition $G^(\infty)$
Consider a strategy $\hat{\gamma}$ for G that achieves individual payoffs

$$\hat{P}_i = P_i(\hat{\gamma}) > \bar{P}_i := \min_{\gamma_{-i}} \max_{\gamma_i} P_i(\gamma_{-i}, \gamma_i)$$

Then there exists $\delta_0 \in (0,1)$, such that for all $\delta \in (\delta_0, 1)$ the game $G^(\infty)$ has a Nash Equilibrium $\gamma^{(\infty)}$ with $P_i^{(\infty)} = \hat{P}_i$

Infinitely repeated games tend to have **many Nash equilibria**

8 Multistage games

- Example: Tic-tac-toe

"Advanced" games

Features:

- multiple stages
- different order of choice
- variable number of stages
- partial information (dependent on actions)
- memory-constrained players

The (repeated) **matrix form** is not the most effective representation.

8.1 Tree representation

- The game evolves from the root to the leaves
  - Let us consider zero-sum games for now
- Each level of the tree corresponds to a player's turn
- Links correspond to actions
- Each leaf is associated to a final outcome
- Nodes of each player are divided into information sets
  - each node in the same information set has the same branches

Actions and strategies

- Simultaneous play

- Sequential play

From extensive form to matrix form

Games in extensive form can be reformulated in matrix form!

Credible threats

8.1.1 Subgame perfect equilibria

A strategy is a subgame perfect equilibrium if it represents a NE of every subgame of the original game.

- The notion of subgame is not always well defined.

- One special case: games with perfect information.

Example: Chess

8.1.2 Backward induction

Features

- More efficient than exploring the matrix form

- Returns a strategy that is a subgame perfect equilibrium

Can we always apply backward induction?

We can for games with perfect information.

Feedback games

A multi-stage game in extensive form is a feedback game if each "Player 1" node is the root of a separated sub-game.

Backward induction in feedback games

Starting from the leaves, identify subgames for which we can determine the pure NE strategy...

... either because players have full information...

... or because they play a simultaneous game with pure NE.

- Solve the game from the leaves towards the root
- Move up stage-by-stage (not level-by-level)
- Record the pure equilibrium strategy for each information set
- If the algorithm converges to the root, then we have a pure NE.
- No guarantees of convergence, even when a pure NE exists.

Towards randomized strategies

8.2 Mixed strategies

Nash's Existence theorem

8.3 Behavioral strategies

Mixed vs. behavioral

Kuhn's theorem

8.4 Single stage game

How to search for a Nash equilibrium behavioral strategy?

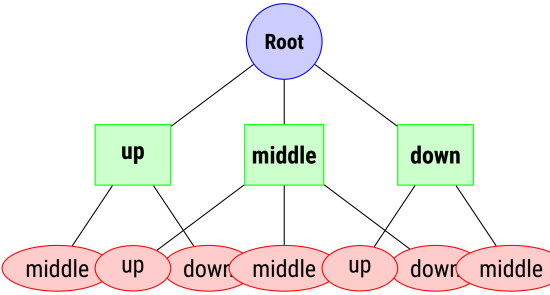
Step 1-3

Backward induction

9 Dynamic games

9.1 Escape game

Zeichnung Baum



Multi-stage game with

- finite actions

- large number of stages (20)

Order of: $1 + 9 + 81 + ... + 320$ linear programs

9.1.1 Taming the complexity of dynamic games

How can you solve this game for 20 stages?

Key idea: In every "position", the probability of escaping (and the optimal strategy) do not depend on the past decisions.

We can perform backward induction on the "positions" (states!) rather than on the game tree.

3 × 20 linear programs instead of 5 × 109

With more actions available at each stage, the speed up is even larger!

We will adapt this idea to dynamic games with infinite action spaces.

9.1.2 From tree model to loop model

A1 state that evolves

A2 outcome

9.1.3 Loop model

9.2 Non-zero sum dynamic games

A three-truck platoon

Loop models are equivalent to tree models. Therefore all results and definitions (NE, pure strategies, mixed strategies) extend here, and have a *control* interpretation.

Subgame-perfect Nash Equilibrium

A strategy is a **subgame perfect** equilibrium if it represents a NE of every subgame of the original game.

Subgame-perfect NE strategies (that is, feedback laws) are what in control we define as **optimal feedback laws**.

9.3 Backward induction

Example pyramide

9.3.1 One-player case

In the case of a one-player game, this reduces to Dynamic Programming.

- Bellman

9.3.2 Two players: Linear-quadratic case

Dynamic game

There exists a **state** that evolves at each stage

$$x$$

The outcome of the game can be expressed as

$$\sum$$

9.3.3 Two player LQR

EXAMPLES

$$x_{k+1} = Ax_k + B_1 u_k + B_2 v_k$$

Non-zero sum game setup

$$J_1$$

$$J_2$$

DERIVATION

$$\hat{u}_k(v_k) =$$
$$\hat{v}_k(u_k) =$$

NE feedback

H_1, H_2 from stacking expressions

Finally, $J_1^\star = ... = x_k^T P_{1,k} x_k$

10 Stochastic games