Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Game Theoretical Motion Planning

Tutorial ICRA 2021

Alessandro Zanardi[1], *Saverio Bolognani*[2], *Andrea Censi*[1] *and Emilio Frazzoli*[1]

[1]A. Zanardi, A. Censi and E. Frazzoli are with the Institute for Dynamic Systems and Control, ETH Zurich, 8092 Zurich, Switzerland.
[2]S. Bolognani is with the Automatic Control Laboratory, ETH Zurich, 8092 Zurich, Switzerland.
This work was supported by the Swiss National Science Foundation under NCCR Automation.

You are reading the Monograph associated to the *"Game-theoretical motion planning"* tutorial session held at ICRA 2021.

Visit the website to get the last updated version:

idsc.ethz.ch/research-frazzoli/workshops/game-theoretical-planning-icra21

This version has been compiled on August 9, 2021.

For typos, errors or suggestions feel free to contact azanardi@ethz.ch.

Contents

Contents iii

1 Preface 1
1.1 To Whom This May Concern 1
1.2 Monograph organization 2

2 Game Theory for Embodied Intelligence 3
2.1 Game Theory 3
2.2 Expressivity of game theory concepts for robot motion planning 5
A change in prospective 5
General features of motion planning in multi-agent setting 6
2.3 Important game-theoretical issues outside of the scope of this document 9
Uncertainty on others' decision model: 9
Evolutionary Games, Populations, Multi-agent Reinforcement Learning 9
Bounded rationality 10
Differential games 10

3 Game Theoretic Models of Dynamic Interaction 11
3.1 Basic concepts and definitions 11
Best responses and Nash equilibria 13
Equilibrium selection and admissibility 14
Randomized strategies 15
3.2 Special classes of games 16
Zero-sum games 16
Potential games 20
Congestion games 22
3.3 Multi-stage games 25
Extensive form 25
Feedback games 28
Backward induction on the game tree 29
Loop model 30
Backward induction on the loop model 32
Open-loop and receding horizon Nash equilibria 34

4 The AI-DO Challenge 37
4.1 The AIDO challenges with full state information 38
4.2 How to get started 38
4.3 Evaluation of the challenges 39

Bibliography 41

1 Preface

1.1 To Whom This May Concern . . 1
1.2 Monograph organization 2

The scope of this manuscript is to ease the initial steps for a reader interested in modeling the strategic nature of motion planning problems in a multi-agent environment. Historically, game theory has been devoted to studying *rational* decision-making in numerous fields: social science, economics, system science, logic, computer science, and many more.

Today, as robots leave the factory floors for a more complex world, we do believe that many of the game theoretical concepts are well suited to capture the dynamic and interactive nature of multi-agent motion planning. The promise (and hope) is that explicitly taking into account the others' decision making in its own, endows standard techniques with a richer descriptive power. If this promise holds true, a better decision making for our robots will facilitate a seamless integration in our society.

1.1 To Whom This May Concern

- A reader who has a strong background in game theory but little in motion planning, might discover the relevance of concepts they have always known for robotics applications.
- A reader who has a background in motion planning but knows little of game theory might find enlightening how standard techniques can be

augmented with game theoretical concepts to explicitly take into account others' reasoning in its own decision making.

- If both motion planning and game theory are new concepts, we hope to spark the curiosity and motivate the relevance of the topic.

1.2 Monograph organization

- Chapter 2 introduces and motivates in plain English game theoretical concepts for the task of motion planning. It presents forward references to the corresponding mathematical models that appear later in the document and topics that fall out of scope;
- Chapter 3 presents game theory basics models and some extensions;
- Chapter 5 introduces the reader to two new game-theoretical challenges opened in the context of the Artificial Intelligence Driving Olympics (AI-DO).

2 Game Theory for Embodied Intelligence

2.1 Game Theory 3
2.2 Expressivity of game theory concepts for robot motion planning 5
A change in prospective 5
General features of motion planning in multi-agent setting . . . 6
2.3 Important game-theoretical issues outside of the scope of this document 9
Uncertainty on others' decision model: 9
Evolutionary Games, Populations, Multi-agent Reinforcement Learning 9
Bounded rationality 10
Differential games 10

2.1 Game Theory

Game theory is concerned with the study of strategic decision making among multiple *players* (often called *agents*). Each player chooses from a set of available *actions* (often called *alternatives*) in a bid to maximize her *payoff* (the objective function in this case is called *utility* or *reward*) or to minimize it (here we talk about *cost* function or *loss*). Crucially, the payoff of a player depends also on the other players' actions. This means that is not convenient (or *rational*) for the individual players to blindly optimize their objective without considering how the others would act.

Cooperative vs Non-cooperative. A major distinction in field of game theory is based on whether or not the players can enter into a cooperative agreement. If cooperation is possible, this would imply that the decision making is carried out collectively and everyone would benefit to the possible extent without any inefficiency in the system. This is known as *cooperative game theory*, which deals with issues of bargaining, coalition formation, excess utility distribution, etc.; cooperative game theory will not be part of this tutorial. The interested reader is referred to [21, Chp. 13-15]

This chapter is ...

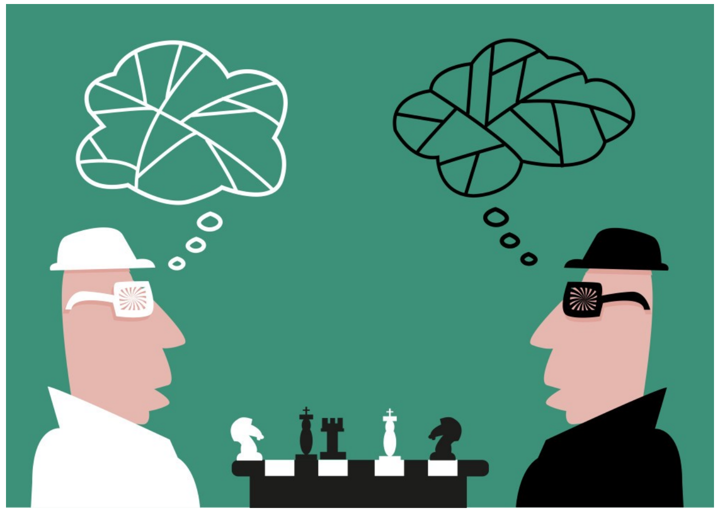
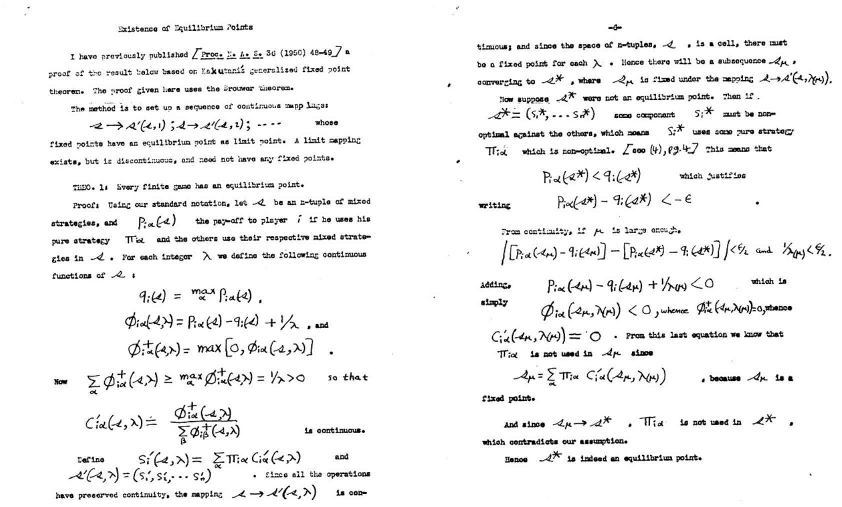


Figure 2.1: Chess is the emblem of strategic decision making for games. As for embodied intelligence, a set of scenarios are laid out in front of you and your actions will have long term consequences. While the game of chess is well structured, indeed everyone has *perfect information* and there is little cache, the world we live in is more complex, the environment where a robot lives is often unstructured and present a high degree of stochasticity.

Of broader interest for the problem at hand is the *non-cooperative* counterpart. Each player carries out the decision making process individually based on the information available (this includes also a decision making model of the others) till there is no incentive anymore to change their decision. If all the players are satisfied with their chosen actions, we say they reached an equilibrium. This idea is what takes on the name of *Nash Equilibrium* in honor of John Nash who introduced it and proved that it always exists at least one equilibrium in games with a finite number of available actions [19]. As we will see more formally in Def. 3.4, a Nash Equilibrium represents a situation in which every player has no incentive to unilaterally deviate from the current strategy. Loosely speaking, it describes the best payoff that each player could expect if all are behaving rationally.

Sometimes the plain concept of Nash Equilibrium has been regarded as not satisfactory in practice due to the implicit assumptions it requires. First all the players need to have full information about the others' objectives, second one assumes that everyone is playing rationally. We will see that all these assumptions that might seem (and probably are) unrealistic for practical problems such as multi-agent motion planning can be relaxed. What is more, game theory offers a much broader set of tools that allows to deal with cases of practical relevance. For instance one can consider dominant and security strategies, partial information, bounded rationality, multi-stage decisions, etc.



Existence of Equilibrium Points

I have previously published [Proc. N. A. S. 36 (1950) 48–49] a proof of the result below based on Kakutani's generalized fixed point theorem. The proof given here uses the Brouwer theorem.

The method is to set up a sequence of continuous mappings: $s \to s'(s,1)$; $s \to s'(s,2)$; ---- whose fixed points have an equilibrium point as limit point. A limit mapping exists, but is discontinuous, and need not have any fixed points.

THEO. 1: Every finite game has an equilibrium point.

Proof: Using our standard notation, let s be an n-tuple of mixed strategies, and $p_{i\alpha}(s)$ the pay-off to player i if he uses his pure strategy $\pi_{i\alpha}$ and the others use their respective mixed strategies in s. For each integer λ we define the following continuous functions of s:

$$q_i(s) = \max_\alpha p_{i\alpha}(s),$$

$$\phi_{i\alpha}(s,\lambda) = p_{i\alpha}(s) - q_i(s) + 1/\lambda, \text{ and}$$

$$\phi^+_{i\alpha}(s,\lambda) = \max[0, \phi_{i\alpha}(s,\lambda)].$$

Now $\sum_\alpha \phi^+_{i\alpha}(s,\lambda) \geq \max_\alpha \phi^+_{i\alpha}(s,\lambda) = 1/\lambda > 0$ so that

$$C'_{i\alpha}(s,\lambda) = \frac{\phi^+_{i\alpha}(s,\lambda)}{\sum_\beta \phi^+_{i\beta}(s,\lambda)}$$ is continuous.

Define $s'_i(s,\lambda) = \sum_\alpha \pi_{i\alpha} C'_{i\alpha}(s,\lambda)$ and $s'(s,\lambda) = (s'_1, s'_2, \dots s'_n)$. Since all the operations have preserved continuity, the mapping $s \to s'(s,\lambda)$ is con-

-6-

tinuous; and since the space of n-tuples, s, is a cell, there must be a fixed point for each λ. Hence there will be a subsequence s_μ, converging to s^*, where s_μ is fixed under the mapping $s \to s'(s, \lambda(\mu))$.

Now suppose s^* were not an equilibrium point. Then if $s^* = (s_1^*, \dots s_n^*)$ some component s_i^* must be non-optimal against the others, which means s_i^* uses some pure strategy $\pi_{i\alpha}$ which is non-optimal. [see (4), pg. 4] This means that

$$p_{i\alpha}(s^*) < q_i(s^*)$$ which justifies

writing $$p_{i\alpha}(s^*) - q_i(s^*) < -\epsilon.$$

From continuity, if μ is large enough,

$$\left| [p_{i\alpha}(s_\mu) - q_i(s_\mu)] - [p_{i\alpha}(s^*) - q_i(s^*)] \right| < \epsilon/2 \text{ and } 1/\lambda(\mu) < \epsilon/2.$$

Adding, $$p_{i\alpha}(s_\mu) - q_i(s_\mu) + 1/\lambda(\mu) < 0$$ which is simply

$$\phi_{i\alpha}(s_\mu, \lambda(\mu)) < 0, \text{ whence } \phi^+_{i\alpha}(s_\mu, \lambda(\mu)) = 0, \text{ whence}$$

$C'_{i\alpha}(s_\mu, \lambda(\mu)) = 0$. From this last equation we know that $\pi_{i\alpha}$ is not used in s_μ since

$$s_\mu = \sum_\alpha \pi_{i\alpha} C'_{i\alpha}(s_\mu, \lambda(\mu))$$, because s_μ is a fixed point.

And since $s_\mu \to s^*$, $\pi_{i\alpha}$ is not used in s^*, which contradicts our assumption.

Hence s^* is indeed an equilibrium point.

Figure 2.2: In 1949, John Nash provided a one-page proof that games with any number of players have a mixed "Nash equilibrium" in his 27-page Ph.D. thesis. In 1994, he was awarded the Nobel Prize for this pioneering work.

2.2 Expressivity of game theory concepts for robot motion planning

A change in prospective

The basic problem of motion planning is to compute a collision-free trajectory for the robot given a representation of the environment. In order to do so, one has to cope with different sources of uncertainty. Some derive uniquely from the ego robot: Sensing and control actuation, for instance, are subject to noise and disturbances, the environment representation can be incomplete (e.g. due to occlusions,...) or present errors (e.g. due to poor perception algorithms). Yet, for robots deployed in our societies there is a new source of uncertainty to take into account: others' decisions and intentions.

The former type of uncertainties–the ones deriving from the ego robot–remain challenging problems in adverse weather conditions (e.g. snow, heavy rain, fog,...) or when one is trying to push the performance to the limit (e.g. drone or car racing). Nevertheless the last decade has seen such impressive strides that we can ascribe these challenges as "solved", at least for "nominal" conditions. A corroborative example is provided by the numerous self-driving companies that are starting to deploy and test fleets of autonomous vehicles around the globe. At the same time, as autonomous vehicles begin to be deployed in urban scenarios we realize that the latter is a new challenge to overcome: uncertainty on others' intents.

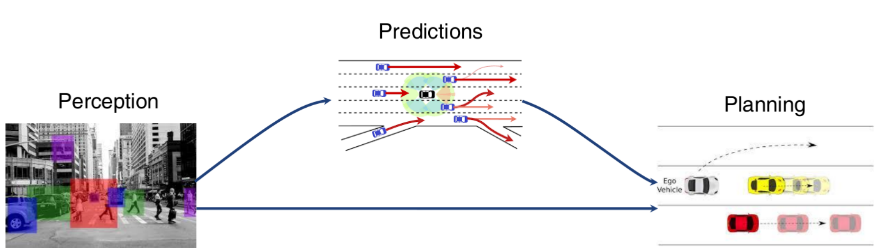


Figure 2.3: Standard motion planning relies on plain perception and predictions. This yields "passive" behaviors where the autonomous vehicle is subordinate to others' decisions since the prediction module does not account for the induced reactions in others' behavior.

A textbook classical planning pipeline can be simplified to Fig. 2.3. The raw data sensors are processed by a perception pipeline into a representation of the environment, predictions of future locations of the other agents are then generated and fed to the planner which is responsible for finding an “optimal” trajectory to follow. What are the shortcomings? The claim is that such a pipeline leads to over-conservative behaviors, the ego-vehicle plans subject to the others’ expected behavior without taking into account that its own actions will create reactions in the others’ behavior. Such an approach leads to passive behaviors. Many videos online show how current autonomous vehicles fail to blend in, for example failing to merge onto the highway.

The paradigm shift required is the following: From predicting what the others will do, we need to plan predicting what the others will do **conditioned on our actions**.

General features of motion planning in multi-agent setting

To study motion planning in a multi-agent setting it is necessary to identify precisely the context. What is the scenario? What information is available to the agents? What is the decision sequence of the agents (if any)?... In the ensuing paragraphs we identify the most relevant characteristics of the possible scenarios with game theoretical language.

Zero-sum vs General-sum games: A remarkable distinction can be drawn between scenarios that are intrinsically adversarial (such as racing) and others that are not (e.g. urban driving). A race is inherently a *zero-sum* game: there is a winner and the losers (or multiple tiers of winners). The “outcome of a race” depends on the joint state which defines a winner ($+1$) and a loser (-1). We will see in Sec. 3.2 that zero-sum games are the ones in which one players’ payoff is equivalent to another’s loss.

Nevertheless, racing scenarios do not describe the interaction of multiple robots simply sharing the same workspace. For instance, in every-day urban driving agents have personal objectives (where to go, driving style preferences,...) but also coupled objectives (avoid collision, keep safety distance,...); all these do not sum to zero in general, hence the name *general-sum* games.

Playing sequence: Another aspect that defines the type of game is the playing sequence of the players. In general here the main distinction is drawn between *simultaneous* or *sequential* play (see Section 3.3 for a more formal distinction). Simultaneous play assumes that the players choose their actions at the same time independently. On the other hand sequential play assumes that the players take turns in selecting their actions, hence before making a decision they have the chance to observe the actions of all the other players; for instance as it happens in a chess match. In general the latter case provides a “simpler" structure which is well suited for sequential games (e.g. poker, chess,...) but could be meaningful also in driving scenarios when there is an inherent asymmetry in the scenario. For example, at an intersection one could model the vehicle coming from the right as a *leader* and the vehicle that yields the right of way as the *follower* giving rise to a particular class of games called Stackelberg games.

Agents in the real world do not take decisions at the exact same instant and they do not necessarily have a fixed playing sequence. In a bid to describe what happens in the real world one should model agents taking decisions asynchronously with a delay on observations of the others' actions. Despite this consideration, the above scenarios are good approximations to capture the structure of the game.

In practice, if there is a clear asymmetry among the players (e.g. one is leading a race the others are behind, one has the right of way,...) it is sensible to model the scenario with a *leader-follower* structure (e.g. [15]). For all the other cases simultaneous play has a meaningful interpretation. One admits that due to perception latency you cannot observe the others' actions till the next time instant. De facto, having a simultaneous decision even if it does not happen precisely at the same time instant.

Solution concepts

- ▹ **Equilibria selection:** Oftentimes games present multiple admissible equilibria (see Def. 3.6), what should then the agents then do? The problem is ill-posed unless one introduces additional assumptions. This issue in the literature takes the name of *equilibria selection* or *equilibria refinement* and it has been extensively studied in the literature for other fields. One way to tackle the problem is to refine the set of equilibria based on some additional criteria (e.g. stability, faireness,...). Another option is to change the rules of the game to have it more well posed with a clear unique solution. In game theoretic context this idea has been widely adopted for auctions to design bidding schemes that encourages the players to bid truthfully. In the literature the act of re-designing the rules of the game to have better posed solutions falls under the name of *mechanism design*). To make a parallel with the context of autonomous agents: The former would correspond to select a special type of equilibrium (e.g. normalized one [14], social optima [34]) or a special structure of the game (e.g. autonomous vehicles subordinate to humans [22]). The latter would correspond to introducing external structures (e.g. traffic lights, which are indeed the classic example of *correlated equilibria*) or rules (yield to who comes from the right).
- ▹ **Security strategies:** Another important aspect is that rather than finding an equilibrium, players might be interested in computing strategies that can guarantee them at least a certain payoff. In this sense it has to be a strategy robust to what the other players might decide to do. Yet again, game theoretical tools come in hand. This idea falls under the name of *security strategies* in the game theoretic context (see Def. 3.13, 3.14). Naturally more advanced concepts of "security" are required in the context of safety critical applications such as autonomous driving that for now fall out of the scope of this tutorial.

Complex decisions The decision making that robots have to face when deployed in our societies are extremely complex. The difficulty arise from the fact that they cannot control any of the external factors: what the others will do, how the environment changes. Moreover there can be non-trivial implications and consequence for one's actions. As one "unfreezes" the robot [31], a risk component is inevitable. But not all risks are the same.

An active debate is rising around the themes of liability issues, ethical decisions, and common sense (e.g. [7]). To give a sense of the complexity of the decisions Fig. 2.4 depicts an example of rules and objectives taken into account at the planning level for AVs.

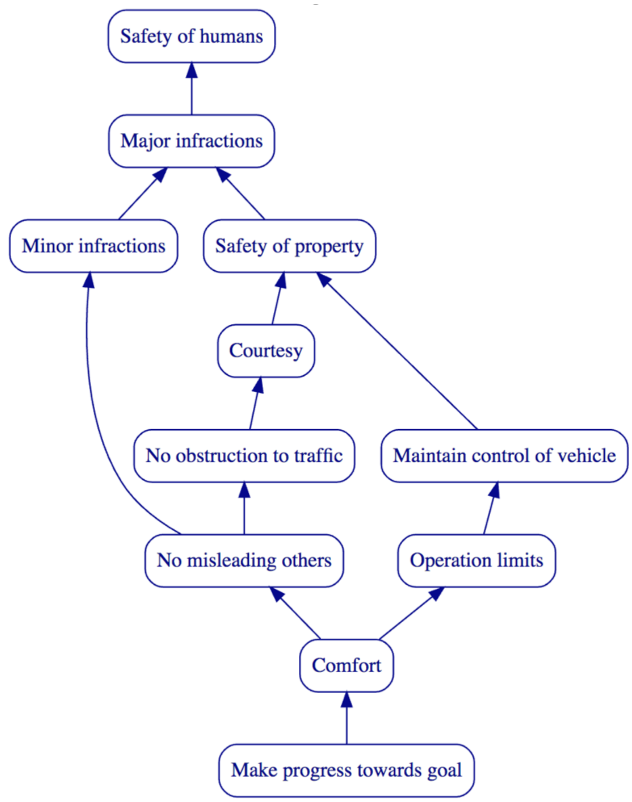


Figure 2.4: The figure shows a hierarchy of rules utilized in [4] to score different trajectories for self-driving cars. The rules are organized in as a direct graph with the most important ones at the top.

Hence, the problem can be decoupled in two: First the agents need to be able to predicted reliably the outcomes of the game given certain actions. Second, given a distribution of uncertain outcome take sensible decisions in *any* situation, even when some rules might be violated. Some work that look at the formalization of such complex decisions are for example [4, 6, 33].

Agents competing for resources Game theory has been successfully deployed in many engineering applications; some problems have in common a particular structure that led to special types of games. In this tutorial we will see *potential* games (Sec. 3.2) where the solution of a game can be computed minimizing a single multivariate function, the potential function indeed. A further example are *congestion* games (Sec. 3.2), a particular type of potential games in which players compete for common resources. This last formulation is typical of every engineering problem where agents compete for common resources, for instance bandwith allocation in radio transmission, or routing in networks (traffic, packets transmissions,...). These formulation turn out to be applicable also to embodied intelligence, aren't after all robots competing for spatio-temporal resources of the environment?

2.3 Important game-theoretical issues outside of the scope of this document

In this tutorial we discuss the themes above somewhat in details. For completeness we want to mention other important themes that fall outside the scope of this manuscript with some pointers to the literature.

Uncertainty on others' decision model:

A common assumption in the standard context is that the decision making model (i.e. the cost functions) of every player is known to everybody. As this assumption does not hold true, it spawns two interesting threads:

- ▹ How to take decisions under uncertainty about the others' model;
- ▹ How to learn others' decision model to shrink the uncertainty.

The problem of learning cost functions from observations falls under the name of "*Inverse Optimal Control*" or equivalently "*Inverse reinforcement learning*". On the topic a good starting point could be the (tutorial) held at ICML 2018. Seminal works such as [1, 20, 29, 35] that have shaped the research on the topic in the last decade. More recent works dealing specifically with this problem in the context of driving are for example [26, 5, 23, 25].

Also in game theory one can find similar concepts devoted to solve similar setups. In games of *incomplete information* each player might not know about the other players' cost function, action set, state, or even the number of players in the game. All these can be translated into the player not knowing which game they are currently in. De facto, playing a meta-game which is composed by a distribution of possible games. Harsanyi in 1967 [11] introduced *Bayesian games* remapping these games to a standard game with an additional player, called *Nature*, who is acting before any of the other players which poses the problem in a more tractable form.

Evolutionary Games, Populations, Multi-agent Reinforcement Learning

Even though this tutorial focuses on the single interactions amongst a handful of players, game theoretical aspects encompass the whole context of mobility.

Some of the single interactions could actually result ill-posed or in a paradox if analyzed in isolation. Famous paradoxes that could apply to the road scenario are the "*game of chicken*" [24] or the "tragedy of the commons" [10], where the only rational solutions are the ones where one player is forced to yield in favor of the other. Yet if no external mechanism is devoted to break the tie in favor of one of the players the problem seems "unsolved".

These cases tend to be better described if we introduce the time dimension. For example defining a meta-game that considers the single game repeated many times; not just the single encounter but n of them. This takes the name of *repeated games*.

As we move from the interactions of a handful of players to many players we enter the realm of *population game theory*. The analysis is now at the population level, which policies are prone to survive? Which are doomed to perish? These aspects are often described under the name of *evolutionary game theory* or *multi-agent reinforcement learning* which shift the focus on how agents can learn to play together developing appropriate policies (i.e. a culture).

Bounded rationality

The idea of *bounded rationality* steams from studies on human decision making, where it has been observed that oftentimes the decisions taken are in some sense "suboptimal" but good enough (e.g. [32]). As humans, robots have resource constraints: limited sensing, limited computation. This sparks the curiosity about approximate solutions, suboptimality and anytime algorithms. How does the game theoretical model are affected if we have limited computational capacity?

Differential games

Differential games are games in which the dynamic behavior of the systems (and therefore also of the interaction between agents) is described by an ordinary differential equation. This is in contrast with the discrete-time version where players interact in "stages", where each stage corresponds to a game on its own. The continuous-time nature of differential games is very much aligned with a lot of robotics problems, because kinematic equations are naturally and originally in continuous time. However, translating the decision strategies of the players in a differential game into protocols and algorithms (that necessarily have to be executed as iterations) is a non-trivial task. An interesting, but mostly unexplored, direction is to consider a hybrid setting (continuous-time dynamics and discrete-time or event-based decisions). Such a setup is already challenging for optimal control problems, and is expected to require a rather complex technical analysis.

3 Game Theoretic Models of Dynamic Interaction

3.1 Basic concepts and definitions . 11
Best responses and Nash equilibria 13
Equilibrium selection and admissibility 14
Randomized strategies 15
3.2 Special classes of games 16
Zero-sum games 16
Potential games 20
Congestion games 22
3.3 Multi-stage games 25
Extensive form 25
Feedback games 28
Backward induction on the game tree 29
Loop model 30
Backward induction on the loop model 32
Open-loop and receding horizon Nash equilibria 34

3.1 Basic concepts and definitions

Game theory is the study of mathematical models of conflict and cooperation between rational decision-makers. In order to define a game, several elements need to be specified:

- ▹ the *players*, i.e. the agents that take the decisions and have an individual preference on the outcome of the same
- ▹ the *actions* available at each player
- ▹ the *information structure* that specifies what information is available at each player before making their decision (for example, other players' decisions)
- ▹ the *outcome* of the game, which in general depends on all players' decision.

Game-theoretic models are extremely general, and can be adapted to a wide class of practical application from extremely diverse fields. In this chapter, we briefly review the notation and the concepts that are useful to formalize the games that are most likely to appear in multi-agent motion planning problems. We refer to [12] for a textbook that shares a similar spirit (although for a larger class of engineering problems), while [2] provides a very comprehensive reference for the more general topic of dynamic non-cooperative games.

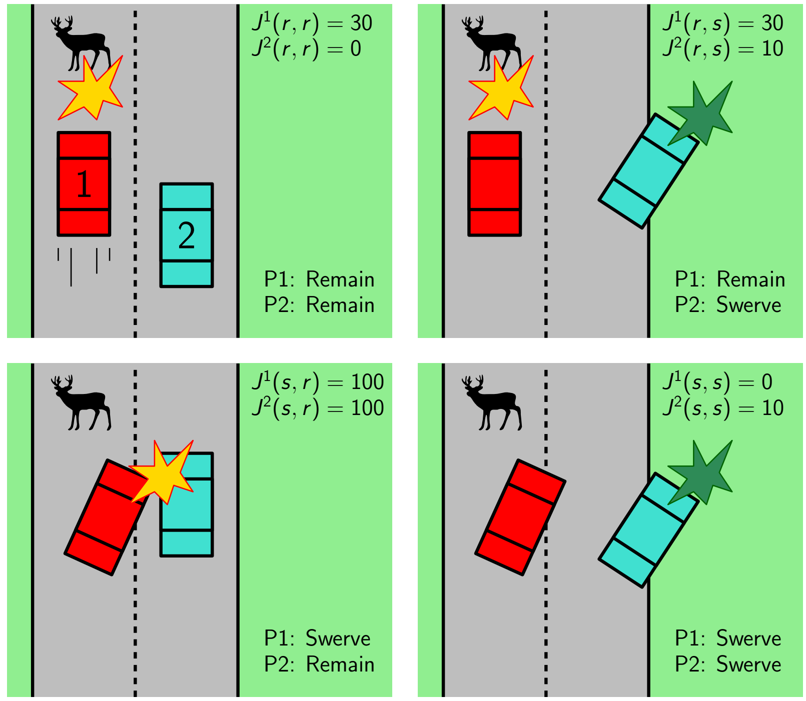


Figure 3.1: Schematic representation of the *emergency-maneuver game*, where two cars have to take a simultaneous decision on how to respond to a sudden obstacle on the road.

In the common case of two-player games with finite action sets, we will refer to the two players as $P1$ and $P2$, endowed with the two action sets

$$\Gamma = \{\gamma_1, \dots, \gamma_n\} \qquad \Sigma = \{\sigma_1, \dots, \sigma_m\},$$

and real-valued outcomes $J^1(\gamma, \sigma)$ and $J^2(\gamma, \sigma)$, respectively. In the more general case with N players, we will denote by $\gamma^i \in \Gamma^i$ the strategy of a player i, and we will use the shorthand notation γ^{-i} to represent the actions of all players but player i. The joint action of all players will be denoted simply by γ.

Example 3.1 (Emergency maneuver)**.** Consider the decision problem that is schematically represented in Figure 3.1. The players in this game are the two drivers, and they have the same actions available to them: to *remain* in their lane or to *swerve* to the right. The information structure in this game is very simple: both players need to take a decision at the same time, without knowing what the other player has decided (we call this setup *simultaneous play*). The outcome of the game is represented via a cost (reported in the figure), which could describe the resulting damage suffered by each of the players' cars.

Games with two players, a finite number of actions, and simultaneous play, are often represented via their *matrix form*, i.e.,

$$\begin{array}{c} \\ \text{Remain} \\ \text{Swerve} \end{array} \begin{array}{c} \begin{array}{cc} \text{Remain} & \text{Swerve} \end{array} \\ \begin{bmatrix} (30,0) & (30,10) \\ (100,100) & (0,10) \end{bmatrix} \end{array} \quad \text{or} \quad A = \begin{bmatrix} 30 & 30 \\ 100 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 10 \\ 100 & 10 \end{bmatrix},$$

where each element of the matrix corresponds to the resulting outcome for the two players:

$$A_{ij} = J^1(\gamma_i, \sigma_j), \qquad B_{ij} = J^2(\gamma_i, \sigma_j).$$

Best responses and Nash equilibria

Once a game is defined, we face the problem of *solving the game*. The interpretation of such a task differs depending on the specific application:

- In some applications, we are interested in *predicting* the decision that agents will make, under the assumption that these agents are rational, informed of the rules of the game, and interested in achieving the best possible outcome; the game is a tool to predict the behavior of agents.
- In some other applications agents are expected to collaborate in order to take joint decisions that exhibit some desirable properties; the formalization of the decision problem as a game is an instrumental step in order to identify the desired joint decision and to design *computational approaches* for the solution of the game.

In either case, solving a game consists in identifying joint players' decisions such that

- no player can improve their own outcome by changing their decision;
- equivalently, no player regrets their decision after observing the outcome;
- if players were to repeat the game, they would make the same decision.

The technical concept that formalizes this idea is based on the definition of what is the player *best response* to the other players' decisions.

Definition 3.2 (Best response). We denote by $R^i(\gamma^{-i})$ the set of actions that yield the best possible outcome to player i, when the other players play strategies γ^{-i}:

$$R^i(\gamma^{-i}) = \arg\min_{\gamma^i \in \Gamma^i} J^i(\gamma^i, \gamma^{-i})$$

Remark 3.3. We are assuming that the set of actions available to each player is fixed and known in advance. This can be extended to the more general case where the action space for each player i is a set $\Gamma^i(\gamma^{-i})$, function of the actions γ^{-i} of the other players. Such a setup is interesting for the problem of motion planning problems that we are considering, because agents' position in space affect the available space for the other players. On the other hand, some care is needed in order to make sure that the simultaneous decision problem remains well posed (how do players know their available actions before observing other players' decisions?) and in order to extend the technical results that we are discussing here to this more general case. We refer the reader to [8] and [9] for a good reference on this class of problems.

With the definition of best response at hand, we can proceed and introduce the core definition of Nash Equilibrium.

Definition 3.4 (Nash equilibrium). A joint strategy γ is a Nash Equilibrium if

$$\gamma^i \in R^i\left(\gamma^{-i}\right) \qquad \forall i.$$

Example 3.5 (Emergency maneuver, cont.). Let us consider the same game as in Example 3.1, and identify the best responses for the two players.

- Player 1's best response is defined element-wise as

$$R^1(\text{remain}) = \text{remain}, \quad R^1(\text{swerve}) = \text{swerve}$$

- Player 2's best responses is defined element-wise as

$$R^2(\text{remain}) = \text{remain}, \quad R^2(\text{swerve}) = \text{swerve}$$

By simple inspection, we can identify two Nash equilibria:

- (remain,remain), with outcome $(30, 0)$
- (swerve,swerve), with outcome $(0, 10)$

Notice that the two Nash equilibria are not interchangeable, i.e., (remain, swerve) and (swerve,remain) are not Nash equilibria.

Equilibrium selection and admissibility

As observed in the previous example, games may admit multiple Nash Equilibria. A natural question is how to predict which equilibrium players will choose (if we are using the concept of Nash equilibria as a predictive modeling tool) or which Nash equilibrium to prescribe. A partial answer is provided by the concept of *admissibility*.

Definition 3.6 (Admissible Nash equilibria). A Nash equilibrium γ^* is admissible if there is no other Nash equilibrium $\tilde{\gamma}$ such that

$$J^i(\tilde{\gamma}) \leq J^i(\gamma^*) \quad \forall i$$

with at least one inequality strict.

In other words, the set of admissible Nash equilibria represent the Pareto-front in the set of all Nash equilibria. Another convenient representation of admissibility makes use of the concept of partial order on the set of Nash equilibria, that we define by saying that

$$\gamma^* \preceq \tilde{\gamma} \quad \text{if} \quad J^i(\gamma^*) \leq J^i(\tilde{\gamma}) \quad \forall i.$$

The minimal elements of the resulting partially-ordered set is the set of admissible Nash equilibria.

Example 3.7 (Emergency maneuver, cont.). The two Nash equilibria that we computed in Example 3.5 are both admissible, as the two corresponding outcomes are not comparable:

$$(30, 0) \underset{\not\succeq}{\overset{\not\preceq}{}} (0, 10)\,.$$

The fact that the admissible Nash equilibrium for this game is not unique captures the difficulty of predicting (or agreeing on) the behavior of rational agents in this game.

Randomized strategies

We now extend the class of decisions that each player can take by allowing them to randomize their action.

Definition 3.8 (Mixed strategy). A mixed strategy is a stochastic strategy where a player selects an action according to a probability distribution over the available action set. In a finite game, mixed strategies are described by discrete probabilities associated to each action:

$$y^i = \begin{bmatrix} y^i_1 \\ \vdots \\ y^i_{n_i} \end{bmatrix}, \quad y^i \in \mathcal{Y}^i = \left\{ y \mid \sum_{h=1}^{n_i} y_h = 1, y_h \geq 0 \right\}, \quad y^i_h = \mathbb{P}[\gamma^i_h]$$

In two-player games we will use the symbols $y \in \mathcal{Y}$, $z \in \mathcal{Z}$ to refer to the mixed strategies of P1 and P2, respectively.

Based on this notion, we can define the concept of *expected outcome* of a game for player i as

$$J^i(y) = \sum_{\gamma \in \Gamma} J^i(\gamma) \mathbb{P}[\gamma] = \sum_{\gamma \in \Gamma} J^i(\gamma) \prod_i \mathbb{P}[\gamma^i]$$

where Γ is the Cartesian product of the players' action spaces, $\mathbb{P} \cdot$ indicates the probability that the joint action γ is played, and we used the fact that players randomize their actions independently.

With a slight abuse of notation, where we refer to deterministic strategies using the same symbols that represent the corresponding actions, we can extend the definition of best response actions.

Definition 3.9 (Best response to a mixed strategy). We denote by $R^i(y^{-i})$ the set of actions that yield the best possible outcome to player i, when the other players play the mixed strategies y^{-i}:

$$R^i(y^{-i}) = \arg\min_{\gamma^i \in \Gamma^i} J^i(\gamma^i, y^{-i})$$

The concept of Nash equilibrium can be extended to include randomized strategies, yielding the following definition and the consequent fundamental existence result.

Definition 3.10 (Mixed Nash equilibrium). A joint mixed strategy y is a Nash Equilibrium if, for each player i, the support of the mixed strategy y^i is a subset of the best response $R^i\left(y^{-i}\right)$.

Theorem 3.11 (Nash existence theorem). A mixed Nash equilibrium is guaranteed to exist in every finite N-player game.

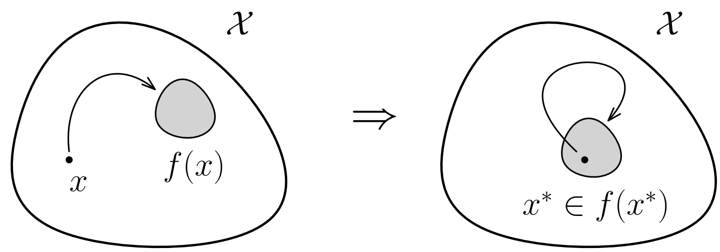


Figure 3.2: A schematic illustration of Kakutani's fixed point theorem, a key technical tool to prove Nash existence theorem.

Proof. The theorem can be proved very concisely using Kakutani fixed-point theorem, which says that every upper-hemicontinuous set-valued function f that maps each point x of a convex compact set $\mathcal{X}$ to a convex closed subset $f(x) \subseteq \mathcal{X}$ has a fixed point (see Figure 3.2).

Given a joint mixed strategy y in $\mathcal{Y}$ (the cartesian product of N simplices), we define the best-response map

$$y \mapsto R(y) = R^1(y^{-1}) \times \dots \times R^N(y^{-N}).$$

Notice that :

- ▷ $R(y)$ is a set-valued map, as the best response may not be unique
- ▷ $\mathcal{Y}$ is compact and convex
- ▷ $R(y) \subseteq \mathcal{Y}$ by construction
- ▷ $R(y)$ is closed (from continuity of J_i in y)
- ▷ $R(y)$ is convex (if two mixed strategies y' and y'' give the same outcome, their mixture $\lambda y' + (1-\lambda)y''$ gives also the same outcome).

We can then apply Kakutani's Fixed Point theorem and conclude that the best-response map has a fixed point, which is, by definition, a mixed Nash equilibrium. □

3.2 Special classes of games

The fundamental concepts that we defined in Section 3.1 are extremely general but offer very few angles of attack for the solution of a game, unless some additional structure of the game is assumed. In this section, we present a small selection of classes of games that are of particular interest for the problem of trajectory planning and at the same time are computationally tractable.

Zero-sum games

A zero-sum game is a 2-player game in which

$$J^1(\gamma, \sigma) = -J^2(\gamma, \sigma), \quad \forall \gamma \in \Gamma, \sigma \in \Sigma.$$

Because of this property, it is customary to indicate a single outcome $J(\gamma, \sigma)$ of the game, corresponding to $J^1(\gamma, \sigma)$, and to refer to Player 1 as the *minimizer* and Player 2 as the *maximizer*.

There exist multiple practical setups in which such a structure is justified:

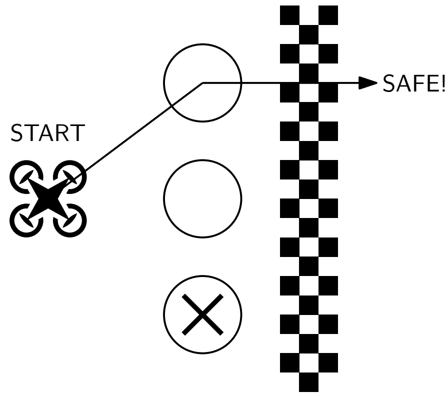


Figure 3.3: A simple zero-sum game for which a pure Nash equilibrium does not exist.

- in some applications, players are truly adversarial (for example, in all games that end with a winner and a loser);
- in some other applications, the adversarial nature of the opponent is a proxy to achieve robustness of the player decision against worst-case scenarios (as an example, see [16], where the curvature of the road ahead of a vehicle is considered the adversarial action of a fictitious player);
- finally, some games that are not zero-sum can be converted into equivalent zero-sum games (an example is *security games*, see [13]).

The notion of Nash equilibrium, specialized to zero-sum games, takes also the name of *saddle-point* of the same, because at a Nash equilibrium $\overline{y}, \underline{z}$ the outcome of the game satisfies the saddle-like relation

$$J(\overline{y}, z) \leq J(\overline{y}, \underline{z}) \leq J(y, \underline{z}) \quad \forall y \in \mathcal{Y},\ z \in \mathcal{Z}.$$

Example 3.12 (Escape game)**.** Consider the simplified escape game represented in Figure 3.3. Player 1 is the quadricopter that needs to reach the same zone, and has three possible paths to do so. Player 2 is adversarial and is trying to block Player 1. Both players, at the same time, need to decide which path to take/block.

The matrix representation of the game is

	block LEFT	block MIDDLE	block RIGHT
go LEFT	$(+1, -1)$	$(-1, +1)$	$(-1, +1)$
go MIDDLE	$(-1, +1)$	$(+1, -1)$	$(-1, +1)$
go RIGHT	$(-1, +1)$	$(-1, +1)$	$(+1, -1)$

and it is evident that no pure Nash equilibrium exists.

A mixed Nash equilibrium exists, and consists in the two mixed strategies

$$y = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}, \qquad z = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}.$$

It is easy to verify that the support of these mixed strategies is the best-response to the other player's mixed strategy.

The following concepts are particularly relevant for zero-sum games, as shown later, although they can be similarly defined for generic games. Intuitively, they represent the most conservative decision that a player can take, under the assumption that other player is allowed to respond to this decision and choose their own action on the basis of that. It is natural to expect that the outcome of the game for a player that adopts this conservative approach is worse (or equal) to what the player can achieve when the two players play simultaneously.

Definition 3.13 (Mixed security levels). Mixed security levels are defined as

$$\overline{V} := \min_{y \in \mathcal{Y}} \max_{z \in \mathcal{Z}} J(y, z) \qquad \underline{V} := \max_{z \in \mathcal{Z}} \min_{y \in \mathcal{Y}} J(y, z)$$

Definition 3.14 (Mixed security policies). A mixed security policy is any

$$\overline{y} \in \arg\min_{y \in \mathcal{Y}} \max_{z \in \mathcal{Z}} J(y, z) \qquad \underline{z} \in \arg\max_{z \in \mathcal{Z}} \min_{y \in \mathcal{Y}} J(y, z)$$

It is easy to prove that $\underline{V} \leq \overline{V}$ (remember that by playing $\overline{y}$ and $\underline{z}$, the two players can guarantee that the outcome of the game is larger than $\underline{V}$ and smaller than $\overline{V}$).

The reason why security policies are particularly interesting in zero-sum games lies in the following result.

Theorem 3.15. The two security levels in a zero-sum game coincide

$$\overline{V} = \min_{y \in \mathcal{Y}} \max_{z \in \mathcal{Z}} J(y, z) = \max_{z \in \mathcal{Z}} \min_{y \in \mathcal{Y}} J(y, z) = \underline{V}$$

and the security policies are saddle-point (Nash Equilibrium) strategies for the game.

Proof. We will prove that if $\underline{V} = \overline{V}$, then the security policies form a Nash Equilibrium, and that if a Nash equilibrium exists, then the two security level coincide. Because a Nash equilibrium is guaranteed to exist by Nash existence theorem, this coincides with the statement of the theorem.[a]

Let us first assume the existence of a pair of strategies y^* and z^* such that

$$J(y^*, z) \leq J(y^*, z^*) \leq J(y, z^*).$$

It then follows that

$$J(y^*, z^*) = \min_{y} J(y, z^*) \leq \max_{z} \min_{y} J(y, z) = \underline{V}$$

$$J(y^*, z^*) = \max_{z} J(y^*, z) \leq \min_{y} \max_{z} J(y, z) = \overline{V}$$

By concatenating the two, we obtain that $\underline{V} \geq \overline{V}$ and, because $\underline{V} \leq \overline{V}$ (as noted before), it must be that $\underline{V} = \overline{V}$.

To prove the other direction, observe that

$$\overline{V} = \min_y \max_z J(y,z) = \max_z J(\overline{y}, z)$$
$$\underline{V} = \max_z \min_y J(y,z) = \min_y J(y, \underline{z}).$$

Because of the definition of min and max, it must be that

$$\underline{V} = \min_y J(y, \underline{z}) \leq Y(\overline{y}, \underline{z}) \leq \max_z J(\overline{y}, z) = \overline{V}.$$

Under the assumption that $\underline{V} = \overline{V}$, these inequalities must be equalities, and correspond to the saddle point condition for the two security strategies $\overline{y}, \underline{z}$. □

[a] This is not the usual procedure to prove the equivalence of security level, nor historically the original one. We refer to von Neumann's minimax theorem [12, Chapter 5] for a longer proof that is based on elementary results from convex analysis and does not require Nash existence theorem (nor any fixed point theorem).

This important results has some immediate consequences. Because the security levels are unique, but multiple security policies could exists, Nash equilibria of a zero-sum game are interchangeable and they are all admissible, as they all return the same value of the game.

Moreover, security levels and security policies can be easily computed via linear programming. This is particularly clear one we highlight the linear nature of $J(y, z)$ via its matricial form:

$$\overline{V} = \min_y \max_z \quad y_i z_j a_{ij} = \min_y \max_j \left(\sum_{i=1}^{m} \quad y_i a_{ij} \right)$$

where in the second equality we used the fact that solutions of a linear program can be found in the set of *basic feasible solutions* (points of the simplex where n constraints are active, corresponding to pure strategies).

This is equivalent to the linear program

$$\begin{aligned} \min_{y,V} \quad & V \\ \text{subject to} \quad & \sum_{i=1}^{m} y_i a_{ij} \leq V, \quad j = 1, \dots, n \\ & y \in \mathcal{Y},\ V \in \mathbb{R} \end{aligned}$$

or in compact form

$$\begin{aligned} \min_{y,V} \quad & V \\ \text{subject to} \quad & A^\top y \leq \mathbf{1} V \\ & y \in \mathcal{Y},\ V \in \mathbb{R} \end{aligned}$$

where $\mathbf{1} = (1, \dots, 1)^\top \in \mathbb{R}^m$ is the vector of ones.

Potential games

Potential games [18] are characterized by a quite technical condition that does not have an immediate connection to an intuitive practical aspect of the game. On the other hand, potential games exhibit a number of convenient features, such as the existence of a Nash equilibrium in pure strategies and the effectiveness of iterative search methods to compute it [28]. These facts alone justify the study of potential games as a special class of games. The potential structure of the game has been recognized in multiple problems that are closely connected to multi-agent motion planning, such as intersection management [3, 34] and cooperative control [17].

Definition 3.16 (Potential game). A N-player game is a potential game if there exists a *potential function* $P : \Gamma^1 \times \Gamma^2 \times ... \times \Gamma^N \to \mathbb{R}$ such that for every player i

$$J^i(\gamma^i, \gamma^{-i}) - J^i(\tilde{\gamma}^i, \gamma^{-i}) = P(\gamma^i, \gamma^{-i}) - P(\tilde{\gamma}^i, \gamma^{-i}) \qquad \forall \gamma^i, \tilde{\gamma}^i, \gamma^{(-i)}.$$

This definition can be extended to a larger class of games, with substantially the same properties, but which only require the existence of an order on the outcomes and on the potential function [34].

Definition 3.17. Ordinal potential game A game is a potential game if there exists an *ordinal potential function* $P : \gamma_1 \times \gamma_2 \times ... \times \gamma_N \to \mathbb{R}$ such that for every player i

$$J^i(\gamma^i, \gamma^{-i}) - J^i(\tilde{\gamma}^i, \gamma^{-i}) > 0 \quad \text{iff} \quad P(\gamma^i, \gamma^{-i}) - P(\tilde{\gamma}^i, \gamma^{-i}) > 0$$

for all $\gamma^i, \tilde{\gamma}^i, \gamma^{(-i)}$.

It is easy to verify that *directional minima* of the potential functions correspond to pure Nash equilibria of the potential game. In particular, global minima of the potential are guaranteed to be directional minima (but not viceversa).

This observation suggests an immediate computational approach to the solution of potential games that taps into nonlinear optimization methods and consists in searching for the global minimum of the potential function. Unfortunately, such a task can become very hard if the potential is non-convex and in general when the number of players is large.

Example 3.18 (Joint maneuver, cont.). Let us consider again the problem describe in Example 3.1, with outcome matrices

$$A = \begin{array}{c} \\ \text{remain} \\ \text{swerve} \end{array} \begin{array}{c} \begin{array}{cc} \text{remain} & \text{swerve} \end{array} \\ \begin{bmatrix} 30 & 30 \\ 130 & 30 \end{bmatrix} \end{array} \qquad B = \begin{array}{c} \\ \text{remain} \\ \text{swerve} \end{array} \begin{array}{c} \begin{array}{cc} \text{remain} & \text{swerve} \end{array} \\ \begin{bmatrix} 0 & 10 \\ 100 & 10 \end{bmatrix} \end{array}$$

One can easily verify that this non-zero-sum game is an ordinal potential game (an exact potential function also exists), with ordinal potential function

$$P(\gamma_1, \gamma_2) = \begin{array}{c} \\ \text{remain} \\ \text{swerve} \end{array} \begin{array}{c} \begin{array}{cc} \text{remain} & \text{swerve} \end{array} \\ \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix} \end{array}$$

As a result, by looking for directional minima of P we recover what we already observed in Example 3.5: the game has two Nash equilibria. Notice that the global minimum of the potential function does not exhibit any special feature; in particular, it is not the only admissible Nash equilibrium of this game. However, the potential function can act as a possible equilibrium selection tool, if the potential represents a meaningful quantity. For example, in some games, the social cost (the sum of all players' outcomes) is a valid potential function, and therefore qualifies the global minimum as the most efficient decision from an aggregate perspective. On the other hand, many other potential games have a potential that is not connected to the overall efficiency of the game (see the next paragraph on congestion games). We refer to [12, Chapter 13] for further examples.

Potential games enjoy significant computational tractability: the iterated update of agents' strategies based on a best-response principle is in fact guaranteed to converge to a Nash equilibrium in potential games with finite actions. The iterated best-response method follows these simple steps.

Consider an initial pure strategy profile $\gamma(0) = (\gamma^1(0), \dots, \gamma^{N)}(0)$. Let $k = 0$.

1. If $\gamma(k)$ is a pure Nash equilibrium $\rightarrow$ stop
2. Else, there exists a player i for which $\gamma^i(k) \notin R^i(\gamma^{-i}(k))$.
3. Update: $\qquad \gamma^i(k+1) \in R^i(\gamma^{-i}(k)), \qquad \gamma^{-i}(k+1) = \gamma^{-i}(k)$.
4. $k = k + 1$, go to step 1.

Convergence of the best-response iteration follows from the fact that the algorithm needs to stop in a finite game, and it needs to stop at a directional minimum of the potential, which corresponds to a Nash equilibrium.

Two variations of the best-response iteration are worth being mentioned. Both these extension refers to the case of continuous action spaces, namely the case in which $\gamma^i \in \mathbb{R}^{n_i}$.

- ▹ [9] proposes a regularized version of the best-response update, where a regularization term $\tau(k)\|\gamma^i(k+1) - \gamma^i(k)\|$ is added in the definition of the best response, and where τ_k is a vanishing weight. This approach is guaranteed to converge, also for the more general case of Generalized Nash equilibria problems.
- ▹ [34] proposes a variation of the best-response update where the decrease of the player's cost at each update needs to be bounded away from zero. The sequence is guaranteed to terminate at a ϵ-Nash-equilibrium, i.e. a point where the players' regret is bounded and arbitrarily small.

Notice that these iterated-best-response methods enjoy a two-fold interpretation. On the one hand, they can be considered as offline computational routines to solve a potential game (and they are also often applied to games that are not potential games, with no a-priori guarantees of convergence). On the other hand, they can be interpreted as online protocols where agents update their decision based on other players' decisions. This class of protocols is distributed/parallelizable and private, as each player only needs to know their own cost function and to solve their own best-response problem.

Congestion games

Congestion games are a class of games that abstract an extremely wide selection of problems in engineering, including problems in robotics, communication, and control.

The main elements of a congestion game are its N players, and the set $\mathcal{M} = \{1, \dots, M\}$ of resources. A player's strategy corresponds to a subset of resources that the player will use:

$$\gamma^i \subseteq \mathcal{M}.$$

Consequently, each resource may be used by one or more players, and we define the load on resource j as the number of players who use it

$$\ell_j(\gamma) := |\{i \mid j \in \gamma^i\}|.$$

The cost incurred by each player depends on the load on the resources that the player is using, i.e.,

$$J^i(\gamma) = \sum_{j \in \gamma^i} f_j(\ell_j(\gamma)),$$

where the function f_j is resource-specific, non-decreasing, and the same for all players.

Example 3.19 (Routing game)**.** Suppose, as in Figure 3.4, that there are two ways to reach city B from city A, and both include some driving and a trip on the ferry. The two paths are perfectly equivalent, the only difference is whether you first drive, or take the ferry. The total time needed to complete the trip depends on what other travellers do.

- ▹ The ferry time is constant, 40 minutes
- ▹ The road time depends on the number of cars on the road.

We consider $N = 200$ travellers, each of which is trying to minimize their travel time.

This game is clearly a congestion game, with four resources (N ferry, N road, S ferry, S road), 200 players, and a resource cost $f_j(\ell_j)$ that describes the time spent on the resource j.

The actions available to each player are

$$\gamma^i = \begin{cases} \{\text{N road}, \text{N ferry}\} & \text{North} \\ \{\text{S road}, \text{S ferry}\} & \text{South} \end{cases}$$

and all players have the same cost function

$$J^i(\gamma^i, \gamma^{-i}) = \begin{cases} 40 + 15 + 0.1\ell_{\text{N road}} & \text{if } \gamma^i = \text{North} \\ 40 + 15 + 0.1\ell_{\text{S road}} & \text{if } \gamma^i = \text{South}. \end{cases}$$

The core observation that guaranteed tractability of the class of congestion games is stated in the following theorem.

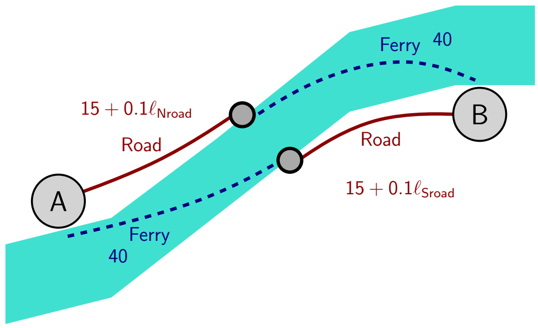


Figure 3.4: Pictorial representation of the routing game, an example of congestion game.

Theorem 3.20. The function

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

is an exact potential function for congestion games.

Proof. Consider a player i, and two joint pure strategies $\gamma = (\gamma^i, \gamma^{-i})$ and $\tilde{\gamma} = (\tilde{\gamma}^i, \gamma^{-i})$.

It suffices to show that

$$P(\tilde{\gamma}) - P(\gamma) = J^i(\tilde{\gamma}) - J^i(\gamma).$$

Note that

- $\ell_p(\tilde{\gamma}) = \ell_p(\gamma) - 1$ for every resource $p \in \gamma^i \backslash \tilde{\gamma}^i$
- $\ell_q(\tilde{\gamma}) = \ell_q(\gamma) + 1$ for every resource $q \in \tilde{\gamma}^i \backslash \gamma^i$
- $\ell_j(\tilde{\gamma}) = \ell_j(\gamma)$ for every other resource j.

$$\begin{aligned} P(\tilde{\gamma}) - P(\gamma) &= \sum_{j=1}^{M} \sum_{k=1}^{\ell_j(\tilde{\gamma})} f_j(k) - \sum_{j=1}^{M} \sum_{k=1}^{\ell_j(\gamma)} f_j(k) \\ &= \sum_{j=1}^{M} \left[\sum_{k=1}^{\ell_j(\tilde{\gamma})} f_j(k) - \sum_{k=1}^{\ell_j(\gamma)} f_j(k) \right] \\ &= \sum_{q \in \tilde{\gamma}^i \backslash \gamma^i} f_q(\ell_q(\tilde{\gamma})) - \sum_{p \in \gamma^i \backslash \tilde{\gamma}^i} f_p(\ell_p(\gamma)) \end{aligned}$$

which corresponds exactly to the difference between

$$J^i(\tilde{\gamma}) = \sum_{j \in \tilde{\gamma}^i} f_j(\ell_j(\tilde{\gamma})) \qquad \text{and} \qquad J^i(\gamma) = \sum_{j \in \gamma^i} f_j(\ell_j(\gamma)).$$

□

This result, in turn, guarantees that pure Nash equilibria for the game exist (and correspond to directional minima of P). This does not exclude the chance that mixed Nash equilibria could also exist, as the following example shows.

Example 3.21 (Routing game, cont.)**.** Consider the same game as in Example 3.19. Many pure Nash equilibria exist. An intuitive way to generate them is to see how the best-response iteration converges to one of them, depending on the initialization and on the order in which players are asked to best-respond. It is easy to verify that the equilibria of the best-response dynamics are all strategies where half of the players use the North path, and half of the players use the South path, yielding a travel time of

$$J^i(\gamma^i, \gamma^{-i}) = 40 + 15 + 0.1 \cdot \frac{200}{2} = 65 \text{ minutes}$$

It is also easy to verify that no other pure Nash equilibrium exist, as in any other partitioning of the agents there would be at least one agent that could improve their cost by changing strategy.

There are however mixed Nash equilibria as well: the simplest to find is the one where all players select one of the two paths with probability 50% (but many others exist).

The fact that an exact potential function is known for congestion games allows to make an immediate observation, based on the fact that this potential function is different from the total cost of all agents (also called *social cost*), that would amount to

$$S(\gamma) = \sum_{j=1}^{M} \ell_j(\gamma) f_j(\ell_j(\gamma)).$$

This implies that, in general, Nash equilibria of a congestion game do not correspond to efficient use of the resources by the users. This inefficiency if quantified by the so-called *price of anarchy*, defined as the ratio

$$PoA := \frac{\max_{\gamma \in \Gamma_{\text{NE}}} W(\gamma)}{\min_{\gamma \in \Gamma} W(\gamma)} \geq 1$$

where Γ is the set of all possible pure strategies for all agents while Γ_{NE} is the set of all strategies which are pure Nash equilibria.

Example 3.22 (Braess paradox). Consider a variation of the game presented in Example 3.19, where a bridge is now available to cross the river, as in Figure 3.5. The bridge is ideal: it has no capacity constraints and it takes not time to cross it (no matter how many people use it).

We can verify by inspection that a new Nash equilibrium emerges, in which all players use both the North and South roads together with the bridge. Their total travel time therefore amounts to

$$J^i(\gamma^*) = 2\,(15 + 0.1 \cdot 200) = 70 \text{ minutes}$$

and they have no incentive to deviate from this path, as using the ferry will yield a travel time of $40 + 15 + 0.1 \cdot 200 = 75$ minutes. Interestingly, the new travel time is larger than the travel time that agents achieved at the Nash equilibrium when no bridge existed (65 minutes), even if those strategies are still available to the players. The resulting price of anarchy is therefore at least

$$PoA \geq \frac{70}{65} = 1.077.$$

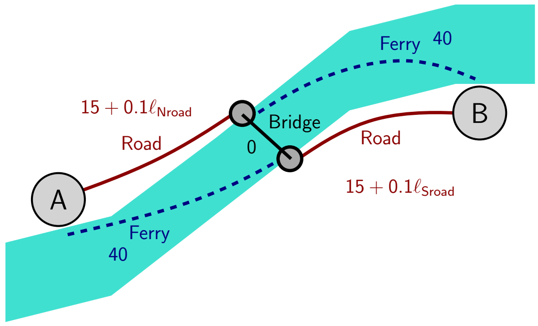


Figure 3.5: Pictorial representation of a variant of the routing game where social inefficiency arises from the selfish behavior of the agents (Braess paradox).

In fact, we have

$$PoA = \frac{70}{64.375} = 1.087$$

where the social-optimal γ at the denominator can be computed numerically and corresponds to

$$\gamma : \begin{cases} \text{50 players on N road - bridge - S road (55 minutes)} \\ \text{75 players on N road - N ferry (67.5 minutes)} \\ \text{75 players on S ferry - S road (67.5 minutes).} \end{cases}$$

3.3 Multi-stage games

In many dynamic interactions, a series of decisions has to be taken by the players involved. As the interaction progresses, more information becomes available, and players can use the available information (including past decisions by the other players) in their decision process.

Such a structure of the decision process requires a different formalism than the one that we saw for static games with simultaneous play, but it can still be effectively studied via the same game-theoretical tools. In fact, we will see that after a careful definition of the key elements of a *multi-stage game*, the previous results still apply.

Extensive form

The *extensive form* is a practical and convenient representation of multi-stage games. As shown in the example in Figure 3.6, multi-stage games are represented as a tree, where

- the game evolves from the root to the leaves
- each node of the tree corresponds to a player's turn to take a decision
- links correspond to actions
- each leaf is associated to a final outcome of the game
- nodes of each player are partitioned into information sets $\mathcal{I}_h^i$ (ambiguity in the player i knowledge)
- at each information set, a set of possible actions $\mathcal{U}_h^i$ is available.

Based on these new elements, the definition of strategies can be accordingly extended to the multi-stage case.

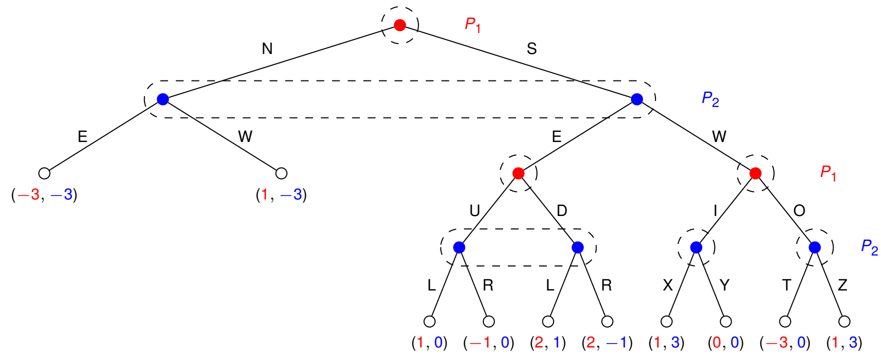


Figure 3.6: An example of multi-stage game represented in extensive form

Definition 3.23 (Pure strategy). A pure strategy for player i is a map that assigns an action to each information set $\mathcal{I}^i_h$

$$\gamma^i : \mathcal{I}^i_h \mapsto u^i_h = \gamma^i(\mathcal{I}^i_h) \in \mathcal{U}^i_h.$$

Example 3.24 (Simple games in extensive form). Figure 3.7 shows two very simple zero-sum games represented in extensive form.

In the case of two players, we use the more compact notation where $\mathcal{I}_h$ and $\mathcal{J}_k$ represents the information sets for the two players, $\mathcal{U}_h$ and $\mathcal{V}_k$ represent the corresponding action spaces, and γ and σ represent their strategies.

The first case is a game with simultaneous play (exactly like those that we have studied in the previous section). We have the information sets and action spaces

$$\begin{aligned} &\text{IS} : \{\mathcal{I}_1\} && \text{IS} : \{\mathcal{J}_1\} \\ &\mathcal{U}_1 = \{N, S\} && \mathcal{V}_1 = \{E, W\} \end{aligned}$$

and the resulting outcome can be represented in the usual matrix representation:

$$\begin{array}{c} \\ \gamma(\mathcal{I}_1){=}N \\ \gamma(\mathcal{I}_1){=}S \end{array} \begin{array}{c} \begin{array}{cc} \sigma(\mathcal{J}_1){=}E & \sigma(\mathcal{J}_1){=}W \end{array} \\ \left[\begin{array}{cc} +1 & -1 \\ 0 & -1 \end{array}\right] \end{array}$$

The second case is a game there Player 2 is aware of Player 1 decision and can use this information (sequential play). The information sets and the action spaces are

$$\begin{aligned} &\text{IS} : \{\mathcal{I}_1\} && \text{IS} : \{\mathcal{J}_1, \mathcal{J}_2\} \\ &\mathcal{U}_1 = \{N, S\} && \mathcal{V}_1 = \{E, W\},\ \mathcal{V}_2 = \{U, D\}. \end{aligned}$$

Games with such a structure (also known as Stackelberg games) can be studied with specialized tools (also in the case of continuous action spaces, see [27]). However, we can still represent this game (and any multi-stage game) in the standard matrix form, by assigning a pure strategy to each row and column of

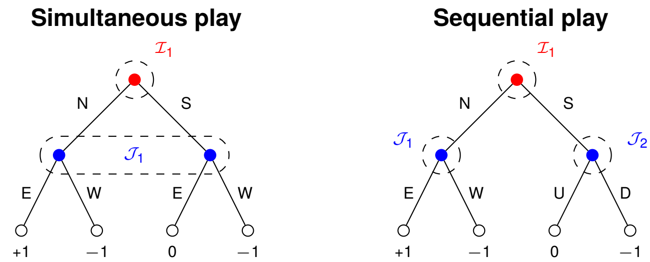


Figure 3.7: A simple example of games represented in extensive form

the matrix:

$$\begin{array}{c} \\ \\ \gamma(\mathcal{I}_1)=N \\ \gamma(\mathcal{I}_1)=S \end{array} \begin{array}{cccc} \sigma(\mathcal{J}_1)=E & \sigma(\mathcal{J}_1)=E & \sigma(\mathcal{J}_1)=W & \sigma(\mathcal{J}_1)=W \\ \sigma(\mathcal{J}_2)=U & \sigma(\mathcal{J}_2)=D & \sigma(\mathcal{J}_2)=U & \sigma(\mathcal{J}_2)=D \end{array}$$
$$\begin{bmatrix} +1 & +1 & -1 & -1 \\ 0 & -1 & 0 & -1 \end{bmatrix}$$

Example 3.24 shows that, while multi-stage games are more naturally represented in their extensive form, it is always possible to represent them in matrix form by adhering to the definition of pure strategies for multi-stage games.

An immediate consequence of this fact is that games in multi-stage form enjoys all the properties and technical results that we have seen for static games. In fact, from a computational point of view, one can solve multi-stage games by simply deriving the static game equivalent and applying the computational methods that we have already reviewed. The main drawback of such an approach is the size of the equivalent static game, as the number of pure strategies increases exponentially in the number of information sets of the player.

The solution of an equivalent static game may consist of mixed strategies. It it worth recalling that a mixed strategy is a probability distribution over the pure strategies $\gamma^i \in \Gamma^i$

$$y^i \in \mathcal{Y}^i \subset \mathbb{R}^{m^i}, \quad m^i = |\Gamma^i|, \qquad y^i_j = \mathbb{P}(\gamma^i_j).$$

In the multi-stage game, such a randomized decision consists in players randomly selecting a pure strategy at the beginning of the game and committing to it for the entire execution of the game. This is not however the only way players can randomize their decision in a multi-stage game, as the following definition shows.

Definition 3.25 (Behavioral strategy)**.** A behavioral strategy for player i is a map that assigns a probability distribution over the actions space $\mathcal{U}^i_h$ available at each information set $\mathcal{I}^i_h$, i.e.,

$$\mathcal{I}^i_h \mapsto \gamma^{i,\text{be}}(\mathcal{I}^i_h) \in \mathcal{Y}^i_h \subset \mathbb{R}^{|\mathcal{U}^i_h|}$$

Mixed strategies and behavioral strategies are not equivalent: there exist multi-stage games where some behavioral strategies do not have an equivalent mixed strategy (in terms of distribution of the outcome over the leaves) and vice versa.

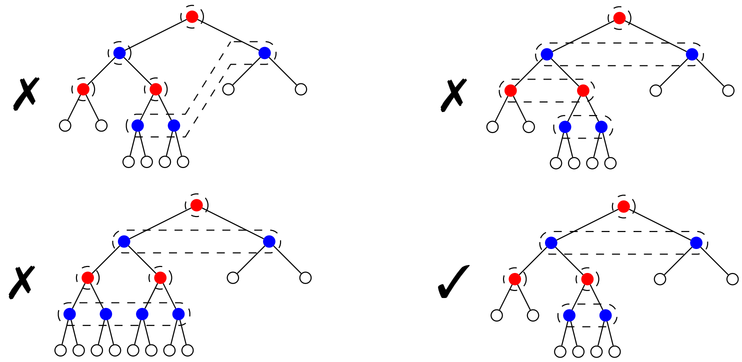


Figure 3.8: Example of feedback and non-feedback multi-stage games.

Feedback games

For the purpose of studying dynamic interaction between agents and multi-agent trajectory planning problems, it is reasonable to restrict the study of multi-stage games to the smaller class of *feedback games*. While this class is not exhaustive, it is large enough to contain most of practical scenarios in these fields. Moreover, it exhibits some important properties that generic multi-stage games don't have, and it is computationally much more tractable.

Definition 3.26 (Feedback games)**.** A multi-stage game is a feedback game (see Figure 3.8) if

1. no information set spans over multiple stages
2. the subtrees in each stage are not "connected" by any information set

The first fundamental property of feedback games is the close relation between behavioral strategies and randomized strategies, as stated by the following results.

Theorem 3.27. In a feedback game, a behavioral strategy γ^b corresponds to a mixed strategy y.

Proof. We can construct the equivalent mixed strategy y element by element. Consider the pure strategy γ^i_j. For each information set $\mathcal{I}^i_h$, let $\gamma^i(\mathcal{I}^i_h) = u^*_h \in \mathcal{U}^i_h$.
Then the elements y^i_j of the equivalent mixed strategy y^i are defined as

$$
\begin{aligned}
y^i_j &= \mathbb{P}(\gamma^i_j) = \\
&= \mathbb{P}(u^i_1 = u^*_1, u^i_2 = u^*_2, \dots, u^i_r = u^*_r) = \\
&= \mathbb{P}(u^i_1 = u^*_1)\,\mathbb{P}(u^i_2 = u^*_2) \cdots \mathbb{P}(u^i_r = u^*_r).
\end{aligned}
$$

Notice that we used the independence of the randomization at different information sets (in the behavioral strategy), and the fact that in a feedback game you don't visit the same information set twice. □

Theorem 3.28 (Kuhn's theorem)**.** In a feedback game, for any mixed strategy y^i for Player i, there exists a behavioral strategy $\gamma^{i,\mathrm{be}}$ for Player i such that, for any mixed strategy y^{-i} played by the other players, y^i and $\gamma^{i,\text{be}}$ yield the same probability distribution over the leaves of the tree, and therefore over the outcomes.

While Kuhn's theorem does not state that mixed strategies are behavioral strategies, the two classes are equivalent for all purposes (in feedback games). This result is particularly powerful from a computational perspective: the set of behavioral strategies is much smaller than the set of mixed strategies (as one can see by looking at the dimension of the two sets), and a behavioral solution can be found via backward iteration on the game tree, as we are going to see next.

Backward induction on the game tree

The modular structure of behavioral strategies (i.e., the fact that they are made of independent randomized decisions at each information set) allow to compute Nash equilibria behavioral strategies in an inductive way, from the leaves from the root of the game tree. The following example shows how a single level of the tree can be solved, and provides the building block for the backward iteration solution of the multi-stage game.

Example 3.29. Single-stage feedback games Consider the zero-sum game in extensive form represented in Figure 3.9. A behavioral saddle-point strategy can be found by following there steps.

1. For each $\mathcal{J}_k$, $k = 1, 2$, construct the corresponding matrix game where the edges entering in $\mathcal{J}_k$ are the actions for Player 1, and the edges leaving $\mathcal{J}_k$ are the actions for Player 2.

	N	S
L	+1	−1
M	−1	+1

	A	B	C
R	−2	0	+1

2. Compute the mixed Nash equilibrium for each matrix game. The resulting Nash equilibrium mixed strategy for Player 2 is their Nash equilibrium behavioral strategy.

$$\sigma^*(\mathcal{J}_1) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \quad \sigma^*(\mathcal{J}_2) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

3. Assign the value of the corresponding matrix game to each information set $\mathcal{J}_k$.

$$V_1 = 0, \quad V_2 = +1$$

4. The behavioral Nash equilibrium for Player 1 is given by the mixed strategy corresponding to the most favorable set $\mathcal{J}_k$.

$$\gamma^*(\mathcal{J}_1) = \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \end{bmatrix}, \quad V = 0$$

With such an approach to solve single-stage games in extensive form, we can attack a multi-stage game recursively, from the leaves towards the root of the tree.

The resulting backward-induction procedure will follow these steps:

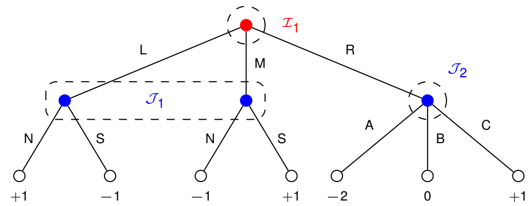


Figure 3.9: A single-stage feedback games in extensive form.

1. Apply the single stage-procedure to each single-stage game at the last stage of the game (these are independent games thanks to the feedback structure of the game). We obtain
   - a pair $(\gamma^{(K)*}, \sigma^{(K)*})$ of NE behavioral strategies for stage K
   - the value of each one of these single stage games.

$$J\left(\left\{\gamma^{(1,\dots,K-1)}, \gamma^{(K)*}\right\}, \left\{\sigma^{(1,\dots,K-1)}, \underset{\blacktriangle}{\sigma^{(K)}}\right\}\right)$$
$$\leq J\left(\left\{\gamma^{(1,\dots,K-1)}, \underset{\blacktriangle}{\gamma^{(K)*}}\right\}, \left\{\sigma^{(1,\dots,K-1)}, \underset{\blacktriangle}{\sigma^{(K)*}}\right\}\right)$$
$$\leq J\left(\left\{\gamma^{(1,\dots,K-1)}, \underset{\blacktriangle}{\gamma^{(K)}}\right\}, \left\{\sigma^{(1,\dots,K-1)}, \sigma^{(K)*}\right\}\right)$$

2. Label the roots of these games with the value of the games rooted in them, and prune the tree. These roots become leaves of the remaining tree.
3. Repeat the recursive procedure on the new tree of height $K-1$.

Implicitly, when the game is reduced by one stage (by pruning the leaves), the new behavioral strategy that is returned by the procedure is a saddle-point strategy assuming that Nash-equilibrium strategies are played in the following stages. This is evident if we explicitly write the saddle-point condition for the second-last stage:

$$J\left(\left\{\gamma^{(1,\dots,K-2)}, \gamma^{(K-1)*}, \gamma^{(K)*}\right\}, \left\{\sigma^{(1,\dots,K-2)}, \underset{\blacktriangle}{\sigma^{(K-1)}}, \sigma^{(K)*}\right\}\right)$$
$$\leq J\left(\left\{\gamma^{(1,\dots,K-2)}, \underset{\blacktriangle}{\gamma^{(K-1)*}}, \gamma^{(K)*}\right\}, \left\{\sigma^{(1,\dots,K-2)}, \underset{\blacktriangle}{\sigma^{(K-1)*}}, \sigma^{(K)*}\right\}\right)$$
$$\leq J\left(\left\{\gamma^{(1,\dots,K-2)}, \underset{\blacktriangle}{\gamma^{(K-1)}}, \gamma^{(K)*}\right\}, \left\{\sigma^{(1,\dots,K-2)}, \sigma^{(K-1)*}, \sigma^{(K)*}\right\}\right)$$

For this reason, the backward-iteration procedure is guaranteed to return a *subgame perfect* Nash equilibrium in behavioral strategies. It is easy (but a bit tedious) to prove that a subgame-perfect Nash equilibrium in behavioral strategies is a Nash equilibrium in behavioral strategies. In fact, subgame perfection is a refinement of the concept of Nash equilibrium, and a particularly desirable one in the applications that we are considering.

Loop model

While backward induction is a more efficient way to compute a Nash equilibrium strategy (and a subgame perfect one!) compared to the analysis of the equivalent static game, it is still a computationally intensive task. In fact, a static game needs

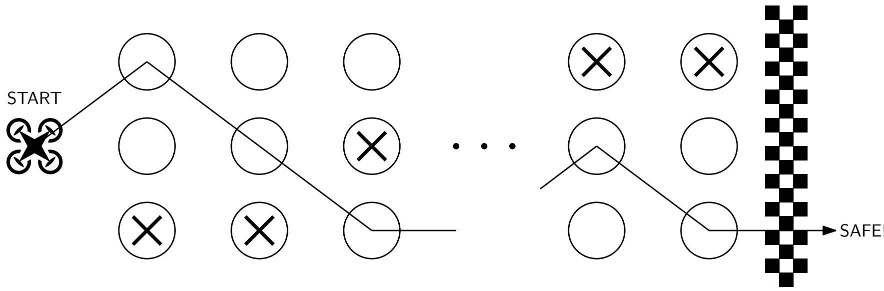


Figure 3.10: Schematic representation of the multistage escape game, a 2-player multistage zero-sum game.

to be solved per each information set, and the number of information sets is directly linked to the number of available actions.

Example 3.30 (Multistage escape game)**.** Consider a variation of the escape game presented in Example 3.12, but in which the path of Player 1 to safety is made of multiple steps, and Player 2 can block Player 1 at any of these steps (Figure 3.10.

If we were to represent this game in extensive form, we would obtain a tree where the number of information sets is exponential in the number of stages. Namely, as 3 actions are available to both players at each stage, it is of the order of $1 + 9 + 81 + \ldots + 3^{20}$ information sets. In order to solve the game via backward induction on the game tree, we would have to solve a static game per information set. Even considering that zero-sum games are easy to solve via linear programming, this would amount to approximately 10^9 linear programs.

It is immediate to notice that many of these programs are however identical: all subgames at a given stage k in which Player 1 is located at the same row are identical for the purpose of determining the final outcome of the game, as the actions taken in the previous stages are inconsequential. For this reason, no more than 3 linear program need to be solved per stage, limiting the total number of games to be solved to 3×20 games.

In the following, we show how a proper choice of a *state* of the game, together with some minimal structure of the decision process, make multistage games tractable when the action spaces for the players is very large, or even continuous. We focus in particular on this latter case (action spaces being a compact subset of $\mathbb{R}^n$), to highlight the connection of this setup with optimal control problems. We also restrict the presentation to the case of two players, for the sake of compactness of the notation, although the extension to N players does not present any difficulty.

Consider a feedback game in extensive form in which at each stage k

- ▷ Player 1 action is $u_k \in \mathcal{U}_k \subseteq \mathbb{R}^m$
- ▷ Player 2 action is $v_k \in \mathcal{V}_k \subseteq \mathbb{R}^n$.

Let us also assume that there exist a state of the game x_k that evolves according to

$$x_{k+1} = f(x_k, u_k, v_k).$$

In general, this assumption is weak: x_k could simply contain the entire history of the player's actions, and we would recover a representation that is quite equivalent to the extensive form that we have seen already. The dynamic model that we are deriving becomes interesting when a "compact" (as in: parsimonious) representation of the state of the game exists.

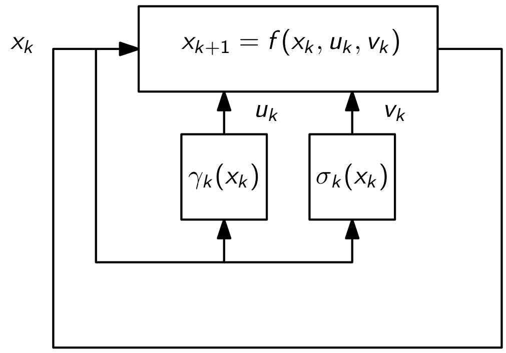


Figure 3.11: Loop model representation of a two-player dynamic game.

We also assume that the outcome is *stage-additive*:

$$J^i = \sum_{k=1}^{K} g_k^i(x_k, u_k, v_k)$$

The consequence of having a Markovian state (containing enough information about the past to predict the future evolution of the game) and a stage-additive cost is that at each stage, the state is a complete representation of the game for all strategic decision purposes. The tree representation, which served the purpose of keeping track of the history of players' actions, can be replaced by the compact *loop model* in Figure 3.11.

The connection to discrete-time feedback control is striking: *strategies* (i.e., maps from information sets to actions) correspond to *feedback laws* (i.e., maps from the state to the inputs). In Figure 3.11 we are assuming simultaneous decision by the two players, but variations are possible, including output feedback, delayed feedback, etc.

Backward induction on the loop model

The loop representation of the game allows to solve the game via backward induction efficiently, compared to solving the entire game tree. The resulting procedure is similar to the standard dynamic programming approach for optimal control, but with a Nash-equilibrium condition that replaces the usual Bellman equation. Namely, the following recursion need to be performed.

Let $V_{K+1}^1(x_K) = V_{K+1}^2(x_K) = 0$.

1. Determine γ_K^*, σ_K^* (functions of x_K) that are Nash equilibria for the subgames at the last stage K, rooted in x_K, i.e.

$$\gamma_K^*(x_K) = \arg\min_u g_K(x_K, u, \sigma_K^*(x_K)) + V_{K+1}^1(f(x_K, u, \sigma_K^*(x_K)))$$
$$\sigma_K^*(x_K) = \arg\min_v g_K(x_K, \gamma_K^*(x_K), v) + V_{K+1}^2(f(x_K, \gamma_K^*(x_K), v))$$

2. Let $V_K^1(x_K)$, $V_K^2(x_K)$ be the value of the subgame rooted in x_K for the two players.
3. If $K > 1$, then $K \leftarrow K - 1$, go to step 1.

The feasibility of such procedure clearly depends on how tractable is the computation of the Nash equilibrium for the subgames with outcome equal to the stage cost plus the value function at the next-stage state. In fact, the class where this Nash equilibrium can be computed in explicit form is quite limited, and an instance is reported in the next example.

Example 3.31 (Two-player LQR). Consider the two-player dynamic game where the state of the system evolves linearly according to

$$x_{k+1} = Ax_k + B^1 u_k + B^2 v_k \quad \text{for } k = 0, \dots, K-1.$$

The two players have individual quadratic cost functions

$$\begin{aligned} J^1(u_1, \dots, u_{K-1}, v_1, \dots, v_{K-1}) &= \sum_{k=0}^{K-1} \left(x_k^T Q^1 x_k + u_k^T R^1 u_k \right) + x_K^T S^1 x_K \\ J^2(u_1, \dots, u_{K-1}, v_1, \dots, v_{K-1}) &= \sum_{k=0}^{K-1} \left(x_k^T Q^2 x_k + v_k^T R^2 v_k \right) + x_K^T S^2 x_K \end{aligned}$$

where $S^i \geq 0$ represent a terminal cost, $Q^i \geq 0$ is a stage cost on the state, and $R^i > 0$ is the stage cost on the input.

We define the value of the game at state x in stage k as

$$\begin{aligned} V_k^1(x) &= \min_{u_k, \dots, u_{K-1}} \sum_{s=k}^{K-1} (x_s^T Q^1 x_s + u_s^T R^1 u_s) + x_K^T S^1 x_K \;\text{ when } v_s = v_s^* \\ V_k^2(x) &= \min_{u_k, \dots, u_{K-1}} \sum_{s=k}^{K-1} (x_s^T Q^2 x_s + v_s^T R^2 v_s) + x_K^T S^2 x_K \;\text{ when } u_s = u_s^*. \end{aligned}$$

At the last stage K, the value function is known:

$$V_K^1(x) = x^T S^1 x, \qquad V_K^2(x) = x^T S^2 x$$

We conjecture that the value function is quadratic for all k, i.e.,

$$V_k^1(x) = x^T P_k^1 x, \qquad V_k^2(x) = x^T P_k^2 x,$$

and we solve the static stage game with costs

$$\begin{aligned} J^1 &= x_k^T Q^1 x_k + u_k^T R^1 u_k + V_{k+1}^1(Ax_k + B^1 u_k + B^2 v_k) \\ J^2 &= x_k^T Q^2 x_k + v_k^T R^2 v_k + V_{k+1}^2(Ax_k + B^1 u_k + B^2 v_k). \end{aligned}$$

As both costs are quadratic in the player's decision, the best responses $\hat{u}_k(v_k)$ and $\hat{v}_k(u_k)$ can be computed in closed form and it's a linear function of the other player's decision:

$$\begin{aligned} \hat{u}_k(v_k) &= -\left(R^1 + {B^1}^T P_{k+1}^1 B^1\right)^{-1} {B^1}^T P_{k+1}^1 \left(Ax_k + B^2 v_k\right) \\ \hat{v}_k(u_k) &= -\left(R^2 + {B^2}^T P_{k+1}^2 B^2\right)^{-1} {B^2}^T P_{k+1}^2 \left(Ax_k + B^1 u_k\right). \end{aligned}$$

In order to find a Nash equilibrium for the problem, we need to find the fixed point (u_k^*, v_k^*) that satisfies

$$\hat{u}_k(v_k^*) = u_k^*, \qquad \hat{v}_k(u_k^*) = v_k^*,$$

that is

$$u_k^* = -\left(R^1 + {B^1}^T P_{k+1}^1 B_1\right)^{-1} {B^1}^T P_{k+1}^1 \left(Ax_k + B^2 v_k^*\right)$$
$$v_k^* = -\left(R^2 + {B^2}^T P_{k+1}^2 B_2\right)^{-1} {B^2}^T P_{k+1}^2 \left(Ax_k + B^1 u_k^*\right).$$

As this fixed-point condition is a system of linear equations, its solution is going to be a linear feedback of the state x_k:

$$u_k^* = H_k^1 x_k, \qquad v_k^* = H_k^2 x_k.$$

This allows to compute, by simple substitution, the value of the stage game as

$$V_k^1 = x_k^T P_k^1 x_k, \qquad V_k^2 = x_k^T P_k^2 x_k$$

where

$$P_k^1 = Q^1 + {H_k^1}^T R^1 H_k^1 + (A + B^1 H_k^1 + B^2 H_k^2)^T P_{k+1}^1 (A + B^1 H_k^1 + B^2 H_k^2)$$
$$P_k^2 = Q^2 + {H_k^2}^T R^2 H_k^2 + (A + B^1 H_k^1 + B^2 H_k^2)^T P_{k+1}^2 (A + B^1 H_k^1 + B^2 H_k^2).$$

As the value functions are quadratic, the inductive hypothesis is verified and the recursion can continue towards the root of the game ($k = 1$).

Open-loop and receding horizon Nash equilibria

In most cases, an explicit solution of the backward iteration is not available. It is however possible to compute a numerical solution for the *open-loop* joint problem

$$\begin{aligned}
u_0^*, u_1^*, \dots, u_{K-1}^* \quad &= \quad \underset{u_0,\dots,u_{K-1}}{\arg\min} \quad \sum_{k=0}^{K-1} g_k^1(x_k, u_k, v_k^*) \\
&\quad\ \text{subject to} \quad x_{k+1} = f_k(x_k, u_k, v_k^*) \quad \text{for } k = 0, \dots, K-1 \\
v_0^*, v_1^*, \dots, v_{K-1}^* \quad &= \quad \underset{v_0,\dots,v_{K-1}}{\arg\min} \quad \sum_{k=0}^{K-1} g_k^2(x_k, u_k^*, v_k) \\
&\quad\ \text{subject to} \quad x_{k+1} = f_k(x_k, u_k^*, v_k) \quad \text{for } k = 0, \dots, K-1
\end{aligned}$$

These two optimization problem are coupled, but their joint solution can be characterized (for example via variational inequalities / KKT conditions) and computed numerically. Notice however that the resulting solution is a constant strategy across the state-space: in control-theoretical terms, it is an open-loop optimal strategy, rather than an optimal feedback law. In game-theoretic terms, it is a Nash equilibrium strategy, but not a subgame-perfect one (see Figure 3.12 for a pictorial representation).

Open loop strategies lack robustness: in case of disturbances, actuation noise,

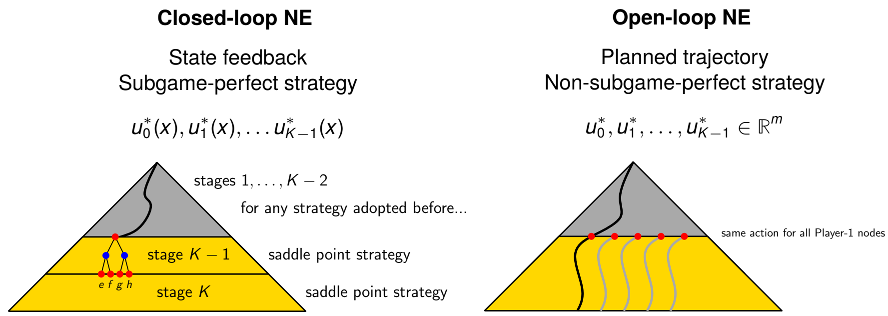


Figure 3.12: A pictorial representation of the difference between open-loop Nash equilibria and closed-loop (subgame-perfect) Nash equilibria.

or imperfect behavior of the agents, they don't represent the Nash equilibrium strategy for the players anymore. If the joint open-loop optimization problem can be solved in real-time, then a practical solution to this problem is to re-initialize the problem and re-solve the problem at every stage, as the game progresses. This receding-horizon property returns a subgame-perfect strategy by construction, at the cost of significant real-time computation effort. The single-player version of this procedure corresponds, in spirit, to the optimization-based control paradigm known as Model Predictive Control.

4 The AI-DO Challenge

4.1 The AIDO challenges with full state information 38
4.2 How to get started 38
4.3 Evaluation of the challenges . . 39

Oftentimes the lack of reproducible benchmarks and scenarios has been regarded as one of the major pitfalls in robotics.

The Artificial Intelligence Driving Olympics (AI-DO) provides a complete framework for reproducible robotics. Participants need to "code" an agent which gets packaged as a docker container to be evaluated in simulation and eventually on real hardware [30].

Beside the classical challenges, we opened two new multi-agent challenges to suit the game theoretical setup. The challenge server takes care of embodying your agent in multiple players and interface them with the multi-agent simulator. Exactly as in the standard game theoretical setup. The protocol of observations and commands is precisely defined for the agent and all the complexity of deploying, and simulating multiple agents is already taken care of; you only need to code the map from observations to commands.

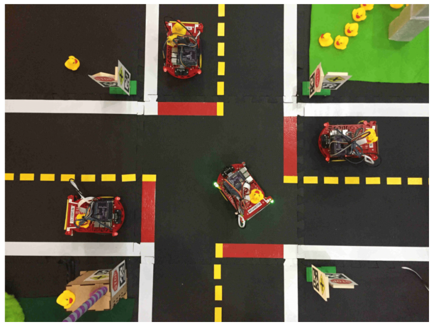


Figure 4.1: The AI Driving Olympics (AI-DO) is a set of competition with the objective of evaluating the state of the art for ML/AI for embodied intelligence. Visit www.duckietown.org/research/ai-driving-olympics for more information.

4.1 The AIDO challenges with full state information

In contrast to the classical challenges where the "observations" of an agent are simply the camera frames, these new challenges feature a protocol that provides full state information about the ego-vehicle and the others.

- ▹ **LFV_multi**: Your agent gets embodied in multiple duckies, the map has no intersections.
- ▹ **LFVI_multi**: Your agent gets embodied in multiple duckies, the map has also intersections. Make sure to properly take into account the nearby vehicles.

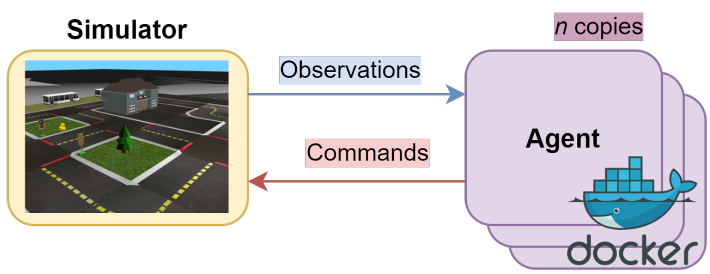


Figure 4.2: Your agents gets embodied in n copies of itself each getting interfaced to the simulator providing observations and requesting commands.

4.2 How to get started

To take part in the AI-DO challenges you need some preliminaries steps to have all the necessary tools (Github, Docker account, Duckietown token). Detailed instructions can be found here.

For the challenges of interest mentioned in Section 4.1

Templates and baseline:

- ▹ The **template** with full state information is available at github.com/duckietown/challenge aido_LF-minimal-agent-full.
- ▹ A **baseline** solution is provided at github.com/idsc-frazzoli/challenge-aido_LF-minimal-agent-full. The example results of the baseline submission can be found at challenges.duckietown.org/v4/humans/submissions/15520.

To work on your own solution, fork the template or the baseline repository.

General resources:

- A general list of resources related to Duckietown can be found here.
- Some useful tutorials to understand the duckietown environment can be found in the form of jupyter notebooks at the duckietown-world repository.

4.3 Evaluation of the challenges

KPIs for the challenge, how to improve the baseline?

Numerous metrics of comfort, time, deviation from the center line are automatically computed for each submission and scenario, see the baseline example.

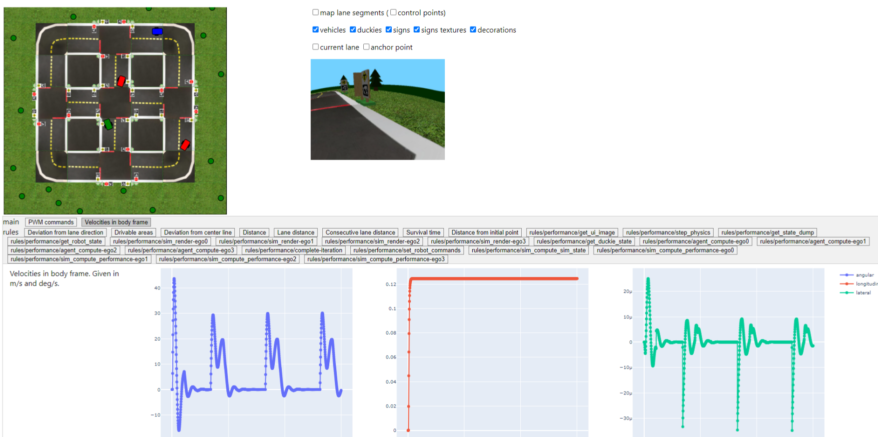


Figure 4.3: An instance of the metrics breakdown that get computed for every submission, these can be used as impartial KPIs for the experiments evaluation.

Bibliography

[1] Pieter Abbeel and Andrew Y. Ng. «Apprenticeship learning via inverse reinforcement learning». In: *Twenty-first international conference on Machine learning - ICML '04*. 346. New York, New York, USA: ACM Press, 2004, p. 1. ISBN: 1581138285. DOI: 10.1145/1015330.1015430. URL: https://ai.stanford.edu/~ang/papers/icml04-apprentice.pdfhttp://portal.acm.org/citation.cfm?doid=1015330.1015430.

[2] Tamer Başar and Geert Jan Olsder. *Dynamic Noncooperative Game Theory, 2nd Edition*. Ed. by SIAM. Vol. 85. 3. Society for Industrial and Applied Mathematics, Jan. 1998, pp. 687–688. ISBN: 978-0-89871-429-6. DOI: 10.1137/1.9781611971132. URL: https://linkinghub.elsevier.com/retrieve/pii/0377221795901736http://epubs.siam.org/doi/book/10.1137/1.9781611971132.

[3] Andreas Britzelmeier and Axel Dreves. «A decomposition algorithm for Nash equilibria in intersection management». In: *Optimization* June (2020). ISSN: 10294945. DOI: 10.1080/02331934.2020.1786088.

[4] Andrea Censi et al. «Liability, Ethics, and Culture-Aware Behavior Specification using Rulebooks». In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, May 2019, pp. 8536–8542. ISBN: 978-1-5386-6027-0. DOI: 10.1109/ICRA.2019.8794364. URL: https://ieeexplore.ieee.org/document/8794364/http://arxiv.org/abs/1902.09355.

[5] Simon Le Cleac'h, Mac Schwager, and Zachary Manchester. «LUCIDGames: Online Unscented Inverse Dynamic Games for Adaptive Trajectory Prediction and Planning». In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 5485–5492. URL: https://arxiv.org/abs/2011.08152.

[6] Anne Collin et al. «Safety of the Intended Driving Behavior Using Rulebooks». In: *IEEE Intelligent Vehicles Symposium, Proceedings* Iv (2020), pp. 136–143. DOI: 10.1109/IV47402.2020.9304588.

[7] Julian De Freitas et al. «From driverless dilemmas to more practical commonsense tests for automated vehicles». In: *Proceedings of the National Academy of Sciences of the United States of America* 118.11 (2021), pp. 1–9. ISSN: 10916490. DOI: 10.1073/pnas.2010202118.

[8] Francisco Facchinei and Christian Kanzow. «Generalized Nash equilibrium problems». In: *4OR* 5.3 (Sept. 2007), pp. 173–210. ISSN: 1619-4500. DOI: 10.1007/s10288-007-0054-4. URL: http://link.springer.com/10.1007/s10288-007-0054-4.

[9] Francisco Facchinei, Veronica Piccialli, and Marco Sciandrone. «Decomposition algorithms for generalized potential games». In: *Computational Optimization and Applications* 50.2 (2011), pp. 237–262. ISSN: 09266003. DOI: 10.1007/s10589-010-9331-9.

[10] Garrett Hardin. «The Tragedy of the Commons». In: *Science* 162.3859 (Dec. 1968), pp. 1243–1248. ISSN: 0036-8075. DOI: 10.1126/science.162.3859.1243. URL: https://www.sciencemag.org/lookup/doi/10.1126/science.162.3859.1243.

[11] J.C. Harsanyi. «Games with Incomplete Information Played by "Bayesian" Players, I-III». In: *Management Science* 14.3 (1967), pp. 159–182.

[12] João P. Hespanha. *Noncooperative game theory : an introduction for engineers and computer scientists*. Princeton University Press, 2017, p. 228. ISBN: 9780691175218.

[13] Dmytro Korzhyk et al. «Stackelberg vs. Nash in Security Games: An Extended Investigation of Interchangeability, Equivalence, and Uniqueness». In: *Journal of Artificial Intelligence Research* 41 (June 2011), pp. 297–327. ISSN: 1076-9757. DOI: 10.1613/jair.3269. URL: https://jair.org/index.php/jair/article/view/10712.

[14] Simon Le Cleac'h, Mac Schwager, and Zachary Manchester. «ALGAMES: A Fast Solver for Constrained Dynamic Games». In: *Robotics: Science and Systems XVI*. Robotics: Science and Systems Foundation, July 2020. ISBN: 978-0-9923747-6-1. DOI: 10.15607/RSS.2020.XVI.091. URL: http://arxiv.org/abs/1910.09713.

[15] Alexander Liniger and John Lygeros. «A Noncooperative Game Approach to Autonomous Racing». In: *IEEE Transactions on Control Systems Technology* (2019). ISSN: 1558-0865. DOI: 10.1109/TCST.2019.2895282.

[16] Alexander Liniger and Luc Van Gool. «Safe Motion Planning for Autonomous Driving using an Adversarial Road Model». In: *Robotics: Science and Systems XVI*. Robotics: Science and Systems Foundation, July 2020. ISBN: 978-0-9923747-6-1. DOI: 10.15607/RSS.2020.XVI.044. URL: http://arxiv.org/abs/2005.07691http://www.roboticsproceedings.org/rss16/p044.pdf.

[17] J. R. Marden, G. Arslan, and J. S. Shamma. «Cooperative Control and Potential Games». In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 39.6 (2009), pp. 1393–1407. ISSN: 10834419. DOI: 10.1109/TSMCB.2009.2017273.

[18] Dov Monderer and Lloyd S. Shapley. «Potential Games». In: *Games and Economic Behavior* 14.1 (May 1996), pp. 124–143. ISSN: 08998256. DOI: 10.1006/game.1996.0044. URL: https://linkinghub.elsevier.com/retrieve/pii/S0899825696900445.

[19] John Nash. «Non-Cooperative Games». PhD thesis. Princeton Universtiy, 1950.

[20] Andrew Y. Ng and Stuart Russell. «Algorithms for Inverse Reinforcement Learning». In: *Proc. 17th International Conf. on Machine Learning* (2000), pp. 663–670.

[21] Martin J. Osborne and Ariel Rubinstein. *A course in game theory*. The MIT Press, 2011, p. 368. ISBN: 0262150417.

[22] Lasse Peters et al. «Inference-Based Strategy Alignment for General-Sum Differential Games». In: (Feb. 2020). URL: http://arxiv.org/abs/2002.04354.

[23] Lasse Peters et al. «Inferring Objectives in Continuous Dynamic Games from Noise-Corrupted Partial State Observations». In: (2021). URL: http://arxiv.org/abs/2106.03611.

[24] Anatol Rapoport and Albert M. Chammah. «The Game of Chicken». In: *American Behavioral Scientist* 10.3 (Nov. 1966), pp. 10–28. ISSN: 0002-7642. DOI: 10.1177/000276426601000303. URL: http://journals.sagepub.com/doi/10.1177/000276426601000303.

[25] Dorsa Sadigh et al. «Information gathering actions over human internal state». In: *IEEE International Conference on Intelligent Robots and Systems* 2016-Novem (2016), pp. 66–73. ISSN: 21530866. DOI: 10.1109/IROS.2016.7759036.

[26] Pier Giuseppe Sessa et al. «Mixed Strategies for Robust Optimization of Unknown Objectives». In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics* (Feb. 2020). URL: http://arxiv.org/abs/2002.12613.

[27] Bernhard von Stengel and Shmuel Zamir. «Leadership games with convex strategy sets». In: *Games and Economic Behavior* 69.2 (2010), pp. 446–457. ISSN: 08998256. DOI: 10.1016/j.geb.2009.11.008. URL: http://dx.doi.org/10.1016/j.geb.2009.11.008.

[28] Brian Swenson, Ryan Murray, and Soummya Kar. «On Best-Response Dynamics in Potential Games». In: *SIAM Journal on Control and Optimization* 56.4 (Jan. 2018), pp. 2734–2767. ISSN: 0363-0129. DOI: 10.1137/17M1139461. URL: https://epubs.siam.org/doi/10.1137/17M1139461.

[29] Umar Syed and Robert E. Schapire. «A game-theoretic approach to apprenticeship learning». In: *Advances in Neural Information Processing Systems 20 - Proceedings of the 2007 Conference* (2009), pp. 1–8.

[30] Jacopo Tani et al. «Integrated Benchmarking and Design for Reproducible and Accessible Evaluation of Robotic Agents». In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2020, pp. 6229–6236. ISBN: 978-1-7281-6212-6. DOI: 10.1109/IROS45743.2020.9341677. URL: https://ieeexplore.ieee.org/document/9341677/.

[31] Peter Trautman and Andreas Krause. «Unfreezing the robot: Navigation in dense, interacting crowds». In: *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings* (2010), pp. 797–803. DOI: 10.1109/IROS.2010.5654369.

[32] Gregory Wheeler. *Bounded Rationality*. Ed. by Edward N. Zalta. Fall 2020. Metaphysics Research Lab, Stanford University, 2020. URL: https://plato.stanford.edu/archives/fall2020/entries/bounded-rationality/.

[33] Wei Xiao et al. «Rule-based optimal control for autonomous driving». In: *ICCPS 2021 - Proceedings of the 2021 ACM/IEEE 12th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2021)* (2021), pp. 143–154. DOI: 10.1145/3450267.3450542.

[34] Alessandro Zanardi et al. «Urban Driving Games with Lexicographic Preferences and Socially Efficient Nash Equilibria». In: *IEEE Robotics and Automation Letters* (2021), pp. 1–1. ISSN: 2377-3766. DOI: 10.1109/LRA.2021.3068657. URL: https://ieeexplore.ieee.org/document/9385938/.

[35] Brian D. Ziebart et al. «Maximum entropy inverse reinforcement learning». In: *Proceedings of the National Conference on Artificial Intelligence* 3 (2008), pp. 1433–1438.