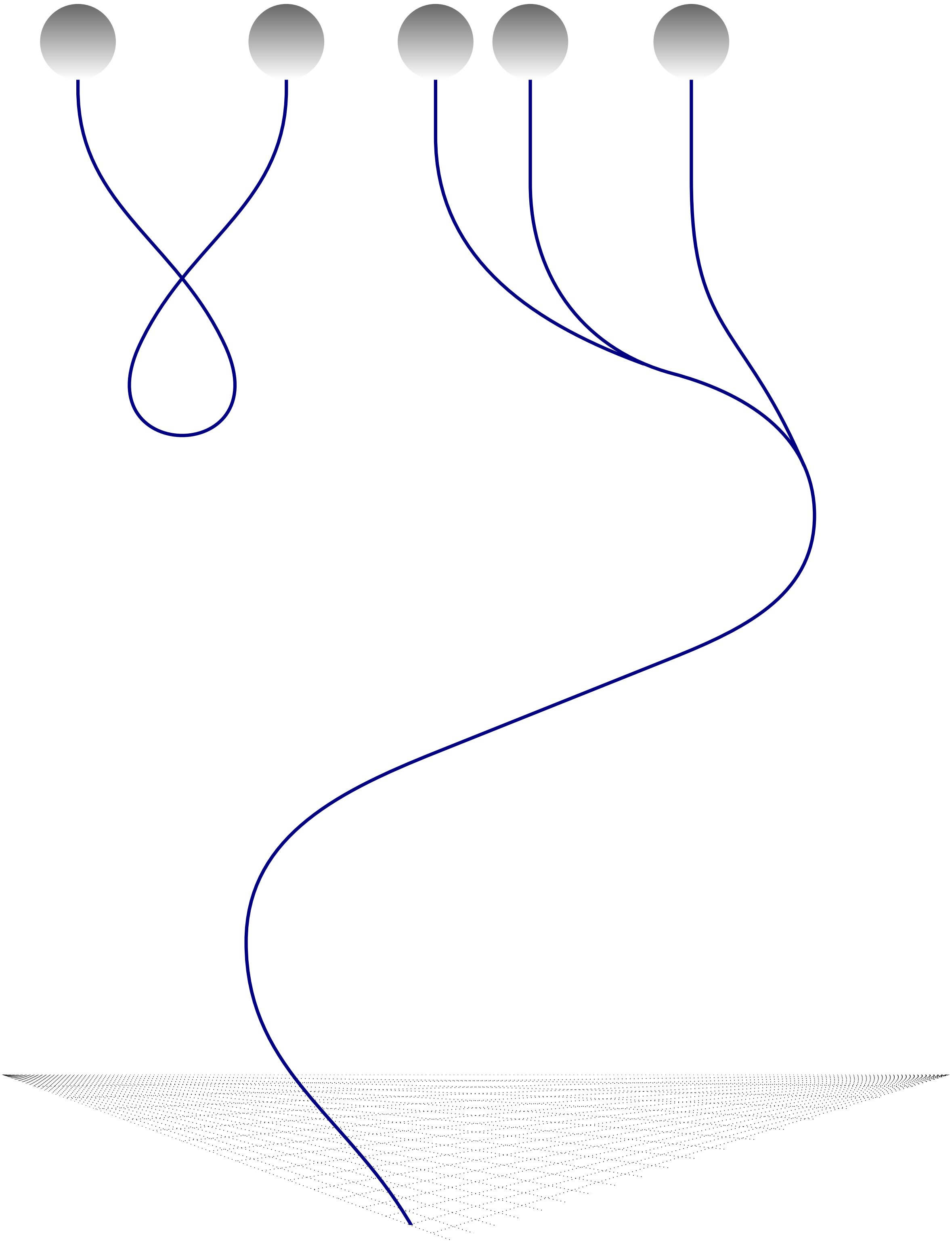


DAVID POENSGEN

ESSAYS ON STOCHASTIC GAMES AND LEARNING IN INTERTEMPORAL CHOICE

Essays on Stochastic Games and Learning in Intertemporal Choice

Inaugural-Dissertation
zur Erlangung des Doktorgrades
des Fachbereichs Wirtschaftswissenschaften
der Goethe-Universität
Frankfurt am Main

vorgelegt von
David Poensgen
aus Frankfurt am Main

Frankfurt am Main 2023

Contents

Introduction and Summary

Intertemporal choice is one of the prime objects of economic study. It is impossible to understand topics such as investment, interest rates, technological advancement, and growth without understanding how firms and societies at large allocate resources between present and future. On a more individual level, time preferences govern decisions relating to saving, education, health, and even labor supply (Golsteyn et al., 2014). The ability to delay gratification in favor of larger goals in the future is often argued to be a main determinant of both success and satisfaction. A second core motif of modern economic analysis is strategic interaction: Few scientific advancements can match the influence of game theory on economics and more generally the social sciences over the last 70 years.

However, there appears to be a divide between these two fields of study and their models: Intertemporal decision making is generally analyzed in the frameworks of partial and general equilibrium. On the other hand, intertemporal tradeoffs often take a rather simplistic form in applied game theoretic models. Extensive form games incorporate some dynamics, but essentially remain one-shot and lack the recursive structure that is typical for intertemporal optimization. And while this structure is present in repeated games, these still presume a static environment in which actions have lasting consequences only through the reactions of others: A quite limiting perspective.

Stochastic games offer a framework that combines both aspects: One can understand them either as multi-player Markov decision processes, or as repeated games with added state transitions (Shapley, 1953; Solan and Vieille, 2015; Mertens et al., 2015). The result is a very general framework that allows to model a wide range of economic situations. Such opportunities notwithstanding, adoption has been slow. A likely reason is the complexity of actually solving stochastic games.

This problem is at the core of three chapters of this cumulative dissertation: Chapters 2 and 3 introduce and discuss algorithms for the computation of stationary equilibria. In the hope that these methods do not remain theoretical possibilities, but become practically used tools, Chapter 4 introduces sgamesolver, a python-based toolbox for stochastic games. In particular, it contains an implementation of the aforementioned methods. The aim of programming this package was to provide ready-made solution tools for stochastic games, so that applied researchers can design and solve stochastic games without heavy investment into learning computational methods. As the odd one out, Chapter 1 presents an experiment – but with direct connection to the topic of intertemporal choice.

The following short summaries give the reader some idea what to expect from the individual chapters. In the spirit of a cumulative dissertation, each chapter is entirely self-contained and can be read independently. Since chapters 2–4 overlap thematically, I could regretfully not avoid some repetition in the introduction of background and key concepts. At the same time, notation is consistent between these chapters, so that the reader may go over the respective sections quickly. It should be noted that Chapters 2 and 3 are rather formal and abstract; readers unfamiliar with stochastic games may consider peeking ahead to Chapter 4, where we tried to include some illustrative examples.

Chapter 1

In the first chapter, I present an experiment that studies learning in a setting where actions have both immediate and delayed consequences. More concretely, subjects make many binary decisions between a total of six abstract options whose value they have to learn by sampling. The core novelty is that the value is disclosed in two parts: One immediately after each choice, and one with a round delay, after the next choice has been made. Although both value components are equally important, subjects systematically react much more strongly to the immediate one. This implies that options with a large immediate component are overvalued, those with a large delayed component undervalued; this resembles the discounting which is typical in intertemporal choice. However, it is a crucial feature of the experiment that it varies the incidence of information, but not of reward: All points earned are paid at the end of the experiment. Thus, the discounting cannot be explained by actual time preferences; rather, it must result from frictions in learning. As the experiment demonstrates, this immediacy bias affects not only behavior, but also beliefs; it is also reflected in characteristic patterns in decision times. A treatment variation lets subjects first learn by observation, without making own decisions; afterwards, subjects make a series of own decisions. The bias is attenuated, indicating that the active decision situation is conducive to the bias.

As I argue, the presence of this immediacy bias has important implications for our understanding of intertemporal choice. First, it suggests that observed discount rates need not always reflect actual time preferences. Outside the experiment, timing of information and reward often coincides; then, observed behavior should combine preference-based discounting and the immediacy bias. But the phenomenon might also be helpful in understanding the formation of time preferences themselves. If preferences are formed by experience, and the impact of experience decreases with delay, immediate experience will have over-proportional influence on preferences. The bias could thereby contribute to a general explanation why impatience and present-bias are such widespread phenomena (Ericson and Laibson, 2019). Moreover, it could help explain individual differences in time discounting. If people are affected to heterogeneous degree, a more severe bias should *ceteris paribus* be associated with less patient preferences. My experiment indeed offers some evidence in that direction. I also elicit answers for a series of hypothetical, intertemporal decision situations. Relating measures of time preferences to individual measures of immediacy bias reveals a significant effect: The more biased subjects are, the less patient their choices.

It is natural to ask why such a bias should exist in the first place. At the end of the chapter, I discuss a potential explanation based on noisy memory and Bayesian decision theory: If immediate feedback is remembered with higher precision, it is actually rational to overweigh it, just as seen in the experiment.

To close this summary with a more personal anecdote: While the chapter has no overt connection to stochastic games – the experiment features no interaction, and the chapter never even mentions the word game – it was actually inspired by my work on them. I was wondering how subjects would learn to play (simple) stochastic games, and quickly came to the hypothesis that instantaneous utilities would be much easier to learn than continuation values, and that subjects would therefore discount much steeper than warranted. I then realized that this effect does not directly rely on strategic interaction, a thought that ultimately resulted in this chapter. Still, I am hopeful to eventually come up with a design that combines my interests in experiments and stochastic games more directly.

Chapter 2

The second chapter is based on work with Steffen Eibelshäuser on Markov quantal response equilibrium (QRE), an adaptation of QRE to stochastic games. This idea has found some mention (Herings and Peeters, 2004) and use (Breitmoser et al., 2010; Battaglini and Palfrey, 2012), but no formal treatment. We focus on a particular variant, logit Markov QRE, which is based on a logit choice rule with precision parameter λ. Following definition and proof of existence, we study three important properties in detail.

First, we show that logit Markov QRE can be given a homotopy interpretation and that the graph of its correspondence is well-behaved in the following sense. It consists of paths and loops that share at most a finite number of intersection points. It always contains a principal branch that starts at the unique QRE at $\lambda = 0$ and converges to a stationary equilibrium of the game as $\lambda \to \infty$. Following this path numerically allows to find a stationary equilibrium for arbitrary finite stochastic games, making it a valuable computational tool. The python package sgamesolver (introduced in Chapter 4) contains a ready-to-use implementation.

The second property we discuss concerns logit Markov QRE as an approximate solution concept. As we show, logit Markov QRE are always ε-equilibria, with a bound for ε that depends on the precision parameter λ and the discount factor, but not on the payoff function of the game. Essentially, by setting λ accordingly, players can guarantee that the loss from following logit choice rather than maximizing does not exceed an arbitrary threshold, no matter what game is played.

Finally, we show a connection of logit Markov QRE to reinforcement learning, specifically to the algorithm *expected SARSA* (van Seijen et al., 2009). We derive a continuous game dynamic from the assumption that all players in a stochastic game learn and play according to expected SARSA, and then show that its stationary points coincide exactly with the set of logit Markov QRE of the game. One reason why this is particularly interesting pertains to the understanding of QRE. A common interpretation is that players still act strategically, i.e. form (accurate) beliefs about others' actions and react to expected payoffs, albeit with mistakes. However, our result shows that logit Markov QRE remains a reasonable solution concept even under much weaker assumptions. It can arise from a dynamic based on a very mechanical form of learning, where players are only required to track own actions and realized instantaneous utilities, without keeping a mental model of the game or other players at all. The dynamic interpretation might also help to give meaning to the traversal of the principal branch mentioned before: In some sense, this resembles all players starting to learn with precision parameter $\lambda = 0$, while then gradually increasing it.

Chapter 3

The third chapter is joint work with Steffen Eibelshäuser, Victor Klockmann, and Alicia von Schenk. We introduce the logarithmic stochastic tracing procedure, a homotopy method to compute stationary equilibria in arbitrary finite discounted stochastic games that improves over existing methods in two ways. First, it is guaranteed to be well-defined for all games of the given class, rather than just generic ones; as almost all games actually studied in applied economics are non-generic, this is an important advantage. The second improvement is speed. The ready-to-use implementation we provide (Chapter 4) is over 500 times faster than the fastest algorithm with comparable scope for which timings have been reported (Dang et al., 2022); at the same time, it allows to solve much larger games in reasonable computation times.

The method is based on the linear tracing procedure for stochastic games by Herings and Peeters (2004), which in turn is based on the method by Harsanyi and Selten (1988) for normal form games. Essentially, the tracing methods work by introducing a prior which is then gradually transformed into equilibrium beliefs. Harsanyi and Selten (1988) understand this as a process of Bayesian strategic reasoning and base their theory of equilibrium selection on this interpretation. Herings and Peeters (2004) show that their linear method allows to compute stationary equilibria in generic stochastic games. However, in non-generic games, the solution set of the homotopy function may contain higher-dimensional subsets that make path tracking impossible. In this chapter, we show that the introduction of logarithmic penalties which are then faded out solves this problem, making our variant well-defined for all games. Moreover, the penalties have a regularizing function and make the homotopy path smooth and interior, thereby improving computational performance significantly.

Because the homotopy path depends on the choice of prior, which is free, the method also allows to search the prior space and uncover potentially multiple equilibria of a given game. This is demonstrated with a practical example in Chapter 4, where we also devote some discussion to the selective properties of the procedure.

Chapter 4

The final chapter, again based on collaborative work with Steffen Eibelshäuser, presents the python package *sgamesolver* in which we implement the homotopy-based solution methods developed in chapters 2 and 3. The main goal behind the package is to provide interested researchers with a ready-made tool to solve stochastic games, without the need to invest heavily into the study of computational methods. In addition to computational performance, ease of use was therefore an important consideration. Finally, we wanted the solution methods to be as general as possible. Which economist has not devised a model, only to realize there is no hope of actually solving it? In an ideal world, modeling decisions could be taken without considering the constraints of a solution technique. That is certainly not attainable, but one can strive for it nonetheless. In that sense we are content that the methods implemented in sgamesolver apply to finite discounted games regardless of their specific structure – and are generally limited only by size.

The chapter itself aims to give a first introduction into using sgamesolver; a more complete online documentation supplements it in this regard. The chapter discusses the general concept of stochastic games and how to define them for use with the program, aided by some examples. We also illustrate how homotopy methods operate in general, the role of the homotopy function, and some specific properties of those implemented. Furthermore, we discuss the principle of the predictor-corrector method which is responsible for actually following the homotopy paths and give a reference of its implementation in sgamesolver. Finally, the slightly more informal nature of this chapter allowed us to also discuss some aspects that found no space in Chapters 2 and 3, for example the selective properties of the logarithmic tracing procedure, and symmetries in stochastic games.

As stated before, we hope that the package will find active usage – previous interaction with users of earlier versions from various disciplines and institutions have always suggested to us that there is demand. Users are encouraged to get in touch, to give us an idea what the package is used for and how it could be improved.

References

BATTAGLINI, M. AND T. R. PALFREY (2012): “The dynamics of distributive politics,” *Economic Theory*, 49, 739–777.

BREITMOSER, Y., J. H. TAN, AND D. J. ZIZZO (2010): “Understanding Perpetual R&D Races,” *Economic Theory*, 44, 445–467.

DANG, C., P. J.-J. HERINGS, AND P. LI (2022): “An Interior-Point Differentiable Path-Following Method to Compute Stationary Equilibria in Stochastic Games,” *INFORMS Journal on Computing*, 34, 1403–1418.

ERICSON, K. M. AND D. LAIBSON (2019): “Intertemporal Choice,” in *Handbook of Behavioral Economics – Foundations and Applications 2*, ed. by B. D, L. D, and D. S, North Holland: Elsevier.

GOLSTEYN, B. H., H. GRÖNQVIST, AND L. LINDAHL (2014): “Adolescent time preferences predict lifetime outcomes,” *The Economic Journal*, 124, F739–F761.

HARSANYI, J. C. AND R. SELTEN (1988): *A General Theory of Equilibrium Selection in Games*, Cambridge, Massachusetts: MIT Press.

HERINGS, P. J.-J. AND R. J. PEETERS (2004): “Stationary Equilibria in Stochastic Games: Structure, Selection, and Computation,” *Journal of Economic Theory*, 118, 32–60.

MERTENS, J.-F., S. SORIN, AND S. ZAMIR (2015): *Repeated Games*, Cambridge: Cambridge University Press.

SHAPLEY, L. S. (1953): “Stochastic Games,” *Proceedings of the National Academy of Sciences*, 39, 1095–1100.

SOLAN, E. AND N. VIEILLE (2015): “Stochastic games,” *Proceedings of the National Academy of Sciences*, 112, 13743–13746.

VAN SEIJEN, H., H. VAN HASSELT, S. WHITESON, AND M. WIERING (2009): “A theoretical and empirical analysis of Expected Sarsa,” in *2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, 177–184.

Chapter **1**

Learned Impatience. Delay of Reinforcement and Time Discounting

Abstract: I study learning in an experimental setting where actions have both immediate and delayed consequences. Subjects make a series of choices between abstract options, with values that have to be learned by sampling. Each option is associated with two payoff components: One is revealed immediately after the choice, the other with one round delay. Objectively, both payoff components are equally important, but most subjects systematically underreact to the delayed consequences. The resulting behavior appears impatient or myopic. However, there is no inherent reason to discount: All rewards are paid simultaneously, after the experiment. Elicited beliefs on the value of options are in accordance with choice behavior. These results demonstrate that revealed impatience may arise from frictions in learning, and that discounting does not necessarily reflect deep time preferences. In a treatment variation, subjects first learn passively from the evidence generated by others, before then making a series of own choices. Here, the underweighting of delayed consequences is attenuated, in particular for the earliest own decisions. Active decision making thus seems to play an important role in the emergence of the bias.

1.1 Introduction

Experience shapes behavior. Actions with satisfying consequences tend to be repeated, those with unpleasant ones avoided – a principle named “law of effect” by Thorndike (1911, p. 244) and studied in countless variations since. On the other hand, economics and increasingly also other social sciences examine behavior primarily through the lens of rational choice theory, where preference rather than experience is the main explanans of behavior. It is natural and important to ask how such analysis can and should incorporate insights from behavioral learning theory. Two pathways from experience to preference immediately spring to mind. The first is learning as a mediator between preference and behavior. It is often argued that the process of learning is an important justification for the assumption of utility-maximizing behavior in the first place (Erev and Roth, 2014). However, where learning involves systematic frictions and adaptation remains incomplete, behavior will match preference only imperfectly. We should then be cautious in relying solely on the principle of revealed preference to explain behavior and conduct welfare analysis. The second pathway is the formation of preference itself: Often, taste is acquired, i.e. learned. Differences in preferences can then be explained by contingencies of experience, a research program that has received increasing empirical validation in recent years. A seminal example is the study by Malmendier and Nagel (2011) on risk attitudes. But it is not only the environment, but also the process of learning itself that should be scrutinized here. If preferences are partly learned, then biases, frictions, and limitations in learning will partly determine what we like.

In this paper, I present a controlled experiment to study a specific learning friction, namely the influence of the delay in the experience of consequences. The core idea is simple. Subjects learn the values of abstract options by sampling; feedback after each choice is given in two parts, one immediate, one delayed. Although both are equally important, I find that subjects heavily discount the delayed feedback, both in choice behavior and stated beliefs. Subjects thus exhibit a costly *immediacy bias*: They overvalue options with a larger immediate value component, and undervalue those where the delayed component is larger. An important feature of the experiment is that it varies the timing of information, but not of reward: All earned points are paid at the same time, at the end of the experiment. Therefore, while the behavior appears to conform to standard time discounting, it can not be explained by time preference, but must be attributed to a bias in learning. A majority of subjects is affected, to heterogeneous degree. The experiment also shows that the immediacy bias is not simply a transient phenomenon, but can have lasting influence on behavior and beliefs. The bias develops quickly and then remains remarkably stable, although subjects have ample time and evidence to correct it throughout the experiment. In a treatment variation, I show that the bias is reduced in a setting where subjects first get to observe the values of the options passively, before then making own decisions.

While it is well established for humans and a wide range of species that learning becomes slower the more time passes between action and reinforcement (Lattal, 2010), the present experiment demonstrates that the effect of varying delay goes beyond the mere speed of learning. It also affects *what* is ultimately learned, here: Which options the subjects end up preferring.

These findings are of twofold importance, in line with the two pathways mentioned earlier. First, the experiment demonstrates the possibility of “as if”-discounting: Behavior that appears impatient, yet is attributable not to deep preference, but to learning frictions. The experiment strictly separates incidence of information and reward; in everyday decisions, they often coincide. In that case, observed impatience will combine actual preference-based discounting and any influence the bias has in the given situation. Welfare analysis should take this into consideration. Moreover, it may also help our understanding of certain empirical regularities, for example the apparent domain specificity of time discounting (Gabaix and Laibson, 2017).

Second, it is possible that the bias plays a role in shaping time preferences themselves. Just like other decisions, inter-temporal trade-offs generate reinforcement, e.g. the instant gratification when indulging, or conversely the satisfaction about a reward for having waited patiently. The bias would then increase the reinforcement value of the former, decrease that of the latter, and thereby systematically skew preferences to become more impatient than they otherwise would be. This paper offers some tentative evidence in this regard: Between subjects, the severity of the bias correlates with typical measures of actual time discounting.

Connecting time preferences and learning frictions might help explain the widespread occurrence of myopia, impatience, and temporal inconsistency, one of the most important areas of research in behavioral economics (see O’Donoghue and Rabin, 2015, and Ericson and Laibson, 2019 for recent overviews). It also suggests why their correction is often so hard: The offending behavior continuously receives reinforcement, which is amplified by the bias. A classical example are physiological addictions, which typically involve substances with almost immediate effect. A related phenomenon is the experience of inner conflict often associated with intertemporal choice. If one adopts the idea of competing systems in decision making, it is natural to think that the more habitual subsystems are most strongly influenced by visceral, most direct consequences. Pursuing distant, abstract goals then often requires to overrule these habitual systems. Many regularly fail in doing so; perhaps a more promising approach is to ensure that behavior aligned with long-term goals also receives regular and early positive reinforcement. The experiment in this paper demonstrates just how important small changes in the timing of such feedback can be.

The paper proceeds as follows. Following a brief discussion of related literature, Sections 1.2 and 1.3 present design and results of the main experiment. Section 1.4 discusses the aforementioned observational treatment and its results. In Section 1.5, I show how a framework of Bayesian updating from noisy memory can offer an explanation for the experimental results. Section 1.6 concludes.

1.1.1 Related Literature

It was already noted by Thorndike (1911, p. 248) that learning slows down with increasing delay between action and consequence, and does eventually cease entirely. This effect has since been documented for a range of species, for example pigeons in Herrnstein (1997, chap. 5).

Commons, Woodford, and coauthors show that stimuli are discounted as temporal distance to action increases, again using data from animal experiments (Commons et al., 1982, 1991). They propose an explanation based on noisy memory, signal detection, and statistical decision theory: If memory is subject to distortions that are additive over time, it is actually optimal to decrease decision weight as delay increases. Gabaix and Laibson (2017) expand this theoretical framework, offering a rational explanation for discounting from information frictions. Their model has agents perform mental simulations to assess the future values of alternatives. If simulation noise increases with time horizon, and simulations are combined with priors to form value estimates, agents will rationally exhibit discounting even when their actual preferences are completely patient.

This paper offers the first clean experimental evidence with human subjects for “as if”-discounting as predicted by these models. In Section 1.5, I will discuss in more detail how these models apply to the experimental setting.

The experimental design follows the so-called *clicking paradigm*, which has found increasing use in the study of decision under risk. In classical list-based experiments, subjects are presented with a list of options and a description of their payoff consequences, say a reward distribution for each. In the clicking paradigm, subjects receive no such information explicitly: Instead, they must learn the distribution by sampling. Interesting differences in choice behavior in the two settings have been documented. Most prominent is the so-called description–experience gap: Rare events are underweighed in the clicking paradigm, while the list method classically evokes overweighting (Hertwig and Erev, 2009). A comprehensive overview is given by Erev and Haruvy (2015).

One interesting result is due to Barron et al. (2008), who investigate order effects between list and clicking methods. Subjects are either given verbal descriptions of lotteries first, and then a chance to sample, or vice versa. Even though subjects' total information is identical afterwards, the final choices in either condition resemble the typical patterns of whichever condition was encountered first. This demonstrates that value judgments formed in a more habitual mode of decision making are not necessarily superseded once complete analytical information becomes available. This suggests that similarly, experience may continue to inform intertemporal choice even where substantial descriptive information is at hand.

Dai et al. (2019) also recently present an application of the clicking paradigm to intertemporal choice. In contrast to this paper, delay is not actually experienced; rather, subjects make choices between lotteries with fixed payment, but with random future payment dates. In a list-like condition, the according distributions are disclosed explicitly; in a clicking condition, subjects get to sample the distribution of different options before then making a binding choice. The authors find a description–experience gap similar to that documented for choice under risk.

The core idea of this paper owes in particular to the work of Herrnstein (1997), who proposed melioration theory as a foundation for the empirical matching law, and as an alternative to the principle of maximization to describe behavior. Melioration theory predicts that behavioral shifts are guided by the local rate of reinforcement, and not global optima; in the temporal context, this of course means immediate gains rather than long-term averages. A series of experiments exist, typically in the clicking paradigm, to demonstrate this effect in dynamic settings. Most well known is the Harvard game, which is closely related to the present experiment, so that brief discussion is justified (a recent overview is Prelec, 2014). Its protocol is as follows: In each of many rounds, subjects can choose either a black or white button. After each choice, a reward is presented whose magnitude depends on past choices, albeit the subject is not told any further specifics. The goal is to maximize total reward in a fixed number of rounds.[1] A choice of black increases the next 10 rewards by 0.2; white increases only the next reward by 1. Subjects are shown only the total reward for each round. Clearly, exclusive choice of black is optimal. However, switching from black to white is always associated with an immediate increase of reward, switching from white to black with an immediate decrease. Consistent with melioration, a substantial share of subjects chooses white often or exclusively.

While elegant in its simplicity, the opaque nature of the experiment limits interpretation. Subjects lack any information on the causal structure of the environment. They do not know how many rounds are (potentially) affected by current choice; whether the solution consists in a single color or a complex sequence, and so on. Using simulations, Sims et al. (2013) demonstrate that a fully rational Bayesian algorithm with perfect memory may need thousands of rounds before arriving at the solution, even when starting from quite reasonable priors. The reason for this is the uncertainty about how far back current consequences have to be attributed. A potential implication is that melioration is not a bias, but a rational response to a fundamentally uncertain environment. Experimental variations of the Harvard game indeed show that lowering complexity reduces the share of meliorizing subjects, for example when fewer rounds are affected by the maximizing option (Prelec, 2014). This raises the question whether melioration would not altogether disappear if subjects reached a clear causal understanding of the reward mechanism. Another reason for cautious interpretation is that the optimal behavior in the Harvard game is a corner solution, making it hard to distinguish erratic from systematically biased behavior.

The experiment I present in this paper is close in spirit, but aims to address these shortcomings. In particular, subjects are explicitly informed about the complete causal structure of the environment. The only aspect not disclosed is a set of payoff vectors which subjects must try to learn. The task is solved easily and quickly by algorithms much simpler than the one mentioned in the preceding paragraph. And while human subjects generally do extract significant value, they also exhibit mistakes in a systematic direction, namely by underweighting delayed feedback. I will discuss these features in more detail in Section 1.2.3.

[1] To be precise, the original implementation has a fixed per-round payment, has choices affect inter-trial delay, and a fixed total duration rather than number of rounds. Both have been used, and the interpretation is of course unchanged.

1.2 Experimental Design

This section discusses the design of the main experiment. For an experiment like this, it is often illuminating to experience it first hand; an online demo is available at `davidpoensgen.github.io/learned-impatience`.

Subjects' task in the experiment was to learn the values of six abstract options, each represented by a distinct color. These values were initially unknown, and could be learned by sampling. The experiment lasted 105 rounds; in each round, subjects were presented two colors and could choose one. Each color would generate a specific amount of points, with a small random perturbation. These points were displayed after each round, allowing subjects to learn throughout the experiment. Payment depended on total points earned, giving subjects an incentive to learn the relative values as quickly and precisely as possible.

As central feature of the experiment, feedback for each choice was not given at once, but split into two components: One shown directly after the choice, one shown with one round delay. Thus, each color x was characterized by two numbers: An *immediate* component x_1 and a *delayed* component x_2, so that its total value was $x_1 + x_2$. Points and feedback were generated as follows. Directly after choosing x, $x_1 + \epsilon$ points were displayed with clear association to the color x and added to the total. One round later, the subject would earn $x_2 + \epsilon'$ points for this same choice, again clearly displayed in association with color x (and alongside the immediate feedback for this round's choice). ϵ and ϵ' were orthogonal noise terms, the purpose of which was to make learning slightly more difficult. Detailed discussion of noise terms and visual presentation will follow below.

Splitting feedback in an immediate and delayed component allows to address the central claim of this paper: Feedback which follows sooner after a decision exerts stronger influence on behavior. This effect is due to information frictions, and can occur independently of any actual preferences that would warrant discounting. I will call this *immediacy bias*; its occurrence is the main hypothesis of the paper.

Hypothesis 1: *Subjects place higher decision weight on x_1 than on x_2. Assuming a latent utility function $u = \delta_1 x_1 + \delta_2 x_2$, this implies $\delta_1 > \delta_2$.*

Importantly, the split into x_1 and x_2 varied the temporal incidence of information, but not of reward: All points were paid out simultaneously, at the end of the experiment. Regardless of individual time preferences, subjects were therefore incentivized to treat x_1 and x_2 symmetrically, and to choose options with higher

Option		Payoff Vectors	
color (e.g.)	(total value)	(immediate: x_1, delayed: x_2)	
		Group A	Group B
🟪	(18)	$(11{,}7)_A$	$(7{,}11)_B$
🟦	(16)	$(6{,}10)_A$	$(10{,}6)_B$
🟦	(14)	$(9, 5)_A$	$(5, 9)_B$
🟩	(12)	$(4, 8)_A$	$(8, 4)_B$
🟧	(10)	$(7, 3)_A$	$(3, 7)_B$
🟥	(8)	$(2, 6)_A$	$(6, 2)_B$

Table 1.1: The six options, valued 8–18, and their payoff vectors for the two groups. Assignment of colors was randomized per subject.

total value $x_1 + x_2$. Observing greater weight on x_1 will be a clear indication that learning frictions can cause behavior that appears impatient (places high weight on immediate outcomes), but can not be explained by actual reward discounting.

1.2.1 Option Values

The central identifying variation was the split of the total value of each option x into the components (x_1, x_2). A direct implication of hypothesis 1 is that options with $x_1 > x_2$ should be overvalued by subjects, and those with $x_1 < x_2$ undervalued. The options in the experiment were designed to allow clean identification of this effect; they are listed in Table 1.1. Subjects were randomly assigned to groups A and B. While the bias is identified within subject, this design helps to rule out potential confounds, as will be detailed shortly.

Subjects in both groups faced six options with total values $8, 10, ..., 18$. These values were split so that alternatingly, x_1 was four points higher, respectively lower, than x_2. This arrangement maximized the number of rounds in which subjects had to choose between options close in value, but with opposite temporal profiles. Hypothesis 1 directly translates to the prediction that subjects make many mistakes in choice sets such as $(7{,}11)_B$ and $(10{,}6)_B$, and few in sets like $(10{,}6)_B$ and $(5, 9)_B$, even though the objective value difference is identical. It is helpful to introduce terminology to distinguish such choice sets. Suppose a choice between x any y, where x is objectively better than y. If $x_1 > x_2$ and $y_1 < y_2$, the choice set will be called *congruent*, as bias and objective value go in the same direction. Conversely, if $x_1 < x_2$ and $y_1 > y_2$, the choice set is called *incongruent*.

Hypothesis 1a: *Error rates are high in incongruent choice sets and low in congruent choice sets.*

Note that by design, subjects could make errors in the opposite direction just as easily and just as often. This allows to differentiate whether observed deviations from optimal behavior are indeed due to temporal ordering, or due to unrelated factors. Consider once more incongruent pairs like $(7{,}11)_B$ and $(10{,}6)_B$: If the bias is strong enough, one will even see reversals in the sense that the objectively worse option is in fact preferred by subjects, i.e. chosen more often or always.

Hypothesis 1b: *If the bias is sufficiently strong, some subjects (or even subjects on average) will prefer worse options with $x_1 > x_2$ over the adjacent, objectively better option with $y_1 < y_2$.*

The preceding hypotheses are within subject, or within group. The bias should also produce a distinct pattern when comparing the two groups. Note that the only difference between the two groups is that x_1 and x_2 are exactly reversed for all options. Consequently, each option should be chosen more often by the group for which it has $x_1 > x_2$: $(11{,}7)_A$ more often than $(7{,}11)_B$, and $(10{,}6)_B$ more often than $(6{,}10)_A$, and so on.

Hypothesis 1c: *When comparing options of equal value between the two groups, the choice frequency is always higher in the group for which the option has $x_1 > x_2$.*

Including two groups with an exact reversal of x_1 and x_2 is also an important safeguard against potential confounds. Suppose some option, e.g. $(10{,}6)_B$, was overvalued by subjects. If the reason for this was unrelated to temporal ordering, e.g. the specific saliency of its payoff numbers, then $(6{,}10)_A$ should be equally overvalued by the other group. If the option is undervalued in one and overvalued in the other, a clean attribution to ordering alone is possible. Similarly, suppose group A reacted more strongly to x_1 across all options. In isolation, a potential explanation would be that x_1 has higher variance than x_2 for group A. But all these properties are exactly reversed for group B, so that if both overreact to x_1, this is again cleanly attributable to temporal ordering.

The other details of the experiment were designed to allow clean identification of the aforementioned effects and rule out potential confounds. They are documented in the following sections.

1.2.2 The Sequence of Choice Sets

In each round, only two rather than all six colors were presented as options. To maximize points earned, subjects therefore had to track a complete ordering over all options, rather than just identify a single best option.

The sequence of these choice sets obeyed the following rules. The 105 rounds were split into 5 blocks of 21 decisions (which followed each other seamlessly). Every block in turn contained each of the 15 possible 2-color combinations, and 6 rounds in which both options were of the same color. These degenerate choice sets were included for several reasons. Most importantly, they forced subjects to sample each color in regular intervals. This greatly limited the need for deliberate exploration, and in particular made sure subjects could not stop sampling a specific color altogether. More generally, note that irrespective of their specific beliefs, subjects would have to sample each color at least once per block, each but the least preferred at least twice per block, and so on. Consequently, even when choices were heavily biased, they would still continuously generate ample evidence to correct this bias.

Using the block structure allowed to collect a rich set of choice data. In particular, a complete measure of subjects' revealed preferences was elicited five times, in regular intervals. The duration also gave choice behavior ample time to stabilize. Together, this allows to address whether the bias is a transient phenomenon which appears early on during learning, but is then corrected when subjects generate more and more information. On the other hand, it is also possible that the higher efficacy of immediate feedback in fact continuously reinforces the biased beliefs on relative values. The bias would then remain stable throughout the experiment. Either alternative has some *ex ante* plausibility; and which one holds true might arguably depend on the specifics of the environment, especially its complexity. However, if the following hypothesis does hold true, it would show that the bias, under the right conditions, can have lasting influence on behavior.

Hypothesis 2: *The bias is stable rather than transient: Once it has developed, it remains stable from block to block.*

The block structure also generates enough observations to measure the bias of individual subjects. In particular, it allows to measure the degree of the bias in a continuous fashion, rather than just detect presence. To do so, one can estimate the ratio of weights placed on x_1 and x_2 per subject, or use the difference in error rates between congruent and incongruent choice sets. Naturally, the expectation is to find considerable heterogeneity in a standard subject population.

Hypothesis 3: *Subjects vary in their degree of immediacy bias. This is reflected in their differential weighting of x_1 and x_2, or in the error rates in different types of choice sets.*

Regarding individual heterogeneity, note that temporal bias is only one of many possible sources of error in the task. It is therefore possible that a subject is biased to some degree, but still scores well in terms of points. On the other hand, a subject may treat x_1 and x_2 symmetrically, but still make many mistakes, simply due to bad memory or lack of concentration. This is a helpful feature, as it allows to separate a specific temporal bias and general ability when relating results from the task to secondary measures.

Subjects were explicitly informed that the order of choice sets was predetermined and independent of their choices. Within the blocks, the order of choice sets was randomized per subject, with the following restriction: If a color was available in round t, it was not available in rounds $t + 1$ and $t + 2$. This ensured that subjects had seen equally many realizations of x_1 and x_2 whenever x was available. The possibility of having more information on x_1 would have been a clear confound. The rule further guaranteed that whenever x was available, neither x_1 nor x_2 could be on screen.

1.2.3 Noise Terms and Task Complexity

Each choice generated points, which primarily depended on the color, but also included a small random disturbance: Choosing x would yield $x_1 + \epsilon$ immediately and $x_2 + \epsilon'$ with one round delay. The disturbances took integer values from 1 to 4 with equal probability, and were independent of choices. In the instructions, subjects were informed about this transparently. They also interacted with a sample task first without and then with disturbances, to get an impression on their impact. Because variance was independent of choices, risk preferences were irrelevant for the task.

Disturbances mainly served as an obstacle to learning. The slight fluctuations make it hard to memorize exact values for the different colors, thus making it more likely that subjects resorted to imprecise estimates. This would allow frictions in learning to play a role.

However, note that the variance is small relative to the value differences between the colors. For options two points apart, the chance of the worse option appearing better is only .14 after a single draw for each, and quickly shrinks with additional sampling. For colors four points apart, this probability is .02, and reversal is impossible for larger value differences.

The experimental environment was designed to concentrate all difficulty in the memorization of values and to avoid other sources of complexity potentially associated with value learning. First is the trade-off between exploration and exploitation, which is of greatly limited importance here. This is partly because variance is so small, partly because choice sets rotate in a way which essentially removes any need for deliberate exploration. The latter ensures presumably inferior options will soon be sampled anyway: Either when up against an even worse option, or when forced in a degenerate choice set.[2]

The causal structure is explicitly revealed to subjects, and values depend only on color, so that subjects do not need to infer any complex rules or memorize states of the world. Subjects clearly see which choice had which consequences. Credit attribution, the major complication in the Harvard game (see Section 1.1.1) is not an issue here.

A simple greedy algorithm with perfect memory solves the task near perfectly; its error rate is below .3% after a few initial samples. Details are found in Appendix 1.A. This illustrates nicely that the disturbances are no hindrance in solving the task, and that no complex strategy or inference are necessary. All it takes is to memorize values. Finally, note that it is not even necessary to treat x_1 and x_2 separately; it suffices to track a single average over both.

1.2.4 Interface and Instructions

The experiment was implemented using oTree (Chen et al., 2016). An online demo can be played at `davidpoensgen.github.io/learned-impatience`. The interface aimed to make the rules as transparent as possible and underlined the structure of the experiment with animations. Figure 1.1 shows the screen which subjects saw throughout the task. At the beginning of each round, two colored buttons appear in region (a), representing the current options. Area (b) shows

[2] Note also that the issue of exploration-exploitation is completely orthogonal to the temporal ordering of feedback: The trade-off would be completely unchanged if feedback was given at once. Thus, even if subjects were mistakenly concerned with exploration, identification of the main hypotheses would be unaffected.

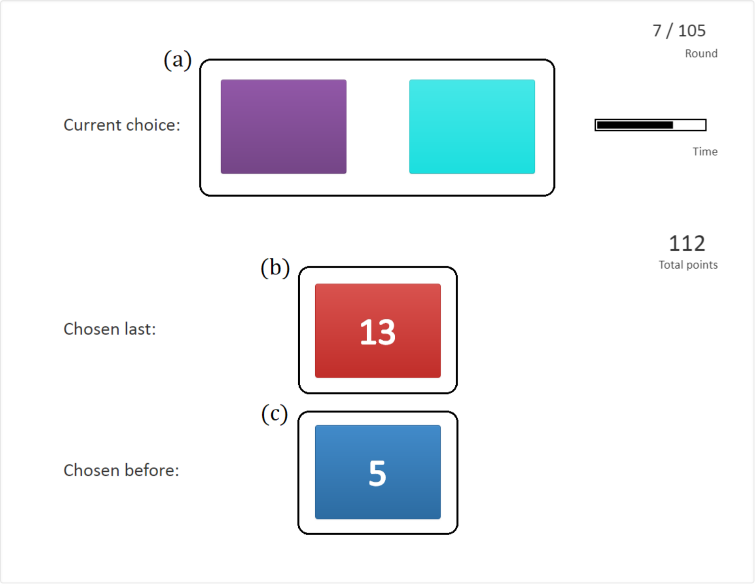


Figure 1.1: Choice screen, as seen by subjects throughout the experiment; boxes a,b,c are added for illustration. Current options are displayed in (a); the most recent choice with immediate feedback in (b); the choice from one round before, with delayed feedback in (c).

the color chosen last round, red, with an immediate payoff of 13, i.e. $red_1 + \epsilon$. Blue, shown in (c), was the choice two rounds ago, with delayed payoff of 5, i.e. $blue_2 + \epsilon'$. All payoffs are visually attributed to the corresponding color, and there is never a need to memorize own past choices.

After the next choice, say purple on the left, the following would happen. The numbers in (b) and (c) fade out. The foregone option disappears. The purple button moves to (b); red moves from (b) to (c); blue leaves the screen downwards. Once this transition is finished, immediate feedback for purple and delayed feedback for red appear simultaneously. This concludes one round of play, and the next begins when two new options appear in (a) 2 seconds later.

Round number and current total points were displayed on the right. Once new options had appeared, subjects had a time limit of 10 seconds per decision, represented by a shrinking bar. If time ran out, a choice was made at random, and a penalty of 5 points deducted. The time limit enforced a steady, comparable pace for all subjects, and prevented excessive use of time to perfectly memorize feedback. Effectively, subjects spent significantly less time per decision than allotted, using about 2 seconds per decision. In over 13,000 observations, only a single timeout occurred.

An important feature of the experiment was complete transparency of its causal structure. Only the payoff vectors were explicitly not disclosed to subjects. They also received no information on their range, average or similar. All other mechanics of the experiment were clearly explained in the instructions. Before starting the main task, subjects played an interactive tutorial, which replicated the main task with slight modifications: Detailed on-screen explanations of all rules could be shown and hidden at any time. Instead of six colors, four shades of gray were used; after a few rounds had been played, their respective values (x_1, x_2) were revealed to make it completely transparent how points are generated.[3] The ϵ-disturbances could be turned on and off at will during the tutorial.

1.2.5 Payment

At the end of the experiment, subjects were paid 0.05€ for each point scored above 1800 in the task, and nothing if below that threshold. This created steep incentives in the relevant region: Each of the 75 non-degenerate decisions effectively had a stake between 0.10€ and 0.50€. Subjects could score between 1715–2065 points (ignoring penalties for timeouts), and random behavior was expected to earn 1890. As intended, all subjects reached the incentivized region by a safe margin. Subjects were paid an additional 2€ for completion of the remaining questionnaire.

1.3 Results

The experiment was conducted at FLEX (Frankfurt Laboratory for Experimental Economic Research) in February and June 2018; participants were recruited using ORSEE (Greiner, 2015). Subjects mean age was 22.5 (sd 3.4), 53.4% were female, all were students with 37% majoring in economics, finance, or business administration, 11% in STEM, and most others in law and social sciences. Sessions lasted around 50 minutes; subjects earned 10.78€ on average.

The results in this section are based on 102 subjects, each playing 105 rounds. Because subjects start without knowledge of values, initial choices must be quite random. All analysis therefore excludes the first block of 21 choices, unless stated otherwise. This represents a natural threshold, because subjects afterwards have seen feedback for every color at least once. In addition, it leaves a well balanced set

[3] Options in the tutorial were valued $(3,6)$, $(1,2)$, $(4,4)$, and $(2,1)$, illustrating that for a given color, the second component could be higher, lower or equal the first, which was also stressed verbally. Moreover, the numbers were chosen such that the second component had higher mean and variance, to avoid any priming that the first might be of more importance.

of observations which includes each binary decision exactly four times per subject. Degenerate choice sets (containing the same color twice) are also excluded from analysis. This leaves 4 blocks with 15 decisions per subject, for a total of 6120 observations. Finally, a single decision is excluded because the time limit was exceeded. Choices are independent between, but not within subjects; all reported tests and standard errors account for this.

1.3.1 Aggregate Choices: Visual Analysis

Aggregated choice frequencies are plotted in Figure 1.2. The resulting graphs are first, visually striking evidence that subjects overreacted to immediate feedback. In particular, all patterns predicted in hypotheses 1a–c are present, as the following explains.

The six options are ordered by total value from 8 to 18 on the x-axis; the y-axis shows choice frequency, conditional on the option being available. To give some orientation: Optimally, one would never pick (8), pick (10) only when paired against (8) and so on. Thus, perfect play would yield the diagonal dotted line. In contrast, completely random behavior would result in the horizontal line at probability 0.5.

The gray graph shows choice behavior of both groups combined. Its slope is roughly halfway between random and perfect play: Subjects did extract significant value from the task, but still erred considerably often. The near linearity of the graph reflects that subjects were able to discriminate better from worse equally well for high- and low-value options.

Red and blue graphs show the same data separated by groups and thereby reveal the highly systematic nature of mistakes. Notice the zigzag-shape of both graphs, which conforms exactly to the predictions of hypotheses 1a–c. First, for all six options, choice frequency is always higher in the group for which $x_1 > x_2$. This was the prediction of hypothesis 1c. The difference is sizable and amounts to roughly 20 percentage points for each option; it is most pronounced for medium values, but only slightly attenuated at the extremes. Further, within each group, choice frequencies are elevated for options with $x_1 > x_2$ and reduced for options with $x_1 < x_2$ – this matches hypothesis 1a. In fact, this effect is strong enough that reversals occur whenever possible: $(6, 2)_B$ is actually chosen more often than $(3, 7)_B$, $(7, 3)_A$ more often than $(4, 8)_A$, and so on. This pattern, predicted by hypothesis 1b, shows that the bias is not only present, but of considerable strength.

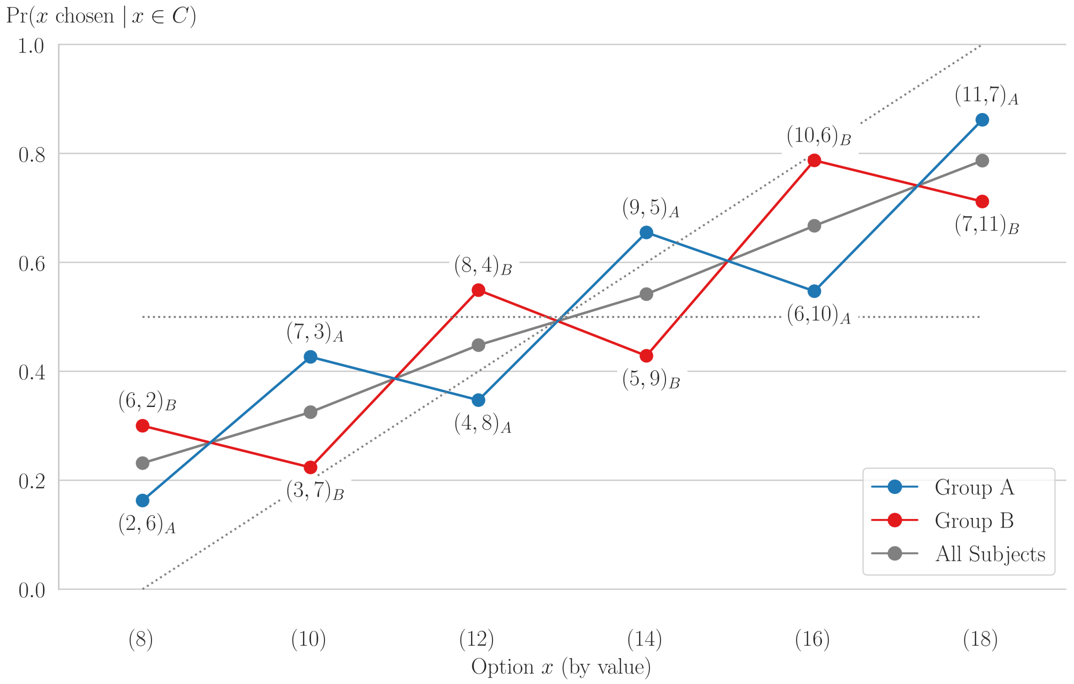


Figure 1.2: Choice frequencies for the 6 options, conditional on availability. Options ordered by total value along the x-axis; the splits into immediate and delayed components are listed alongside the graphs. The horizontal dotted line represents random play, the diagonal dotted line best possible play.

In sum, the graphs clearly illustrate that both groups systematically overreact to immediate feedback, to considerable degree and with regularity across all options.

A natural question is whether the bias is an initial, transient phenomenon and gradually corrected as subjects see more and more evidence. To the contrary, Figure 1.3 shows that the bias is extremely stable. It mirrors the previous graphs, but displays blocks 1–5 separately, thus illustrating the development of choice behavior throughout the task. By design, choices in the first block must be quite random. As expected, the graphs in the first column are therefore flatter; but both Groups A and B already show a hint of the zigzag-pattern. In the second block, the pattern is fully developed. Going forward, changes are small and not systematic. There is no sign that the bias attenuates; if anything, the pattern becomes slightly more pronounced towards the end.

This conforms exactly to hypothesis 2: Subjects form their biased beliefs early on, and do not correct them throughout, even though their choices continuously produce contrary evidence. Cluster-corrected χ^2-tests (not reported) statistically confirm the visual impression. Equality of distributions can be rejected between

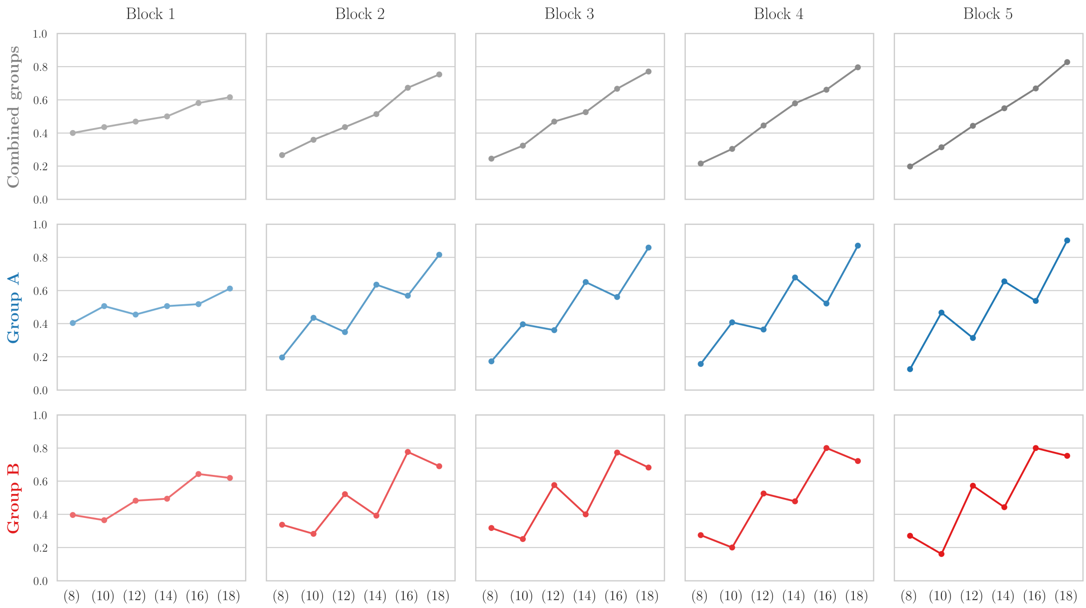


Figure 1.3: Development of choice frequencies over the course of the experiment: Each column represents a block of 21 rounds.

the first and any other block, but not in any pairwise comparison between the later blocks.

1.3.2 Aggregate Choices: Regression Analysis

The bias will now be analyzed in regressions. The dependent variable in all models is the probability of color x being chosen, conditional on the other option y in the choice set: $\Pr(x \text{ chosen}|C_t = \{x, y\})$. Let $\hat{x}_1$ and $\hat{x}_2$ denote the averages of $x_1 + \epsilon$ and $x_2 + \epsilon'$ as seen by the subject up to the current choice; these will serve as explanatory variables.[4]

All models will be based on a latent utility function of the form

$$u(x) \coloneqq \delta_1 \hat{x}_1 + \delta_2 \hat{x}_2$$

The linear probability model (LPM) is then

$$\Pr(x \text{ chosen}|C_t = \{x, y\}) = u(x) - u(y) + 0.5$$

[4] In principle, subjects could react more strongly to the earliest or perhaps the latest realizations they have seen – known as primacy and recency, the latter a typical observation in risky choice (Erev and Haruvy, 2015). In this case, a weighted average would better explain behavior. A comparison of different specifications is performed in Appendix 1.B. A simple average outperforms other alternatives, and will be used throughout. Notably, recency seems to play no role in the current setting.

and the logit specification

$$\Pr(x \text{ chosen}|C_t = \{x, y\}) = \frac{e^{u(x)}}{e^{u(x)} + e^{u(y)}} = \frac{1}{1 + e^{u(y)-u(x)}}$$

This section in addition reports mixed logit models, a refinement of the logit model which will be discussed in more detail when turning to heterogeneity in Section 1.3.4. In principle, all models could be augmented with secondary variables, such as left-right-position, lagged choices or similar. Since these are orthogonal to the variables of interest and the added precision is not needed, this is not pursued here.

The central aim of the experiment is to show that subjects react more strongly to variation in x_1 than in x_2. The regressions offer a direct test: If the bias exists, we should see $\delta_1/\delta_2 > 1$, and the stronger the bias, the higher this ratio. (Conveniently, the coefficient ratio has the same direct interpretation across all specifications, so that accounting for different scaling of the models or marginal effects is not necessary.)

Results are given in Table 1.2. The ratio δ_1/δ_2 is stable across all models, and amounts to over 2.5. This finding is strongly statistically significant: The null hypothesis of $\delta_1 = \delta_2$ is clearly rejected in all models ($p \ll 0.001$), as reported in the table. This confirms hypothesis 1 statistically and thus represents the central result of this paper: Subjects indeed place substantially more weight on the immediately visible consequences of their choices. The regressions also show once more how large the immediacy bias is: In the perception of the average subject, increasing x_1 by 1 point is equivalent to increasing x_2 by 2.5. Put differently, subjects discount x_2 by a factor 0.4 over the short period of roughly 10 seconds between two rounds. For reasons laid out earlier, this discounting can not be attributed to temporal preferences, but must result from learning frictions.

1.3.3 Aggregate Choices: Non-Parametric Analysis

This section will analyze error frequencies, which offers a non-parametric, robust alternative to the preceding regressions. Choice sets are categorized according to the temporal profiles of both options. As mentioned before, choice sets are called *incongruent* if the better option has low immediate value ($x_1 < x_2$), while the worse option has high immediate value ($y_1 > y_2$), for example (6,10) and $(9, 5)$. In such choice sets, immediacy bias should lead to a particularly high error rate. The opposite case – such as $(2, 6)$ and $(7, 3)$ – will be called *congruent*: Here, placing

	LPM	Logit	Mixed Logit (uncorrelated)	Mixed Logit (correlated)
δ_1	0.0663***	0.355***	0.480***	0.482***
	(0.00263)	(0.0236)	(0.0324)	(0.0325)
δ_2	0.0265***	0.136***	0.181***	0.185***
	(0.00333)	(0.0184)	(0.0246)	(0.0249)
δ_1/δ_2	2.50	2.61	2.65	2.61
$\sigma(\delta_1)$			0.265***	0.268
			(0.0238)	
$\sigma(\delta_2)$			0.218***	0.221
			(0.0181)	
$\rho(\delta_1, \delta_2)$				0.264
Test of $H_0 : \delta_1 = \delta_2$				
F/χ^2	95.11	78.73	71.87	68.30
p-value	3.15e-16	7.14e-19	2.30e-17	1.41e-16
(pseudo) R^2	0.310	0.256		
log-likelihood	-3304.5	-3153.7	-2806.3	-2804.1
AIC	6614.9	6313.5	5620.7	5618.2
BIC	6635.1	6333.6	5650.3	5655.2
N	6119	6119	6119	6119

Standard errors in parentheses; cluster = participant. * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$
For the mixed models, coefficient entries represent E(δ).

Table 1.2: Main regression results. Dependent variable is choice probability of x; explanatory variables are $\hat{x}_1$ and $\hat{x}_2$, i.e. immediate and delayed feedback as previously seen by the subject.

more weight on x_1 should actually reduce errors, so that a low error frequency is expected. For all remaining choice sets – where the options both have low x_1 or high x_1 – the immediacy bias should have no direct effect, and they will be disregarded here.

Table 1.3 shows the contingencies of errors in cases of interest. Error rates of 47% in incongruent, and of only 15% in congruent choice sets again show how strong the bias is. A signed rank test on individual error rates in both settings confirms that this difference is highly significant ($p \ll 0.001$). This is further statistical evidence for hypothesis 1, and of 1a in particular.

1.3.4 Individual Heterogeneity

Since each subject contributes 60 observations, it is possible to perform the same regressions subject by subject. The left panel of Figure 1.4 plots the resulting co-

		Choice sets		
		Congruent	Incongruent	Combined
Correct	%	84.9	52.7	68.8
	#	1,559	968	2,527
Error	%	15.1	47.3	31.2
	#	277	868	1,145
Total	%	100.0	100.0	100.0
	#	1,836	1,836	3,672

Wilcoxon signed-rank test: $z = 7.09$, $p \approx 0.000$

Table 1.3: Contingency of errors in congruent choice sets (worse option $x_1 < x_2$, better option $y_1 > y_2$) and incongruent choice sets (worse option $x_1 > x_2$, better option $y_1 < y_2$).

efficients for the LPM, each point representing one subject.[5] Subjects who do not react at all to payoffs are around the origin. Subjects that react to both components equally are close to the 45°-line, the higher their precision the further from the origin. Subjects above the 45°-line react more strongly to immediate feedback. This applies to 84 of 102 subjects, clearly showing that the main hypothesis 1 holds not only for the average subject, but for most subjects individually. Their smooth distribution in the upper half of the quadrant indicates that the aggregate results were not driven by a small subset of subjects who might have misunderstood the task.

The right panel shows the same data, but rotated 45° counterclockwise. The result is simply a linear transformation of the former model, which allows convenient interpretation of coefficients. The estimated utility function is then $u(x) = \delta_{+}(\hat{x}_1 + \hat{x}_2) + \delta_{-}(\hat{x}_2 - \hat{x}_1)$. δ_{+} on the y-axis measures how subjects react to total value $\hat{x}_1 + \hat{x}_2$. This is the objective of the task, and a higher coefficient implies better performance. δ_{-} on the x-axis represents sensitivity to $\hat{x}_2 - \hat{x}_1$, i.e. to the split of total value into immediate and delayed component. To maximize points, this should be ignored: Ideally, this coefficient should be zero; both positive and negative values will cause errors. A majority of subjects however reacts negatively. δ_{-} is a suitable measure for the bias of individual subjects and will be related to other characteristics later. Note that there is only a moderate relation between δ_{+}

[5] The LPM is preferable for this purpose: For some subjects, logit coefficients cannot be obtained as complete separation is achieved.

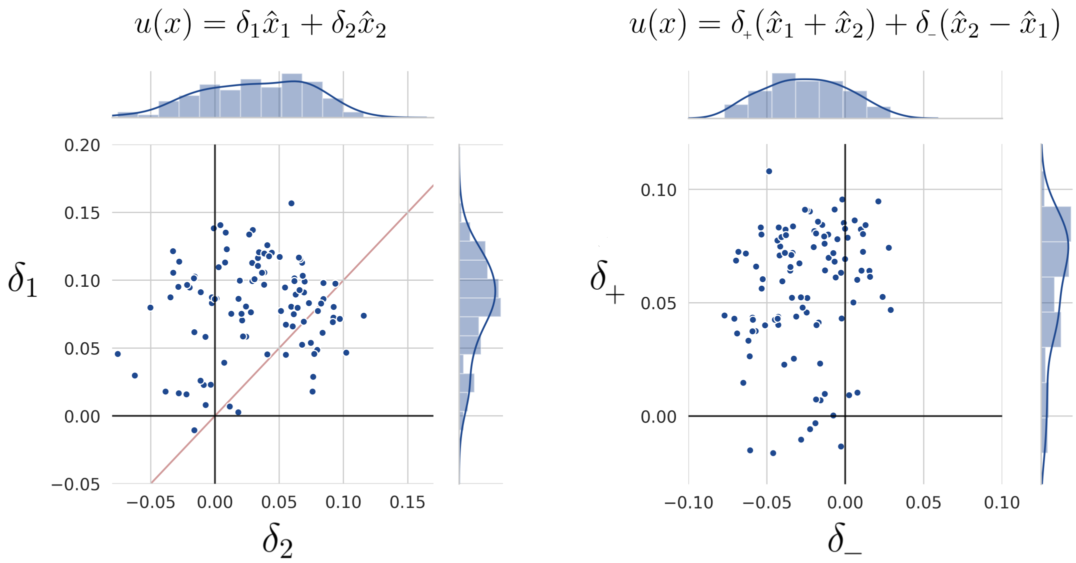


Figure 1.4: Per-subject measures of behavior. Left panel corresponds to LPM from Section 1.3.2. Right panel is rotated by 45°; see above for the implied latent utility function.

and δ_-; this underscores that subjects make plenty mistakes which are unrelated to temporal ordering of feedback.[6]

Mixed logit models are a viable way to get an accurate estimate of the underlying heterogeneity (Train, 2009). Results are reported in columns 3 and 4 of Table 1.2. These models are essentially a random coefficient extension of the regular logit model, with the assumption that coefficients are not constant in the population, but normally distributed: $\delta \sim N(d, W)$. The likelihood contribution of each subject is then

$$L = \int \prod_{t \in T} \frac{e^{\delta_1 x_1^t + \delta_2 x_2^t}}{e^{\delta_1 x_1^t + \delta_2 x_2^t} + e^{\delta_1 y_1^t + \delta_2 y_2^t}} \phi(\delta) d\delta$$

where x^t is the selected, and y^t the foregone option of observation t. The parameters d and W are estimated via maximum simulated likelihood. The correlated model in column 4 allows for full covariance matrix W; column 3 is nested by imposing diagonality; the standard logit model in column 2 is in turn nested with $W = 0$. The ratio of population means, ${}^{\mathrm{E}(\delta_1)}/_{\mathrm{E}(\delta_2)}$ closely resemble those in the fixed coefficient models. The estimated population standard deviations, $\sigma(\delta)$, are highly significant. In addition, the mixed models are clearly favored over the fixed

[6] In particular, a small fraction of subjects is clustered at the origin. Their choices appear completely random, perhaps they simply aimed to finish as fast as possible. As a result, these subjects are completely unbiased ($\delta_- \approx 0$), but also insensitive to total value ($\delta_+ \approx 0$).

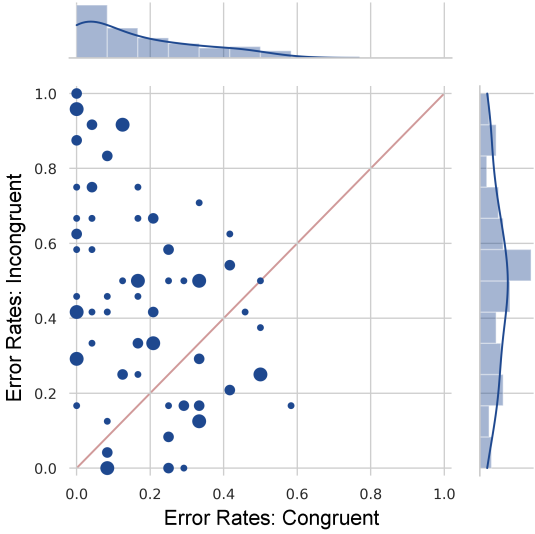


Figure 1.5: Subject heterogeneity: Error rates in incongruent and congruent choice sets. Dots denote 1/2/3 subjects by size.

coefficient models by information criteria (and likelihood ratio tests, not reported). Together, these results confirm that there is indeed considerable heterogeneity in immediacy bias in the sample, confirming hypothesis 3.

Figure 1.5 shows a non-parametric representation of individual heterogeneity. Error rates in congruent choice sets are plotted on the x-axis: Most subjects make no or very few mistakes in these rounds, and only a single subject more than 50%. For incongruent choice sets, the error rates vary much more, approximately centered around 50%. As noted before, any source of errors unrelated to temporal ordering is expected to cause either type of error with equal probability. The fact that almost all subjects fall above the 45°-line once more indicates the systematic nature of mistakes. The difference in error rates between incongruent and congruent choice sets can be used as a secondary, non-parametric measure of an individual's bias.

1.3.5 Beliefs

After completing the main task, subjects were asked which of the colors gave the most points, the second most points, and the least points (both components combined). The same color could not be named twice; the questions were not incentivized.

Figure 1.6 plots the answers by group. As the following will show, all patterns which an immediacy bias would predict are present in the data. Looking at group

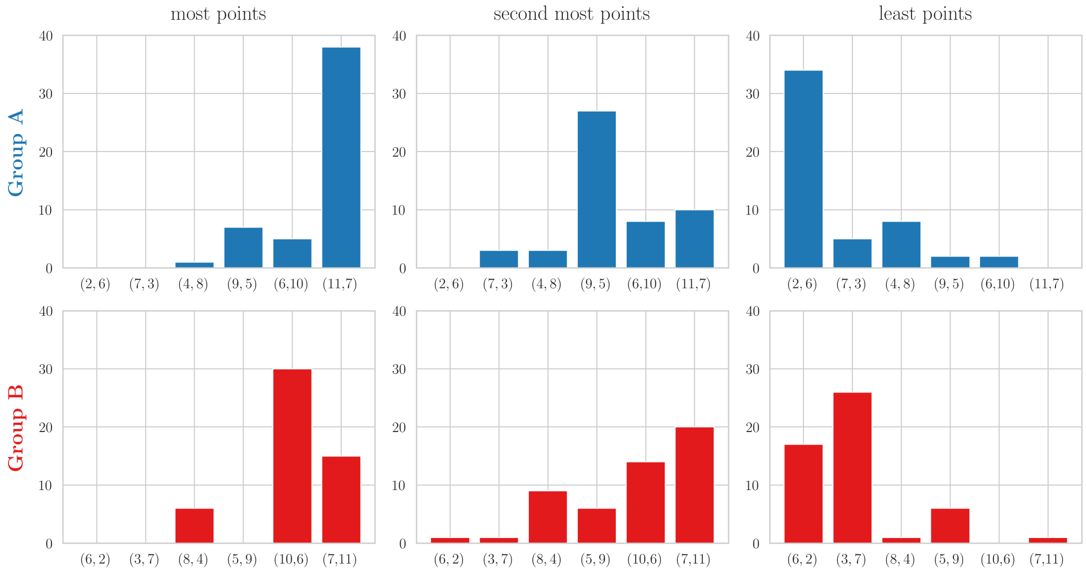


Figure 1.6: Answer counts to the question “Which color do you believe gave the most/second most/least points in total?”. 51 subjects per group.

A, the bias should actually help to identify (11,7) as best, and $(2, 6)$ as worst. And indeed, almost all subjects answered these correctly. In contrast, the question which option gave the second most points should be much harder for A, as the bias implies undervaluation of (6,10) and overvaluation of $(9, 5)$. A clear majority of subjects indeed incorrectly named the latter. As all options for B are reversed, the immediacy bias should make the questions for best and worst options much harder for this group. In fact, a majority of subjects incorrectly named (10,6) rather than (7,11) for most points. Likewise, a majority selected $(3, 7)$ and not $(6, 2)$ for least points.

The reversals in valuation implied by the bias even show in smaller details. Among the subjects who erred in the upper left panel, more named $(9, 5)$ than (6,10); the same holds for $(4, 8)$ over $(7, 3)$ in the upper right. In the lower left, $(8, 4)$ was named repeatedly, but $(5, 9)$ never. In the lower right, more subjects answered $(5, 9)$ than $(8, 4)$.

In summary, beliefs show the same bias as choice behavior: x_1 is given higher weight than x_2. This further corroborates the hypothesis that information frictions can lead to behavior and value judgments in which delayed consequences are discounted. The finding in particular shows that subjects actually believed to act in a manner that would maximize points. A perhaps unlikely, but theoretically possible explanation for biased choices would have been that subjects willingly accepted lower payoffs, because they were impatient to receive positive feedback early.[7] Belief data rules this out.

1.3.6 Bias and Intertemporal Choice

A specific feature of the experiment was that the bias in choices can not be explained by reward discounting. Nevertheless, bias and time preferences might be linked. If preferences are at least partly learned from experience, and this learning is – in some settings – subject to the immediacy bias, then a stronger bias should *ceteris paribus* result in less patient preferences. As shown before, the experiment allowed to identify considerable heterogeneity in degree of bias across subjects. If there is indeed a connection between bias and reward discounting, the following should hold:

Hypothesis 4: *The stronger an individual's bias in the main task, the more impatient this subject is in decisions involving actual reward discounting.*

To test this prediction, a set of hypothetical intertemporal choice data was elicited from all subjects after conclusion of the main task. This was done using a staircase estimator, a procedure that has been validated against incentivized decisions by Falk et al. (2016). In three series of questions, subjects were asked how they would decide between 100€ today or x in one month; 100€ today or x in six months; 100€ in one month or x in six months. By in- or decreasing x from question to question, effectively performing bisection search, indifference points in the range 100–132€ were obtained for each time horizon. By nature of the staircase procedure, more extreme values are censored. In the given sample, indifference points for the three time horizons were highly correlated within subjects. Analysis is reported for today versus in one month; using the other time horizons or an index yields similar results.

The indifference point as measure of impatience was regressed on individual bias, alongside some controls. Table 1.4 shows results; all coefficients are standardized. The first four columns use $\delta_{\_}$ as measure of bias. As the results show, the more biased an individual in the task ($\delta_{\_} \ll 0$) the higher their stated indifference points, and thus the impatience they expressed. This effect is significant and sizable; the results thus support hypothesis 4.

[7] Another argument also makes this explanation unlikely: It requires extreme assumptions on subjects' valuation of positive feedback and discount rates. After all, every single intentional mistake would have cost 0.10–0.50€, with the only upside of gaining a piece of positive feedback just the 10 seconds earlier a round approximately lasted.

	Dependent variable: Subject indifferent between 100€ today and y in 1 month							
δ_-	-0.283*** (0.004)	-0.272*** (0.007)	-0.225** (0.028)	-0.226** (0.032)				
Excess errors in incongruent choice sets					0.287*** (0.003)	0.278*** (0.005)	0.222** (0.030)	0.222** (0.036)
Total points		-0.051 (0.603)	-0.065 (0.526)	-0.056 (0.596)		-0.074 (0.450)	-0.081 (0.420)	-0.071 (0.495)
Study majors			✓	✓			✓	✓
Age & gender				✓				✓
N	102	102	102	99	102	102	102	99

Standardized beta coefficients; p-values in parentheses. ** $p < 0.05$, *** $p < 0.01$.
$\delta_-$is the coefficient on $\hat{x}_2 - \hat{x}_1$; excess errors is error rate in incongruent minus congruent choice sets, see section 1.3.4.
Study majors are categorized as economics/finance/business administration, MINT, law, social sciences, others.

Table 1.4: Regressions of a measure of impatience on immediacy bias and a set of controls.

The relation is unchanged when introducing controls, which notably also carry much less explanatory power than the measure of bias. The first of these is total points. As discussed earlier, temporal bias is only one of many potential sources of errors; in consequence, total points can be used as proxy for general ability in the task. The coefficients are small and insignificant, indicating that ability carries no relation to measured time preferences. Field of study shows some relation to the degree of immediacy bias in the data, yet its inclusion reduces the coefficient only slightly; age and gender are without effect. The remaining columns repeat the same estimation, but using a non-parametric measure of bias instead, namely error rate in incongruent minus error rate in congruent choice sets. Results are unchanged.

1.3.7 Decision Times

To conclude discussion of the main experiment, I briefly turn to decision times, where the immediacy bias also manifests. To recapitulate Section 1.2.4, timing during the task was as follows. Right after each choice, feedback for the two previously chosen colors was revealed. 2 seconds after the feedback, two new options appeared on screen. Subjects now had up to 10 seconds to make a choice, and also to still consider the previous feedback, which remained on screen until

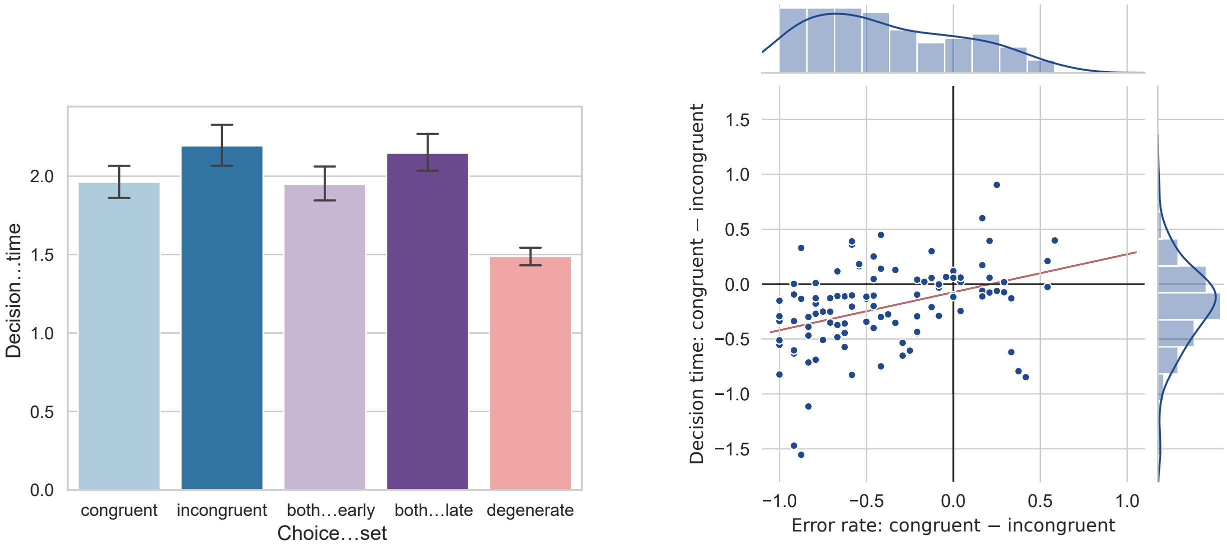


Figure 1.7: Left: Average decision times in different types of choice sets. Error bars show 95% confidence intervals. Right: Difference in error rates between incongruent and congruent choice sets against difference in average decision times, per subject. Red line represents regression fit.

the choice was made. In the following, *decision time* refers to the time a subject took from the appearance of new options until making a choice.

The average decision time in blocks 2–5 was slightly less than 2 seconds; among over 13 000 observations, only a single timeout occurred. The left panel of Figure 1.7 shows average decision times for different types of choice sets. First, the average time of 1.49s in *degenerate* choice sets (the same color twice) is essentially a baseline, indicating how long subjects took when no decision was to be made at all. Next, contrast *congruent* and *incongruent* choice sets, with averages of 1.96s and 2.19s. This difference of .23s is highly significant ($t = -4.62$, clustered per participant). It is a general observation that decisions with smaller perceived value difference are taken more slowly (Shevlin et al., 2022). While congruent and incongruent choice sets are perfectly symmetric in terms of objective value (see Table 1.1), under an immediacy bias subjective value difference is much smaller in incongruent choice sets. Thus, the bias is also apparent in decision times.

This also shows when considering choice sets where either *both early* components are high, or *both late* are high, with decision times of 1.95s and 2.15s. The difference is again highly significant ($t = 4.47$). Here too, objective value is symmetric; the bias induces higher valuation of both options in *both early* and lower in *both late*. Decision times are therefore in line with the finding that when holding value difference constant, higher average value is associated with faster decisions (Shevlin et al., 2022).[8]

Between subjects, the bias in timing is correlated with the bias in choice ($\rho = 0.33$, $p < 0.001$). The former is measured as difference in decision times between congruent and incongruent choice sets; the latter as difference in error rate. The right panel of Figure 1.7 shows data and a regression fit.

1.4 Active and Passive Learning

In the main condition of the experiment, subjects learn from actual reinforcement: All displayed numbers result from a choice made by the subject and are payoff-relevant. Thus, they are not purely neutral information that could help future decisions, but they also carry judgment of own performance, and signify a reward to be received later. It seems plausible that in such a setting the immediacy bias will be particularly strong: The immediate feedback is displayed moments after the decision, when the subject will be anxious to know whether it was a good choice. By the time the delayed feedback comes around, another decision has been made on which the mind is now focused, so that this part of feedback might arouse much less interest.

To address the question whether reducing these elements tied to active decision making reduces also the bias, a variation of the experiment allowed subjects to learn more passively, without such involvement. The new set of subjects first learned simply observing, without making any choices and without payoff implications. Afterwards they would face own incentivized decisions. The hypothesis is that this encourages a less visceral, more detached and analytical processing of the given information. This should reduce the asymmetry between both components and lead to a reduction in bias.

1.4.1 Passive Treatment: Design

The passive treatment kept all rules and mechanics of the main experiment, and implemented only one change: In the first 63 rounds, subjects did not make any decisions, but could learn passively from feedback shown on their screen.

[8] The bias makes no direct prediction in terms of error rates between these choice sets; however, the error rate is higher in *both late* sets, with 21.1% vs 18%. This difference is however only marginally significant with $p = 0.09$ in a clustered LPM. Note that in the current design, a direct comparison of these choice sets with the (in-)congruent ones is not sensible, as the objective value differences have other magnitudes, see Table 1.1.

Specifically, at the beginning of each round, a single button appeared in the center of region (a) in Figure 1.1, moved to (b) a few seconds later, and one round later to (c), both times displaying immediate respectively delayed feedback. Starting from round 64, 42 normal rounds of decisions followed seamlessly, including feedback as usual. These decisions carried the same marginal incentive of 0.05€ per point. Instructions were unchanged, with an added section that explained the learning phase.

To allow comparison with the main treatment, it was important that subjects would see comparable information up to round 64, when decisions in the passive treatment started. This was achieved as follows. Each treatment subject was partnered at random to one of the main subjects from earlier sessions. During the learning phase, the sequence of colors displayed to the new subject were exactly the colors chosen by the other. Values of all colors were the same, and the feedback exactly replicated what the former subject had seen. Timing of the rounds followed the pace the partner had set. In summary, it was almost as if the second subject got to watch over the shoulder of the first during these initial rounds. However, subjects were not explicitly informed how the data had been generated. Moreover, the color the previous subject had foregone was not displayed. This was both to limit similarity to a choice situation, and not to convey information on the first subject's beliefs to the second. This way, partner and new subject entered the final 42 rounds with exactly the same information. The sequence of choice sets in these rounds was again kept identical. The procedure resembles a yoked control design common in studies of operant conditioning. Within the economic literature, Merlo and Schotter (2003) use a similar setup to specifically examine differences in observational and active learning.

If active learning is indeed conducive to the immediacy bias, this should show as follows:

Hypothesis 5: *Subjects in the passive treatment show less bias than subjects in the main condition, when comparing the final 42 rounds. In regressions, their coefficient ratio δ_1/δ_2 is lower. They show less excess errors in incongruent choice sets.*

1.4.2 Passive Treatment: Results

All results in this section are based on 57 subjects in the passive learning condition, whose behavior is compared to the 102 subjects from the main condition. For each new subject, a partner was drawn from the main subjects at random without

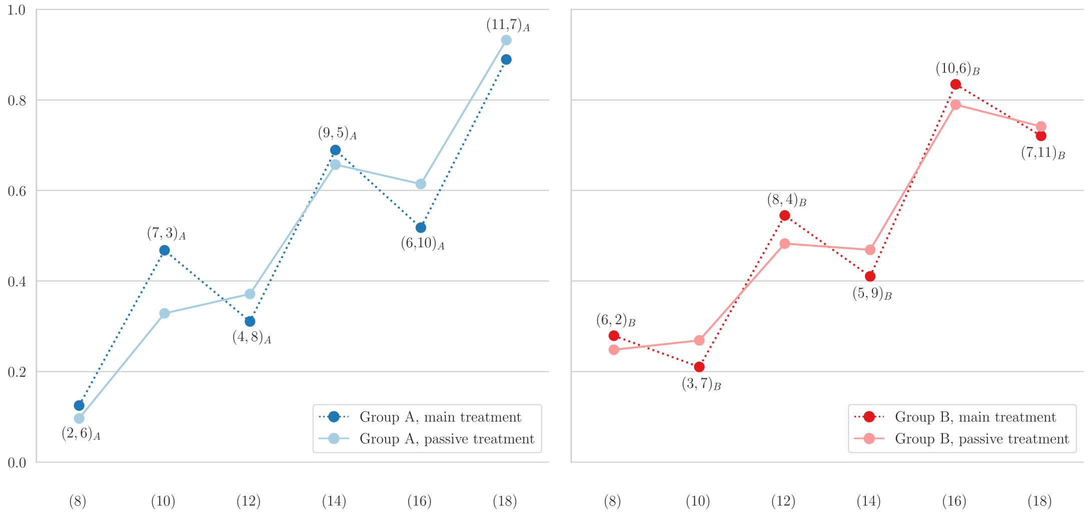


Figure 1.8: Comparison of choice frequencies in main and passive condition.

replacement. All results are robust to omitting the un-partnered subjects from the main condition. Data for each subject consists of the 30 non-degenerate decisions made in the final 42 rounds.

Figure 1.8 plots choice frequencies by group and treatment. For the main condition, the graph is familiar, with a zigzag-pattern and reversals reflecting the immediacy bias. For treatment subjects, the pattern is similar, but visibly attenuated: relative to the main subjects, the graphs are less jagged and closer to a straight line. In comparison, passive subjects' choice frequencies are lower for options with $x_1 > x_2$, and higher for options with $x_1 < x_2$. Both are consistent with a decreased bias. The only exceptions are $(2,6)_A$ (decrease in frequency) and $(11{,}7)_A$ (increase). However, these cases are consistent with a general improvement offsetting a decrease in bias. For the treated group, reversals are attenuated or even absent, e.g. between $(7,3)_A$ and $(4,8)_A$. In summary, the visual evidence suggests that subjects in the passive treatment still show immediacy bias, but to a lesser degree. The following corroborates this both by regressions and non-parametric analysis.

The latent utility function estimated in the treatment regressions is

$$u(x) = (\delta_1 + \mathbb{1}_T\gamma_1)\,\hat{x}_1 + (\delta_2 + \mathbb{1}_T\gamma_2)\,\hat{x}_2$$

where δ and $\hat{x}$ are as before. $\mathbb{1}_T$ is an indicator for the passive treatment group, so that γ_1 and γ_2 capture the differential reaction of treated subjects to immediate and delayed feedback. Table 1.5 summarizes results. δ_1 and δ_2 are almost un-

	Mixed logit uncorrelated	Mixed logit correlated	LPM	Logit	LPM (subsample)
δ_1	0.627^{***}	0.635^{***}	0.0712^{***}	0.403^{***}	0.0731^{***}
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
δ_2	0.229^{***}	0.243^{***}	0.0291^{***}	0.158^{***}	0.0259^{***}
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
γ_1	-0.00654	-0.00236	-0.00246	-0.00586	-0.00410
	(0.927)	(0.976)	(0.635)	(0.916)	(0.493)
γ_2	0.116^{**}	0.122^{**}	0.0115^{*}	0.0672^{*}	0.0146^{**}
	(0.036)	(0.045)	(0.060)	(0.097)	(0.032)
$^{\delta_1}/_{\delta_2}$	2.73	2.61	2.45	2.55	2.82
$^{\delta_1}/_{(\delta_2+\gamma_2)}$	1.80	1.74	1.75	1.81	1.80
N	4769	4769	4769	4769	3420
log-likelihood	-1950.5	-1945.3	-2380.4	-2284.0	–
AIC	3913.1	3904.6	4770.9	4578.0	–
BIC	3956.1	3954.7	4803.2	4610.4	–

p-values in parentheses. Cluster = participant. * $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$
Mixed models: δ are random coefficients; E(δ) reported, Var(δ) omitted for brevity.

Table 1.5: Regression results, comparing the reaction of subjects to x_1 and x_2 in main (δ only) and passive conditions ($\delta + \gamma$).

changed from the main results; this is expected, as these coefficients describe the same subjects (albeit now in a subset of rounds). In terms of treatment effects, γ_1 is a clear null result. γ_2 on the other hand amounts to roughly half of δ_2. This is significant at 5% in the preferred mixed logit models (and at least marginally so in the others). Subjects in the treatment show a similar reaction to x_1, but a much stronger reaction to x_2 in comparison to the main subjects. The bias is clearly attenuated in the treatment, in accordance with hypothesis 5. The ratio of coefficients shrinks from around 2.6 for main subjects to around 1.8 for treatment subjects. The mixed models again clearly outperform the others. As shown in the last column, results are also significant when using only the 57 matched subjects from the main treatment as comparison group.

Error frequencies reported in Table 1.6 show similar results. In congruent choice sets, treated subjects commit 1.6 percentage points more errors. This is in accordance with an attenuation of bias, even though this difference is insignificant. In incongruent choice sets, the error rate is 8 percentage points lower, and the

		Incongruent choice sets			**Congruent choice sets**		
		Main	Passive	Total	Main	Passive	Total
Correct	%	53.5	61.6	56.4	87.5	85.9	86.9
	#	491	318	809	803	438	1,241
Error	%	46.5	38.4	43.6	12.5	14.1	13.1
	#	427	198	625	115	72	187
Total	%	100.0	100.0	100.0	100.0	100.0	100.0
	#	918	516	1,434	918	510	1,428

Wilcoxon rank-sum test: $z = 1.97$, $p = 0.049$ $z = -0.36$, $p = 0.72$

Table 1.6: Comparison of main and passive condition: Error frequencies in incongruent and congruent choice sets.

according rank-sum test on individual error frequencies turns out significant. This again supports hypothesis 5.

As the results show, the immediacy bias is significantly attenuated for treatment subjects, but it is still clearly present. One interpretation of these findings is that the active setting is conducive, but not necessary for its occurrence. There is also an alternative explanation: Subjects in the passive condition might start their active choices with little or no bias, but then quickly develop it. After all, from round 64 onward they do get active feedback; and in the main condition, the bias was already discernible in the first block of choices (see Figure 1.3). Contrasting treatment subjects' earliest choices with their last offers some support for the latter alternative; but the data is unfortunately not conclusive. This is discussed in Appendix 1.C.

1.5 A Bayesian Framework

As mentioned in Section 1.1.1, models based on signal detection can offer an explanation why delayed feedback is discounted. These go back to Commons et al. (1991) and have recently been taken up by Gabaix and Laibson (2017) in the context of time preferences. This section briefly sketches an application to the experiment. As will be seen, noisy memory offers a parsimonious explanation of the immediacy bias, which avoids to presume any biased beliefs or any misunderstanding of the environment on behalf of the subjects.

The core assumption is that memory is unbiased, but imprecise: An agent trying to remember some quantity z will retrieve not this number exactly, but

$z + \zeta$ instead, where ζ is a mean-zero random disturbance. Bayesian decision theory dictates that when an agent bases decisions on a noisy signal like $z + \zeta$, it should be weighted according to its variance. One should think of this correction not necessarily as conscious; often, it may be automatic and hard-wired into the perception and decision processes (see e.g. Khaw et al., 2017).

Consider now a typical choice situation in the experiment. The decision maker has to choose between options x and y. She does not exactly remember the values they have generated previously, but has to rely on her fallible memory, represented by the following vector s:

$$s = \begin{bmatrix} s_{x1} \\ s_{x2} \\ s_{y1} \\ s_{y2} \end{bmatrix} = \begin{bmatrix} \hat{x}_1 + \zeta_{x1} \\ \hat{x}_2 + \zeta_{x2} \\ \hat{y}_1 + \zeta_{y1} \\ \hat{y}_2 + \zeta_{y2} \end{bmatrix}$$

As before, $\hat{x}_1, \ldots, \hat{y}_2$ stands for a summary of the actual evidence as it was visible on screen. ζ are i.i.d. noise terms with mean zero, reflecting imperfect, but unbiased memory. To ease exposition, I will assume normal distributions, but nothing rests on this. Since x and y are interchangeable, it is a natural assumption that the standard deviations of ζ are symmetric, i.e. $\sigma_{x1} = \sigma_{y1} = \sigma_1$ and $\sigma_{x2} = \sigma_{y2} = \sigma_2$.

To proceed with Bayesian updating, a set of priors needs to be specified. I will assume that the decision maker has the same prior distribution $N(p, \sigma_p)$ for all four components, avoiding any underhand introduction of bias. Normality is again just for tractability. As will be shown, the prior mean p is actually irrelevant for the result; one may for example assume that it is the true mean for all components across all options.

Standard Bayesian updating gives posterior distributions, again normal. Their means are a convex combination of prior mean and the noisy signal retrieved from memory:

$$\mathrm{E}(\hat{x}_1|s_{x1}) = \frac{\sigma_p^2}{\sigma_1^2 + \sigma_p^2} s_{x1} + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_p^2} p \qquad \mathrm{E}(\hat{x}_2|s_{x2}) = \frac{\sigma_p^2}{\sigma_2^2 + \sigma_p^2} s_{x2} + \frac{\sigma_2^2}{\sigma_2^2 + \sigma_p^2} p$$

Introducing $\delta_i = \sigma_p^2(\sigma_i^2 + \sigma_p^2)^{-1}$ and decomposing s, posterior expected value of option x can be written

$$\mathrm{E}(\hat{x}_1 + \hat{x}_2|s) = \delta_1 \hat{x}_1 + \delta_1 \zeta_{x1} + \delta_2 \hat{x}_2 + \delta_2 \zeta_{x2} + (2 - \delta_1 - \delta_2) p$$

To maximize expected value, the decision maker should choose x over y iff

$$\mathrm{E}(\hat{x}_1 + \hat{x}_2 - \hat{y}_1 - \hat{y}_2 | s) = \delta_1 \hat{x}_1 + \delta_2 \hat{x}_2 - \delta_1 \hat{y}_1 - \delta_2 \hat{y}_2 + Z > 0$$

where Z is a weighted sum of ζ and thus itself a mean-zero normal random variable. Due to symmetry between x and y, the prior means have canceled out. This equation corresponds exactly to a probit version of the choice models estimated earlier. Moreover, one has $\delta_1 > \delta_2$ if and only if $\sigma_1 < \sigma_2$.

Thus, a model of noisy memory and Bayesian updating can explain the discounting of delayed feedback under the simple assumption that immediate feedback is remembered with higher accuracy, i.e. $\sigma_1 < \sigma_2$. Arguably, this is not an unreasonable assumption. First, noise may simply be increasing due to the passing of time or the presence of intermittent decisions (Commons et al., 1991). In many situations, immediate consequences are easiest to attribute, because there is less ambiguity to which action they have to be attributed. Arguably, when interacting with the physical surroundings, immediate consequences are also most common and often most important. Perception and cognition may be organized such that an over-proportionate share of resources is devoted to tracking them. Finally, there is evidence that changing the delay of feedback from one to only a few seconds can substantially change the neurophysiological pathways in which it is processed (Foerde and Shohamy, 2011; Foerde et al., 2013).

1.6 Conclusion

This paper advocated the view that myopic behavior may arise from frictions in learning: If reinforcement loses efficacy with temporal delay, immediate consequences will receive undue weight relative to delayed ones. If one takes preferences as learned, this has important ramifications for our understanding of time discounting.

As its main contribution, the paper is first to provide clean evidence for the occurrence of such “as if”-discounting in a controlled experiment with human participants. Subjects learn from immediate and delayed value signals, and discount the latter to striking extent. The design allows to rule out temporal preferences as an explanation, so that the observed behavior must be attributed to learning frictions. Corroborating this view, subjects’ elicited beliefs mirrored exactly the bias observed in behavior.

The experiment further showed that this bias may be stable even when facing a continuous stream of contrary evidence. A majority of subjects is affected; the experiment allows to measure the degree of bias per individual, revealing considerable heterogeneity. Consistent with the perspective that preferences in actual intertemporal choice are partly shaped by such biases, it was shown that subjects with higher degree of bias indeed gave more impatient answers in a classical, albeit hypothetical measure of time preferences.

Turning to the question which factors are important for the occurrence of the discussed bias, a treatment variation was introduced in which subjects could learn by observation, before then making active decisions. The leading hypothesis was that a passive setting would mitigate the bias. Here, results were mixed: Subjects in the treatment group showed less, but still considerable bias. This is compatible with the active setting either being conducive, but not necessary, or with subjects quickly picking up a bias during their active decisions.

Finally, it was shown that the bias could be explained by a model of Bayesian updating from noisy memory, based on the simple assumption that delayed feedback is processed or remembered with less precision.

Together, the results support the view that time preferences are shaped by learning processes and their limitations. This offers new perspectives on time preferences as they are studied in behavioral economics. At the same time, it seems worthwhile to consider potential avenues for interventions that target impatient behaviors. After all, if an adverse environment can foster the development of impatience, then designing the right environment for learning may be a good way to teach patience.

Appendix

1.A Simulation: Greedy Algorithm

Figure 1.9 shows choice data obtained from a simple greedy algorithm in the experimental task. The algorithm is programmed as follows: For each color, it stores a continually updated average over all realizations. In the first 21 rounds, it chooses a random option. Afterwards, it always chooses the option with higher current average (ties are resolved randomly).

This strategy requires to store only 6 values and 6 counts, and performs no inference of any sort. Still, in rounds 22–105 it achieves an error rate of only 0.3%. Clearly, no explorative strategy could improve on that. The algorithm's performance illustrates that memorization is the only difficulty present in the given task.

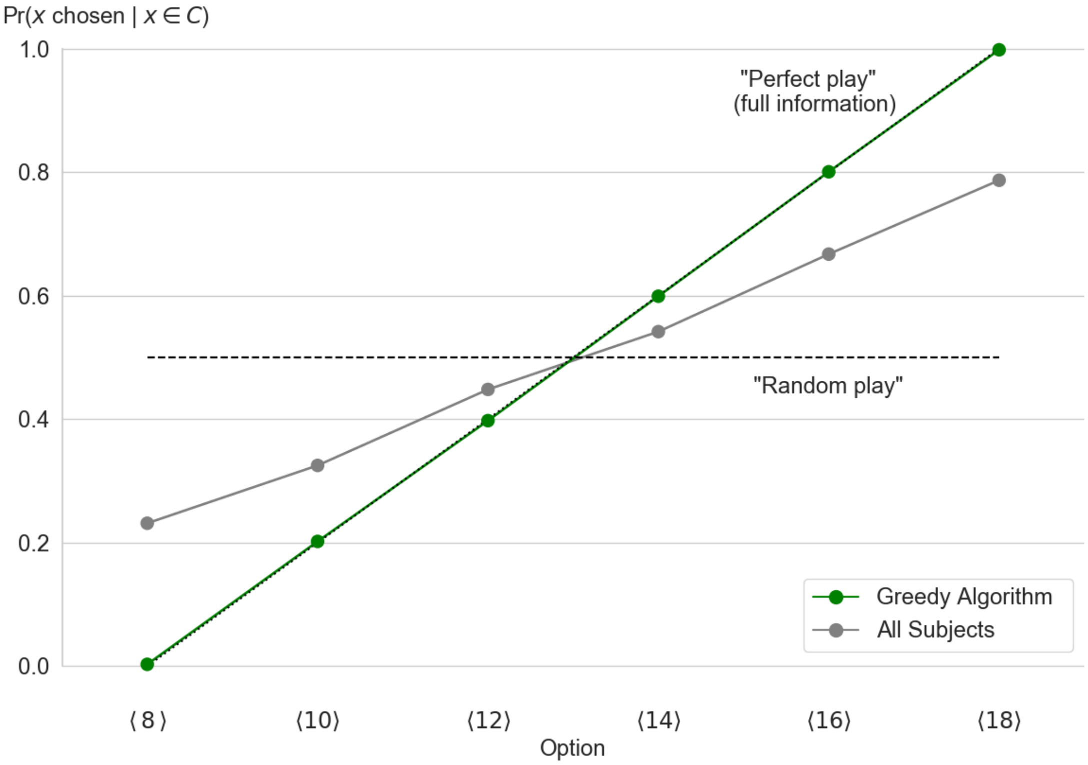


Figure 1.9: Simulation results; the black graph represents a greedy algorithm, perfectly tracking the mean of past observations; it coincides with the diagonal indicating perfect play. Grey is observed human behavior in the same rounds.

1.B Specification of $\hat{x}$

Table 1.7 shows the linear probability model, comparing candidate 3 specifications how all observations of $x_i + \epsilon$ a subject has up to a specific choice situation may be aggregated to $h\hat{x}_i$. These are average over all realizations, as used in the text, the most recent observation, and the average over only the first 21 observations. The first seems to outperform both others. In particular, when combined with the most recent observation (fourth column), the result suggests the latter plays no significant role of its own. This suggests the absence of any recency effect. When paired with the early aggregate, colinearity becomes an issue, and standard errors go up.

Linear probability models: Pr(x chosen$\|C_t = \{x, y\}$)					
δ_1 – mean	0.0663***			0.0674***	0.0462**
	(0.00263)			(0.00506)	(0.0151)
δ_2 – mean	0.0265***			0.0243***	0.0207
	(0.00333)			(0.00550)	(0.0127)
δ_1 – last		0.0580***		-0.00109	
		(0.00233)		(0.00394)	
δ_2 – last		0.0226***		0.00211	
		(0.00299)		(0.00418)	
δ_1 – early			0.0656***		0.0204
			(0.00265)		(0.0152)
δ_2 – early			0.0257***		0.00582
			(0.00326)		(0.0124)
Constant	0.510***	0.510***	0.510***	0.510***	0.510***
	(0.00720)	(0.00705)	(0.00717)	(0.00717)	(0.00719)
Observations	6119	6119	6119	6119	6119

Standard errors in parentheses; cluster = participant.

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table 1.7: Comparison different specifications of $\hat{x}_i$ – mean is the average over all previous realizations, last the last seen, and early is the average only from rounds rounds 1–21.

1.C Passive Treatment: Early and Late Bias

As discussed in Section 1.4, subjects in the passive condition appear biased, but to a lesser degree than those in the main condition. One possible explanation is that subjects end the passive learning phase unbiased, or almost unbiased, but then quickly adapt a bias as they start making own decisions. If this is true, the earliest active decisions of these subjects should show less bias than those following later.

The data has some features that are consistent with this interpretation. Figure 1.10 plots the first 6 active choices of these subjects on the left, and their final 6 on the right. It is apparent that the graphs on the left are much closer to a straight line, which would indicate unbiasedness, while the final choices show the typical serrated pattern. It is also obvious that these graphs are based on much noisier data than any shown earlier. This is no coincidence, as each graph now represents only 28 × 6 choices. Accordingly, with the given data, the difference is not statistically significant.

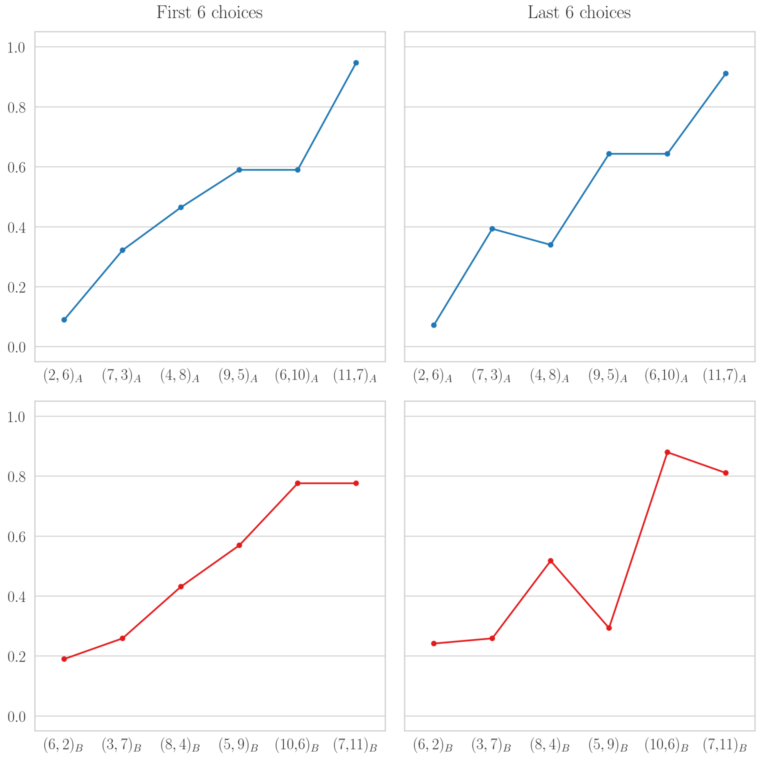


Figure 1.10: Choice behavior right at the beginning (left) and at the end of the decision phase for subjects in the passive treatment.

References

Barron, G., S. Leider, and J. Stack (2008): “The Effect of Safe Experience on a Warnings’ Impact: Sex, Drugs, and Rock-n-Roll,” *Organizational Behavior and Human Decision Processes*, 106, 125–142.

Chen, D. L., M. Schonger, and C. Wickens (2016): “oTree. An open-source platform for laboratory, online, and field experiments,” *Journal of Behavioral and Experimental Finance*, 9, 88–97.

Commons, M. L., M. Woodford, and J. R. Ducheny (1982): “How reinforcers are aggregated in reinforcement-density discrimination and preference experiments,” in *Quantitative Analyses of Behavior 2: Matching and Maximizing Accounts*, ed. by M. L. Commons, R. J. Herrnstein, and H. Rachlin, Cambridge, M.A.: Ballinger, 25–78.

Commons, M. L., M. Woodford, and E. J. Trudeau (1991): “How each reinforcer contributes to value: "Noise" must reduce reinforcer value hyperbolically,” in *Signal Detection. Mechanisms, Modelsm and Applications*, ed. by M. L. Commons, J. A. Nevin, and M. C. Davison, Hillsdale: Lawrence Erlbaum, 139–168.

Dai, J., T. Pachur, T. J. Pleskac, and R. Hertwig (2019): “What the Future Holds and When: A Description-Experience Gap in Intertemporal Choice,” *Psychological Science*, 30, 1218–1233.

Erev, I. and E. Haruvy (2015): “Learning and the Economics of Small Decisions,” in *The Handbook of Experimental Economics: Volume 2*, ed. by J. H. Kagel and A. E. Roth, Princeton and Oxford: Princeton University Press, chap. 10, 638–716.

Erev, I. and A. E. Roth (2014): “Maximization, learning, and economic behavior,” *Proceedings of the National Academy of Sciences*, 111, 10818–10825.

Ericson, K. M. and D. Laibson (2019): “Intertemporal Choice,” in *Handbook of Behavioral Economics – Foundations and Applications 2*, ed. by B. D, L. D, and D. S, North Holland: Elsevier.

Falk, A., A. Becker, T. Dohmen, D. Huffman, and U. Sunde (2016): “The Preference Survey Module: A Validated Instrument for Measuring Risk, Time, and Social Preferences,” *IZA Discussion Paper*, 9674, 605–611.

Foerde, K., E. Race, M. Verfaellie, and D. Shohamy (2013): "A Role for the Medial Temporal Lobe in Feedback-Driven Learning: Evidence from Amnesia," *Journal of Neuroscience*, 33, 5698–5704.

Foerde, K. and D. Shohamy (2011): "Feedback Timing Modulates Brain Systems for Learning in Humans," *Journal of Neuroscience*, 31, 13157–13167.

Gabaix, X. and D. Laibson (2017): "Myopia and Discounting," *Working Paper.*

Greiner, B. (2015): "Subject pool recruitment procedures: organizing experiments with ORSEE," *Journal of the Economic Science Association*, 1, 114–125.

Herrnstein, R. J. (1997): *The Matching Law: Papers in Psychology and Economics*, Cambridge, MA: Harvard University Press.

Hertwig, R. and I. Erev (2009): "The description–experience gap in risky choice," *Trends in Cognitive Sciences*, 13, 517–523.

Khaw, M. W., Z. Li, and M. Woodford (2017): "Risk Aversion as a Perceptual Bias," Working Paper 23294, National Bureau of Economic Research.

Lattal, K. A. (2010): "Delayed reinforcement of operant behavior," *Journal of the Experimental Analysis of Behavior*, 93, 129–139.

Malmendier, U. and S. Nagel (2011): "Depression Babies: Do Macroeconomic Experiences Affect Risk Taking?" *The Quarterly Journal of Economics*, 126, 373–416.

Merlo, A. and A. Schotter (2003): "Learning by not Doing: An Experimental Investigation of Observational Learning," *Games and Economic Behavior*, 42, 116–136.

O'Donoghue, T. and M. Rabin (2015): "Present Bias: Lessons Learned and to Be Learned," *American Economic Review*, 105, 273–279.

Prelec, D. (2014): "Consuming at the Wrong Rate: Lessons from the Harvard Game," in *Sustainable Consumption. Multi-Disciplinary Perspectives in Honour of Professor Sir Partha Dasgupta*, ed. by D. Southerton and A. Ulph, Oxford: Oxford University Press, chap. 7, 161–174.

SHEVLIN, B. R., S. M. SMITH, J. HAUSFELD, AND I. KRAJBICH (2022): "High-value decisions are fast and accurate, inconsistent with diminishing value sensitivity," *Proceedings of the National Academy of Sciences*, 119, e2101508119.

SIMS, C. R., H. NETH, R. A. JACOBS, AND W. D. GRAY (2013): "Melioration as Rational Choice: Sequential Decision Making in Uncertain Environments," *Psychological Review*, 12, 139–154.

THORNDIKE, E. L. (1911): *Animal Intelligence. Experimental Studies*, New York: Macmillan.

TRAIN, K. E. (2009): *Discrete Choice Methods with Simulation. 2nd Ed.*, Cambridge University Press.

Chapter 2

Markov Quantal Response Equilibrium: Existence, Computation, and Characterization

This chapter is based on joint work with Steffen Eibelshäuser.

Abstract: We introduce and prove existence of Markov quantal response equilibrium (QRE), an application of QRE to finite discounted stochastic games. We then study a specific case, logit Markov QRE, which arises when players react to total discounted payoffs using the logit choice rule with precision parameter λ. We show that the set of logit Markov QRE always contains a smooth path that leads from the unique QRE at $\lambda = 0$ to a stationary equilibrium of the game as $\lambda \to \infty$. Following this path allows to solve arbitrary finite discounted stochastic games numerically; an implementation of this algorithm is publicly available as part of the package sgamesolver. We further show that all logit Markov QRE are ε-equilibria, with a bound for ε that is independent of the payoff function of the game and decreases hyperbolically in λ. Finally, we establish a link to reinforcement learning, by characterizing logit Markov QRE as the stationary points of a game dynamic that arises when all players follow the well-established reinforcement learning algorithm expected SARSA.

2.1 Introduction

Economic environments are typically not stable, but highly dynamic. Current choices carry not only immediate consequences, but also shape the options available in the future. Examples include pricing, the accumulation or depletion of resources, savings or capacities, as well as entering legal obligations through contracts. Such intertemporal trade-offs are clearly reflected in partial and general equilibrium analysis, where dynamic programming (Bellman, 1954) is ubiquitous. However, if one looks at the analysis of strategic interaction, the picture changes. The most prominent models here are either one-shot games, or at best repeated games – which incorporate some dynamics between the players, but assume an essentially state-less world in which the only lasting consequences of actions stem from the reactions of others. This limitation is not due to a lack of theoretical concepts. Dynamic interaction among forward-looking economic agents can be modeled as a stochastic game, a broad class of games that dates back to Shapley (1953) and generalizes both repeated games (by introducing states) and Markov decision processes (by introducing strategic interaction).

But stochastic games are typically difficult to solve. Analytical solution is generally not feasible. This is true for dynamic programming problems as well – but here, powerful numerical methods are available. Unfortunately, these methods are not readily transferable to stochastic games: They are iterative in nature, and typically do not converge when strategic interaction is present. When multiple players interact in a dynamic environment, the corresponding Bellman operator is generally not a contraction – in contrast to single-player case. Therefore, developing well-suited numerical methods is a crucial step to enable applied economists to analyze strategic interaction in general dynamic environments.

The most common solution concept for stochastic games is stationary equilibrium (Shapley, 1953), a refinement of subgame perfect Nash equilibrium in which strategies are conditional on the current state of the game, but otherwise independent of past play. The most famous algorithms to compute stationary equilibria are due to Pakes and McGuire (1994, 2001). The algorithms are based on value function iteration, but are not guaranteed to converge. In addition, they only allow to find stationary equilibria in pure strategies – which in many games do not exist. Up to now, to the best of our knowledge, the only algorithms able to compute stationary equilibria in *mixed* strategies are based on homotopy continuation (Herings and Peeters, 2004; Govindan and Wilson, 2009; Dang et al., 2022, Chapter 3). However, existing homotopy methods often involve drawbacks. Take for example the linear stochastic tracing procedure by Herings and Peeters (2004), which resembles the famous linear tracing procedure for normal form games (Harsanyi, 1975; Harsanyi and Selten, 1988). It starts at arbitrary prior beliefs about other players' strategies and gradually transforms beliefs until equilibrium beliefs are obtained. However, convergence is only guaranteed for generic games. Convergence fails if, at some intermediate belief along the homotopy path, the set of Nash equilibria is at least two-dimensional, which may very well happen in applications.[1]

In this paper, we introduce (logit) Markov quantal response equilibrium, an approximate solution concept for stochastic games. As we show, it offers a new homotopy method to compute stationary equilibria that is guaranteed to converge for *all* finite discounted stochastic games. As a by-product, it can also be used as a selection criterion for stationary equilibria. The method is based on the logit quantal response framework (McKelvey and Palfrey, 1995, 1998). As a foundation, we provide a formal extension of quantal response equilibrium (QRE) to the domain of stochastic games, substantiating Breitmoser et al. (2010), and prove existence, finiteness and a limiting relation to stationary equilibria. Furthermore, we generalize the homotopy interpretation of QRE proposed by Turocy (2005, 2010) for normal-form and extensive-form games to the domain of stochastic games.

We discuss two further specific properties of logit Markov QRE that, to the best of our knowledge, have not been established even for QRE in normal-form games yet. First, we show that logit Markov QRE are ε-equilibria and establish a bound for ε that, interestingly, is independent of the payoffs of the game and decreases in the precision parameter λ of the model. Thus, while players do incur some loss by adopting the logit choice rule rather than maximizing perfectly, this loss can be bounded arbitrarily by choosing λ accordingly. Moreover, the result shows that this is possible even before having knowledge of the specific payout structure of the game.

The second property is a link of logit Markov QRE to reinforcement learning. Reinforcement learning is a sub-discipline of machine learning concerned with optimal control in Markov decision processes. Since stochastic games are essentially the multi-player extension of this problem class, it is very natural to apply reinforcement learning algorithms to them. We derive a game dynamic that arises

[1] Harsanyi and Selten themselves were aware of this problem. In order to ensure uniqueness of best responses, they devise the *logarithmic* tracing procedure which adds a logarithmic penalty term, forcing strategies towards the centroid. See also Chapter 3, where we take a similar approach for stochastic games.

if all players adopt expected SARSA, a well-established and often recommended reinforcement learning algorithm (van Seijen et al., 2009; Sutton and Barto, 2018). As we show, logit Markov QRE exactly coincide with the stationary points of that dynamic, which gives it additional credence as an approximate solution concept. Moreover, the finding opens possibilities for the interpretation of the aforementioned homotopy procedure: Following the path can be understood as all players employing the learning procedure with continuously increasing precision.

2.1.1 Related Literature

The following paragraphs summarize the most important references for the concepts we draw upon. Stochastic games and stationary equilibria find extensive formal treatments in Mertens et al. (2015), or in the monographs by Filar and Vrieze (1997) and Basar and Olsder (1999). A comprehensive, general introduction to the homotopy method is given by Zangwill and Garcia (1981), who focus on the mathematical foundation of this technique. Allgower and Georg (1990) complement this with a thorough treatment of its efficient and stable numerical implementation. Overviews of its applications in computational game theory are due to Borkovsky et al. (2010) and Herings and Peeters (2010); this includes in particular the linear tracing homotopy for stochastic games by Herings and Peeters (2004) themselves.

The homotopy we propose is based on the concept of quantal response equilibrium (QRE), first formulated for normal-form games by McKelvey and Palfrey (1995) and subsequently extended to extensive-form games as agent QRE in McKelvey and Palfrey (1998). A recent overview over applications and findings is provided by Goeree et al. (2016). While QRE is originally a behavioral solution concept, the classical equilibria typically arise as limiting cases, which can be utilized for computational purposes. Turocy (2005) is the first to discuss a homotopy based on the QRE correspondence, first for normal-form games. This is extended to extensive-form games in Turocy (2010), allowing the computation of sequential equilibria using agent QRE as a homotopy. Herings and Peeters (2004) first suggested that it should be possible to extend QRE to stochastic games. While the concept of logit Markov QRE and an according homotopy were then first discussed and used by Breitmoser et al. (2010) and Battaglini and Palfrey (2012), an explicit formal treatment is yet lacking in the literature. This is done in the present paper. Specifically, we provide a formal definition of Markov QRE and prove its existence for all finite stochastic games. For the special case of logit

Markov QRE, we show that the set of equilibria is finite and that all branches of the logit Markov QRE correspondence converge to stationary equilibria. Furthermore, we prove the existence of a unique smooth principal branch connecting the centroid of the strategy simplex to a unique limiting stationary equilibrium.

The remainder of this paper is organized as follows. Section 2.2 familiarizes the reader with stochastic games, stationary equilibrium and homotopy continuation. Section 2.3 introduces the concept of Markov quantal response equilibrium and establishes existence. Particular attention is devoted to the special case of logit Markov QRE. Section 2.4 describes the logit Markov QRE homotopy. We show that the set of logit Markov QRE consists of well-behaved, smooth paths and in particular contains a principal branch that can be followed numerically from an easy-to-compute starting point to a limiting stationary equilibrium of the game. This makes the homotopy both a computational tool and a potential selection criterion. Section 2.5 establishes that logit Markov QRE is always an ε-equilibrium with a bound for ε that, interestingly, is independent of the payoffs of the game and decreases. Finally, Section 2.6 relates logit Markov QRE to reinforcement learning, by showing that it coincides with the stationary points of a dynamic that is derived from the well-established reinforcement learning algorithm expected SARSA.

2.2 Prerequisites

In this section, we briefly review the fundamentals of stochastic games, stationary equilibrium, and homotopy continuation.

2.2.1 Stochastic Games

Stochastic games (Shapley, 1953) are essentially the generalization of Markov decision processes to multiple players. They are played as follows. An initial state is determined, possibly according to a random distribution. All players learn about the state and choose one of the actions available to them in that state. The state and their choices together determine instantaneous payoffs and a distribution from which a state for the next period is drawn, which is then played in the same way. The game may have terminal states and end when one of these is reached; otherwise, it will continue indefinitely.

Definition 1. ***Stochastic game.***
A stochastic game $\mathcal{G}$ is a tuple $\left(S, I, \boldsymbol{A}, \boldsymbol{U}, \boldsymbol{\Phi}, \boldsymbol{\Phi}_0, \boldsymbol{\delta}\right)$ *with*

S **:** *set of states.*

I**:** *set of players.*

A_{si}**:** *action set of player i in state s. $A_s = \bigtimes_{i \in I} A_{si}$ is the set of action profiles in state s. $A = \bigcup_{s \in S, i \in I} A_{si}$ denotes the set of all actions of any player in any state (understood as a disjoint union). Thus, $|A|$ represents the total number of actions of the game. We often use the index $_{sia}$ to refer to an action a that belongs to player i in state s.*

$\boldsymbol{u} = \Big(u_{si}(\boldsymbol{a}_s)\Big)_{\boldsymbol{a}_s \in A_s, s \in S, i \in I}$**:** *instantaneous payoff functions $u_{si} : A_s \to \mathbb{R}$.*

$\boldsymbol{\Phi} = \Big(\phi_{s \to s'}(\boldsymbol{a}_s)\Big)_{\boldsymbol{a}_s \in A_s, s, s' \in S}$**:** *state transition probabilities, where $\phi_{s \to s'}(\boldsymbol{a}_s)$ denotes the probability of transitioning from state s to s', if action profile $\boldsymbol{a}_s$ is played. Note that it may be that $\sum_{s' \in S} \phi_{s \to s'}(\boldsymbol{a}_s) < 1$; the remaining probability mass then represents the chance of the game to terminate.*

$\boldsymbol{\Phi}_0 \in \Delta(S)$**:** *a probability distribution over the initial state.*

$\boldsymbol{\delta} = \Big(\delta_i\Big)_{i \in I}$**:** *discount factors for all players.*

We restrict our attention to *finite discounted* stochastic games in *discrete time*, where the sets of states, players, and actions are all finite, time runs in discrete periods with an infinite horizon, and either future payoffs are discounted exponentially with $\delta_i < 1$, or $\boldsymbol{\delta} = \boldsymbol{1}$ but $\boldsymbol{\Phi}$ guarantees eventual termination with probability one. As usual, payoffs and state transitions extend to mixed strategy profiles.

2.2.2 Stationary Equilibrium

The most common solution concept for stochastic games is stationary equilibrium. Stationary equilibrium is a refinement of subgame perfect Nash equilibrium, in which all players are limited to the use of stationary strategies. Stationary strategies, in turn, restrict players to condition their responses exclusively on the current state of the game, but not on the history of play nor on time.

Definition 2. ***Stationary strategy.***
A stationary strategy profile $\boldsymbol{\sigma}$ assigns to each $(s, i) \in S \times I$, called the agent of player i in state s, a mixture $\boldsymbol{\sigma}_{si} \in \Delta(A_{si})$ over her available actions.

We will write σ_{sia} for the probability that a specific action $a \in A_{si}$ is chosen, $\boldsymbol{\sigma}_i$ for a stationary strategy of i, $\boldsymbol{\sigma}_{\text{-}i}$ for a strategy profile of all players except i, $\boldsymbol{\sigma}_s$ for a strategy profile in state s, and so on.

Definition 3. ***Stationary equilibrium.***
A stationary equilibrium $\boldsymbol{\sigma}$ is a subgame perfect equilibrium in stationary strategies.

Remark 1. ***Markov perfect equilibrium.***
Another solution concept found in the literature is Markov perfect equilibrium (MPE), which is a stationary equilibrium which requires that agents facing symmetric situations (in terms of payoffs and continuations) play symmetric strategies (Maskin and Tirole, 2001). In what follows, we will focus on the more general concept of stationary equilibrium, as this frees us from having to consider properties relating to symmetry. However, all results essentially apply to MPE as well. Also note that this distinction is not always made sharply in the literature: The term MPE is sometimes used in place of stationary equilibrium.

The existence of stationary equilibria in stochastic games has long been established in the literature.

Theorem 1. ***Existence of stationary equilibrium.***
Every finite and discounted stochastic game has a stationary equilibrium.

Proof. See Fink (1964), Takahashi (1964), or Sobel (1971). ■

By a straightforward application of Bellman's 1954 principle of optimality, stationary equilibria admit the following recursive representation.

Theorem 2. ***Recursive representation of stationary equilibrium.***
A stationary strategy profile $\boldsymbol{\sigma}$ constitutes a stationary equilibrium if and only if

1. *for all players $i \in I$, there exist state-player values $\boldsymbol{V}_i \in \mathbb{R}^{|S|}$ such that*

$$V_{si} \;=\; \max_{a \in A_{si}} u_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}) + \delta_i \sum_{s' \in S} \phi_{s \to s'}(a, \boldsymbol{\sigma}_{s,\text{-}i})\, V_{s'i}$$

holds for all states $s \in S$ and

2. *for all states $s \in S$, strategy profile $\boldsymbol{\sigma}_s$ constitutes a Nash equilibrium of the normal-form game with action spaces A_{si} for all i and payoffs*

$$U_{si}(\boldsymbol{a}_s) \;=\; u_{si}(\boldsymbol{a}_s) + \delta_i \sum_{s' \in S} \phi_{s \to s'}(\boldsymbol{a}_s)\, V_{s'i} \qquad\qquad (\boldsymbol{a}_s \in \boldsymbol{A}_s,\ i \in I).$$

Proof. See for example Doraszelski and Escobar (2010, p. 374). ■

Theorem 2 reflects that stochastic games can be seen as a set of normal-form games that are linked by state transitions and thus continuation values. Specifically, decision making in stochastic games is based on the present value of payoffs including the subsequent course of play:

$$U_{si}(\boldsymbol{\sigma}_s, \boldsymbol{V}_i) \;:=\; u_{si}(\boldsymbol{\sigma}_s) \;+\; \delta_i \sum_{s' \in S} \phi_{s \to s'}(\boldsymbol{\sigma}_s) V_{s'i}.$$

However, subsequent play and thus state values of course depend on the strategy profile of all players. Thus, it is necessary to determine equilibrium strategies and values in all states simultaneously, which is exactly what makes stochastic games so hard to solve.

Corollary 1. ***Characterization of stationary equilibrium.***
A Markov strategy profile $\boldsymbol{\sigma}$ with associated state values $\boldsymbol{V}$ constitute a stationary equilibrium if and only if for all $s \in S$ and $i \in I$:

$$\begin{aligned} \boldsymbol{\sigma}_{si} \quad &\in \quad \underset{\boldsymbol{\sigma}'_{si} \in \Delta(A_{si})}{\arg\max} \quad U_{si}(\boldsymbol{\sigma}'_{si}, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i), \\ V_{si} \quad &= \quad U_{si}(\boldsymbol{\sigma}_{si}, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i). \end{aligned}$$

Proof. Reformulation of Theorem 2. ■

Due to the maximization operators, the system of equations in Corollary 1 is generally very difficult to solve.[2] We will solve it by first rewriting the equations in terms of quantal response analysis and then applying homotopy continuation.

[2] Pakes and McGuire (1994, 2001) approach the system by means of value function iteration, i.e. by repeatedly solving for equilibrium strategies and updating the resulting state values. However, the procedure is not guaranteed to converge and, at best, pure-strategy equilibria can be found. Herings and Peeters (2004) transform the system by replacing each optimization with the corresponding Karush-Kuhn-Tucker conditions and performing a substitution of variables to ensure differentiability. However, the method is only guaranteed to succeed for generic games.

2.2.3 Homotopy Continuation

Homotopy continuation methods constitute a numerical solution method suited for high-dimensional, nonlinear systems of equations. Compared to most other numerical methods, they have the major advantage of working globally. Iterative Newton-methods for example are only locally convergent, meaning they require a good initial approximation to arrive at a solution at all. In contrast, homotopy methods arrive at solutions without such *a priori* knowledge, rendering them an exceptionally powerful tool. In this section, we will briefly sketch the procedure, as a basic understanding is necessary for the following parts of this paper.

The method generally proceeds in two steps: First the formulation of a suitable homotopy function, which implicitly defines a curve from an easily computed starting point to the desired solution; and then the numerical traversal of this curve until the solution is obtained. Intuitively, this resembles continuously transforming the problem until it is easy to obtain a solution, then reverting it back to the original form, while holding on to the solution.

More concretely, suppose one wants to find a solution $\boldsymbol{x}^*$ to $F(\boldsymbol{x}) = \mathbf{0}$, where $F : \mathbb{R}^n \to \mathbb{R}^n$ is a high-dimensional, nonlinear mapping. One constructs a function $G : \mathbb{R}^n \to \mathbb{R}^n$, such that a solution $\boldsymbol{x}_0 \in G^{-1}(\mathbf{0})$ is known or trivially obtained. Then, a homotopy parameter $\lambda \in [0, \bar{\lambda}]$ with $\bar{\lambda} \in (0, \infty]$ is introduced to construct a homotopy function $H(\boldsymbol{x}, \lambda)$, with $H : \mathbb{R}^{n+1} \to \mathbb{R}^n$, satisfying $H(\boldsymbol{x}, 0) = G(\boldsymbol{x})$ and $H(\boldsymbol{x}, \bar{\lambda}) = F(\boldsymbol{x})$. If H is constructed properly, it thus offers a continuous transformation of the hard problem $F(\boldsymbol{x}) = \mathbf{0}$ into the trivial one $G(\boldsymbol{x}) = \mathbf{0}$ and vice versa. The set of solutions $H^{-1}(\mathbf{0}) = \{(\boldsymbol{x}, \lambda) | H(\boldsymbol{x}, \lambda) = \mathbf{0}\}$ then contains a curve connecting the known solution $(\boldsymbol{x}_0, 0)$ to the desired solution $(\boldsymbol{x}^*, \bar{\lambda})$. Tracing this curve numerically to actually compute the solution is the second part of the method. This is done numerically, which is described in detail in Chapter 4.

The homotopy path might have turning points in the sense that the homotopy parameter λ is not monotonously increasing along the path, as illustrated in Figure 2.1. It is therefore generally not possible to follow the path by naively increasing λ. Instead, it is convenient to parameterize the homotopy path in terms of a path length parameter $\tau \in \mathbb{R}_0^+$ such that $H\Big(\boldsymbol{x}(0), \lambda(0)\Big) = \mathbf{0}$. Then, the path is defined by the following system of ordinary differential equations:

$$\frac{\partial(\boldsymbol{x}, \lambda)_k}{\partial \tau} = \eta \cdot (-1)^k \cdot \det\Big(J_{\text{-}k}(\boldsymbol{x}, \lambda)\Big) \qquad (k = 1, \dots, n+1) \tag{2.1}$$

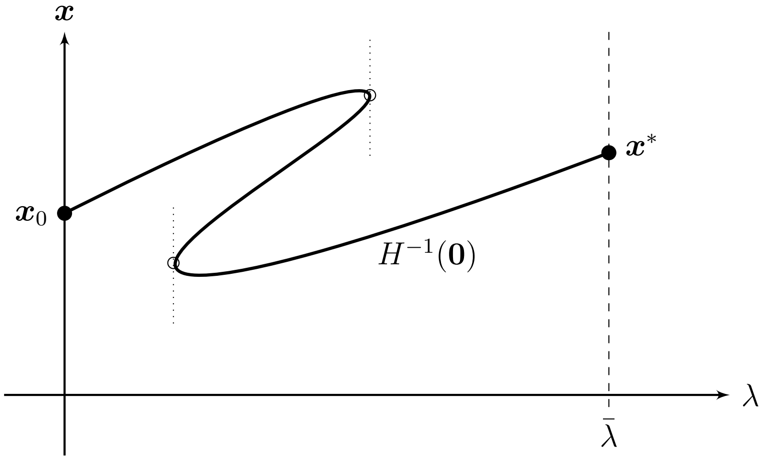


Figure 2.1: Turning Points of Homotopy Path

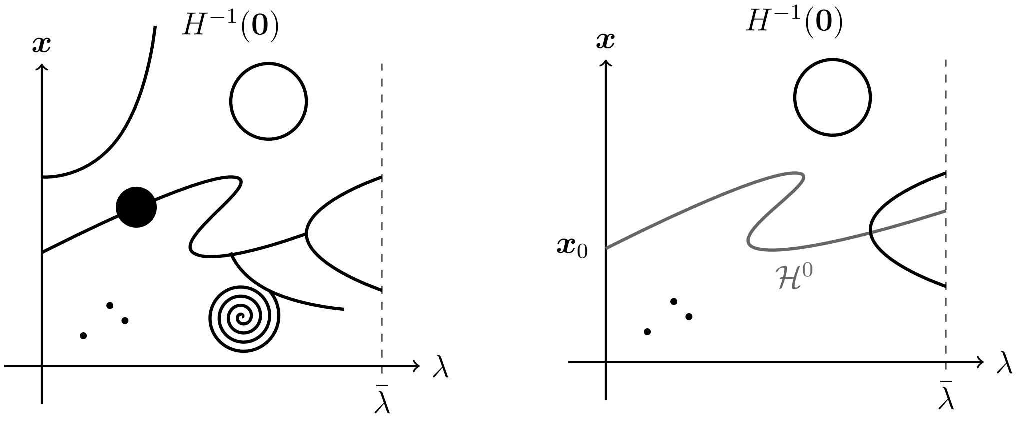
where $J(\boldsymbol{x},\lambda) = \frac{\partial H(\boldsymbol{x},\lambda)}{\partial(\boldsymbol{x},\lambda)}$ denotes the Jacobian matrix $J : \mathbb{R}^{n+1} \to \mathbb{R}^n \times \mathbb{R}^{n+1}$ of the homotopy function, $J_{\text{-}k}(\boldsymbol{x},\lambda)$ denotes the Jacobian without its k-th column and $\eta \in \mathbb{R}^+$ is a normalization factor. For details, see Zangwill and Garcia (1981, ch. 2).

In general, the solution set $H^{-1}(\mathbf{0})$ is not guaranteed to be as well-behaved as suggested by Figure 2.1. It might feature multidimensional segments, bifurcations, dead ends or spirals. For path tracking to be well-defined, the solution set $H^{-1}(\mathbf{0})$ must include a smooth branch $\mathcal{H}^0$ through $(\boldsymbol{x}_0, 0)$ that is almost everywhere one-dimensional, with only isolated crossings of secondary path segments. A corresponding illustration is provided in Figure 2.2.

Having covered the basics of stochastic games, stationary equilibrium and homotopy continuation, we are now in a position to formulate the results of this paper. We start by introducing stationary quantal response equilibrium.

2.3 Markov Quantal Response Equilibrium

In the quantal response framework, players are assumed to perceive payoffs only with some noise. In the resulting quantal response equilibrium (QRE) (McKelvey and Palfrey, 1995), players' actions appear stochastic and the probability of playing a particular action is increasing in its true payoff. This idea can be generalized to dynamic games by treating players at different decision nodes as independent agents. The corresponding equilibrium concept is called agent quantal response equilibrium (McKelvey and Palfrey, 1998). Finally, in the context of stochastic games with states as decision nodes, the corresponding equilibrium



Path tracking infeasible.

Path tracking feasible along smooth branch $\mathcal{H}^0$ (grey).

Figure 2.2: Possible shapes of the zero set of H.

concept is called Markov quantal response equilibrium (Breitmoser et al., 2010; Goeree et al., 2016).[3]

In this section, we formally define Markov quantal response equilibrium and establish existence. Then we characterize the set of logit Markov quantal response equilibria and show that its limit points are stationary equilibria.

2.3.1 Definition

In the context of stochastic games, players decide on optimal actions based on effective payoffs including continuation values. Specifically, let

$$U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) \;=\; u_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}) \;+\; \delta_i \sum_{s' \in S} \phi_{s \to s'}(a, \boldsymbol{\sigma}_{s,\text{-}i})\, V_{s'i}$$

denote the expected payoff from playing action a for player i in state s, given state values and strategies of the other players. In the quantal response framework, agent $(s, i) \in S \times I$ is assumed to perceive payoffs $U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)$ as

$$\hat{U}_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) \;=\; U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) + \varepsilon_{sia}$$

[3] Since we will make no requirements regarding symmetry, perhaps the more fitting term would be stationary QRE (see Remark 1). However, the term Markov QRE is already established, even where symmetry is not assumed.

with noise ε_{sia}. The error vector $\boldsymbol{\varepsilon}_{si} = (\varepsilon_{sia})_{a \in A_{si}}$ is assumed to be distributed according to a joint distribution with zero mean and density function $f_{si}(\boldsymbol{\varepsilon}_{si})$. Let

$$R_{sia} \;=\; \left\{ \boldsymbol{\varepsilon}_{si} \in \mathbb{R}^{|A_{si}|} \;\middle|\; \hat{U}_{sia}(\boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) \geq \hat{U}_{sia'}(\boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) \quad \forall\, a' \in A_{si} \right\}$$

denote agent (s,i)'s response set of action $a \in A_{si}$, specifying the realizations of $\boldsymbol{\varepsilon}_{si}$ such that agent (s,i) perceives action a as the one with the highest payoff. Then, the probability that agent (s,i) plays action a is given by the probability mass of the corresponding response set.

Definition 4. ***Markov quantal response equilibrium.***
A Markov quantal response equilibrium (Markov QRE) is a strategy profile $\boldsymbol{\sigma}$ *such that*

$$\sigma_{sia} \;=\; \int\limits_{R_{sia}} f_{si}(\boldsymbol{\varepsilon})\, d\boldsymbol{\varepsilon} \qquad\qquad (a \in A_{si},\ s \in S,\ i \in I).$$

Proving the existence of Markov QRE in stochastic games is a straightforward application of Brouwer's fixed-point theorem.

Theorem 3. ***Existence of Markov quantal response equilibrium.***
Every stochastic game $\mathcal{G}$ *has a Markov quantal response equilibrium.*

Proof. Similar to McKelvey and Palfrey (1995, theorem 1), with minor modifications. A Markov quantal response equilibrium $\boldsymbol{\sigma}$ is part of a fixed-point $(\boldsymbol{\sigma}, \boldsymbol{V})$ of the function $g(\boldsymbol{\sigma}, \boldsymbol{V}) = \big(g^{\sigma}(\boldsymbol{\sigma}, \boldsymbol{V}), g^{V}(\boldsymbol{\sigma}, \boldsymbol{V})\big)$ with

$$\begin{aligned} g^{\sigma}_{sia}(\boldsymbol{\sigma}, \boldsymbol{V}) &= \int\limits_{R_{sia}} f_{si}(\boldsymbol{\varepsilon})\, d\boldsymbol{\varepsilon} && \overset{!}{=} \sigma_{sia}, \\ g^{V}_{si}(\boldsymbol{\sigma}, \boldsymbol{V}) &= U_{si}(\boldsymbol{\sigma}_s, \boldsymbol{V}_i) && \overset{!}{=} V_{si}, \end{aligned}$$

for all states $s \in S$, players $i \in I$, and actions $a \in A_{si}$. Since strategies and state values are bounded as follows,

$$\begin{aligned} \sigma_{sia} &\in [0,1], \\ V_{si} &\leq \sum_{t=0}^{\infty} \delta_i^t \max_{\substack{s' \in S \\ \boldsymbol{a}_{s'} \in \boldsymbol{A}_{s'}}} \{u_{s'i}(\boldsymbol{a}_{s'})\} \;=\; \frac{1}{1-\delta_i} \cdot \max_{\substack{s' \in S \\ \boldsymbol{a}_{s'} \in \boldsymbol{A}_{s'}}} \{u_{s'i}(\boldsymbol{a}_{s'})\} \;<\; \infty, \\ V_{si} &\geq \frac{1}{1-\delta_i} \cdot \min_{\substack{s' \in S \\ \boldsymbol{a}_{s'} \in \boldsymbol{A}_{s'}}} \{u_{s'i}(\boldsymbol{a}_{s'})\} \;>\; -\infty, \end{aligned}$$

for all $s \in S$, $i \in I$ and $a \in A_{si}$, one can define g on a domain that is compact, convex, and nonempty. Furthermore, since the distribution of noise $\boldsymbol{\varepsilon}$ has a density, g is continuous. By Brouwer's fixed-point theorem, g has a fixed point. ■

For the remainder of this paper, we focus on *logit Markov QRE*, a special case arising from a specific distribution for $\boldsymbol{\varepsilon}$.

2.3.2 Logit Markov QRE

The most popular special case of quantal response is *logit choice* (Luce, 1959) where the probability σ_a of playing action a is given by the generalized logistic function

$$\sigma_a \;=\; \frac{\omega(u_a)}{\sum_{a'} \omega(u_{a'})}$$

with weighting function ω for the payoffs $u_{a'}$ associated with each action a'. Logistic rules of choice in the quantal response context arise from noise that is independently and identically distributed according to a Gumbel distribution with parameter $\lambda \in \mathbb{R}_0^+$ (extreme value distribution of type I).[4] The corresponding equilibrium can be expressed in closed form.

Definition 5. ***Logit Markov quantal response equilibrium.***
A stationary strategy profile $\boldsymbol{\sigma}$ with associated state values $\boldsymbol{V}$ constitute a logit Markov quantal response equilibrium with parameter $\lambda \in \mathbb{R}_0^+$ if and only if for all $s \in S$, $i \in I$, and $a \in A_{si}$:

$$\sigma_{sia} \;=\; \frac{\exp\left(\lambda U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)\right)}{\sum\limits_{a' \in A_{si}} \exp\left(\lambda U_{si}(a', \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)\right)} \tag{2.2a}$$

$$V_{si} \;=\; u_{si}(\boldsymbol{\sigma}_s) \,+\, \delta_i \sum_{s' \in S} \phi_{s \to s'}(\boldsymbol{\sigma}_s)\, V_{s',i} \;=\; U_{si}(\boldsymbol{\sigma}_s, \boldsymbol{V}_i) \tag{2.2b}$$

where

$$U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) \;=\; u_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}) \,+\, \delta_i \sum_{s' \in S} \phi_{s \to s'}(a, \boldsymbol{\sigma}_{s,\text{-}i}) V_{s',i}. \tag{2.2c}$$

[4] The Gumbel distribution has cumulative distribution function $F(\varepsilon) = e^{-e^{-\lambda\varepsilon}}$ and density function $f(\varepsilon) = e^{-\lambda\varepsilon} \cdot e^{-e^{-\lambda\varepsilon}}$. The parameter $\lambda \in \mathbb{R}_0^+$ controls the variance of the distribution. For $\lambda = 0$, the variance is infinite, and for $\lambda \to \infty$, the variance tends to zero.

Note that this resembles the definition of stationary equilibria in Corollary 1, except that maximization is replaced with logit choice. The logit formula for these equilibrium strategies can be derived from the Gumbel distribution as in McFadden (1973).

Theorem 4. ***Existence of logit Markov quantal response equilibrium.*** *Logit Markov QRE exists for all $\lambda \in \mathbb{R}_0^+$.*

Proof. Follows directly from Theorem 3. ■

2.3.3 Limiting Stationary Equilibria

Logit Markov QRE is parameterized by $\lambda \in \mathbb{R}_0^+$, which can be interpreted as the precision with which agents respond to payoffs. When $\lambda = 0$, the equilibrium is completely noisy and consists in uniform mixing over all over actions, i.e. the centroid strategies $\sigma_{sia} = \frac{1}{|A_{si}|}$ for all s, i, and $a \in A_{si}$. On the other hand, logit responses approach best responses as $\lambda \to \infty$. In particular, consider a sequence of logit Markov QRE, $\boldsymbol{\sigma}(\lambda)$, with precision parameter λ. If the sequence converges as $\lambda \to \infty$, then the limit point is a stationary equilibrium.

Theorem 5. ***Limiting stationary equilibria.*** *Consider the set of logit Markov QRE for a given game $\mathcal{G}$, i.e. the set of solutions $(\boldsymbol{\sigma}, \boldsymbol{V}, \lambda)$ to equation (2.2) for all $\lambda \in \mathbb{R}_0^+$. If $(\boldsymbol{\sigma}^*, \boldsymbol{V}^*, \infty)$ is a limit point of this set, then $\boldsymbol{\sigma}^*$ is a stationary equilibrium of $\mathcal{G}$.*

Proof. By contradiction, similar to McKelvey and Palfrey (1995, theorem 2), with minor modifications. Suppose $(\boldsymbol{\sigma}^n, \boldsymbol{V}^n, \lambda^n) \to (\boldsymbol{\sigma}^*, \boldsymbol{V}^*, \infty)$ is a sequence of logit Markov QRE, but $\boldsymbol{\sigma}^*$ is not a stationary equilibrium. Then, according to Theorem 2, there exists at least one state s where $\boldsymbol{\sigma}_s^*$ is not a Nash equilibrium of the normal-form game with payoffs $U_{si}(\,\cdot\,, \boldsymbol{V}_i^*)$. For this to be true, there must be an agent $(s, i) \in S \times I$ with actions $a, a' \in A_{si}$, where

$$U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}^*, \boldsymbol{V}_i^*) > U_{si}(a', \boldsymbol{\sigma}_{s,\text{-}i}^*, \boldsymbol{V}_i^*), \tag{2.3}$$

but $\sigma_{sia}^* = 0$ and $\sigma_{sia'}^* > 0$. Note that the latter means $\lim_n \sigma_{sia}^n < \lim_n \sigma_{sia'}^n$, and therefore, by equation (2.2a) and for n sufficiently large,

$$\frac{\exp\left(\lambda^n U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}^n, \boldsymbol{V}_i^n)\right)}{\sum\limits_{a'' \in A_{si}} \exp\left(\lambda^n U_{si}(a'', \boldsymbol{\sigma}_{s,\text{-}i}^n, \boldsymbol{V}_i^n)\right)} < \frac{\exp\left(\lambda^n U_{si}(a', \boldsymbol{\sigma}_{s,\text{-}i}^n, \boldsymbol{V}_i^n)\right)}{\sum\limits_{a'' \in A_{si}} \exp\left(\lambda^n U_{si}(a'', \boldsymbol{\sigma}_{s,\text{-}i}^n, \boldsymbol{V}_i^n)\right)}$$

Because denominator and λ^n are positive, this implies

$$U_{si}(a, \boldsymbol{\sigma}^n_{s,\text{-}i}, \boldsymbol{V}^n_i) < U_{si}(a', \boldsymbol{\sigma}^n_{s,\text{-}i}, \boldsymbol{V}^n_i)$$

Since U is continuous, this contradicts equation (2.3). ■

Theorem 5 suggests to find stationary equilibria by starting at any logit Markov QRE and then following a sequence of them while letting $\lambda \to \infty$. Provided a limit point exists, it is guaranteed to be a stationary equilibrium. The next section will show that such limit points indeed exist, and that the described procedure is computationally feasible, giving a homotopy interpretation to the logit Markov QRE correspondence.

2.4 Logit Markov QRE Homotopy

We now give a homotopy interpretation to logit Markov QRE: The system of equations characterizing them (Definition 5) can be used to compute stationary equilibria via homotopy continuation. The underlying intuition is as follows. The goal is to solve the complicated problem of finding stationary equilibria. To do so, we solve the simple problem of finding a logit Markov QRE and then distort the solution into a solution of the complicated problem, i.e. into a stationary equilibrium. Specifically, we propose a homotopy method that takes as starting point the unique logit Markov QRE at $\lambda = 0$ and follows a smooth path, called the principal branch of the homotopy, to a limiting stationary equilibrium.

In this section, we define a suitable homotopy function. Based on the homotopy function, we show that the number of logit Markov QRE is finite for all parameter values $\lambda \in \mathbb{R}^+_0$. Furthermore, we establish useful properties of the corresponding homotopy path, namely that it is almost everywhere one-dimensional and that all of its branches stabilize as $\lambda \to \infty$. Together with Theorem 5, this proves convergence of all branches of the graph of the logit Markov QRE correspondence to stationary equilibria.

2.4.1 Homotopy Function

One can obtain a homotopy function suitable for the computation of stationary equilibria directly from equation (2.2). $H : \mathbb{R}^{n+1} \to \mathbb{R}^n$ can be defined as follows:

$$H^{\sigma}_{sia}(\boldsymbol{\sigma}, \boldsymbol{V}, \lambda) = \sigma_{sia} - \frac{\exp\left(\lambda U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)\right)}{\sum\limits_{a' \in A_{si}} \exp\left(\lambda U_{si}(a', \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)\right)} \qquad \forall s, i, a \in A_{si} \tag{2.4a}$$

$$H^{V}_{si}(\boldsymbol{\sigma}, \boldsymbol{V}, \lambda) = V_{si} - u_{si}(\boldsymbol{\sigma}_s) - \delta_i \sum_{s' \in S} \phi_{s \to s'}(\boldsymbol{\sigma}_s)\, V_{s',i} \qquad \forall s, i \tag{2.4b}$$

The number of components is given by $n = |A| + |S \times I|$. Note that by construction, the zero set $H^{\text{-}1}(0)$ corresponds to the set of logit Markov QRE (compare equations 2.2 and 2.4). As shown by Turocy (2005, 2010) in the context of normal form and dynamic games, for strictly computational purposes it is helpful to apply some transformations to the system given by $H(\boldsymbol{\sigma}, \boldsymbol{V}, \lambda) = \mathbf{0}$ to obtain an alternative homotopy function $\tilde{H}$, with

$$\tilde{H}^{\sigma}_{si0}(\boldsymbol{\sigma}, \boldsymbol{V}, \lambda) \;=\; 1 - \sum_{a \in A_{si}} \sigma_{sia} \tag{2.5a}$$

$$\begin{aligned} \tilde{H}^{\sigma}_{sia>0}(\boldsymbol{\sigma}, \boldsymbol{V}, \lambda) \;=\; & \lambda\Big(U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) - U_{si}(a_0, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)\Big) \\ & - \Big(\log(\sigma_{sia}) - \log(\sigma_{si0})\Big) \end{aligned} \tag{2.5b}$$

$$\tilde{H}^{V}_{si}(\boldsymbol{\sigma}, \boldsymbol{V}, \lambda) \;=\; -V_{si} \;+ \sum_{a \in A_{si}} \sigma_{sia} U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) \tag{2.5c}$$

Note that the components referring to each agent's actions are asymmetric; the first line is a sum-to-one condition that determines σ_{si0}, referring to the first action a_0 of each agent. An equation of the form given by the second line then refers to every other action of the agent. A derivation of $\tilde{H}$ from H is given in Appendix 2.A.1. For computational purposes, it is helpful to apply a transformation of variables and use logarithmized strategies $\beta_{sia} := \log(\sigma_{sia})$ as variables; in particular, this prevents numerical blow-up of the Jacobian (Turocy, 2005). The according Jacobian, which is useful for computations, is listed in Appendix 2.A.2.

2.4.2 Existence of a Principal Branch

We now show that the zero set of H always contains a unique, smooth branch connecting the trivial logit Markov QRE at $\lambda = 0$ to a stationary equilibrium of the game at $\lambda = \infty$. Following this path allows to solve the game numerically. We begin with some prerequisites.

Theorem 6. ***Markov logit QRE are isolated.***
For fixed λ, the logit Markov QRE of a game are always isolated. Moreover, their number is always finite.

Proof. We will use the shorthand $\boldsymbol{x} := (\boldsymbol{\sigma}, \boldsymbol{V})$. Suppose $\boldsymbol{x} \in H|_{\lambda}^{-1}(0)$, i.e. $\boldsymbol{x}$ is a logit Markov QRE for the given λ. If $\boldsymbol{x}$ is not isolated in $H|_{\lambda}^{-1}(0)$, it must be part of a component that has locally at least dimension 1; by the real analytic implicit function theorem, this component must contain a real analytic path passing through $\boldsymbol{x}$. Parameterize this path with path length parameter $s \in (-\epsilon, \epsilon)$ as $\boldsymbol{x}(s)$ with $\boldsymbol{x}(0) = \boldsymbol{x}$. $\boldsymbol{x}(s)$, and by extension $H(\boldsymbol{x}(s))$, are real analytic in s. Because the path is in the zero set, we must have $H(\boldsymbol{x}(s)) = 0$ for all $s \in (-\epsilon, \epsilon)$, so that $H(s)$, is the zero function, and for all components H_k of H we must have $\frac{\partial^i H_k}{\partial s^i}$ for all $i \in \mathbb{N}$. Now consider equations (2.5); these consist of a polynomial and a logarithmic expression. After finitely many derivations, the partial derivatives of the polynomial must vanish, and only the derivatives of the logarithms remain. However, these can only vanish if $\sigma_{sia} = \sigma_{sia'}$ for all s, i and $a, a' \in A_{si}$. Thus, the only point that could potentially not be isolated is the centroid; however, this immediately implies that the centroid is also isolated. Thus, all points in $H|_{\lambda}^{-1}(0)$ are isolated.

Once isolation is established, finiteness follows by the following argument. Logit Markov QRE correspond to the zero set of H as given in equation (2.4), which consists of exponential polynomials.[5] By Khovanski's theorem (Marker, 1996, p. 757), the zero set of any set of exponential polynomials consists of finitely many connected components. ■

This has direct implications for the graph of the QRE correspondence: Because its points are isolated in all dimensions except λ, the graph must be 1-dimensional almost everywhere. The only exception are bifurcation points where multiple path segments cross. A necessary condition for such a point is that the Jacobian J of H is rank deficient; because wherever it has full rank n, the implicit function theorem guarantees that the zero set of H is locally a 1-dimensional manifold.

[5] Exponential polynomials in a set of variables may be defined recursively as follows: (i) All polynomials in these variables are exponential polynomials. (ii) Furthermore, if x, y are exponential polynomials, then xy, $x + y$, and e^x are also exponential polynomials. (iii) Only expressions obtainable from (i) and (ii) are exponential polynomials. Note that the set of exponential polynomials is closed under derivation.

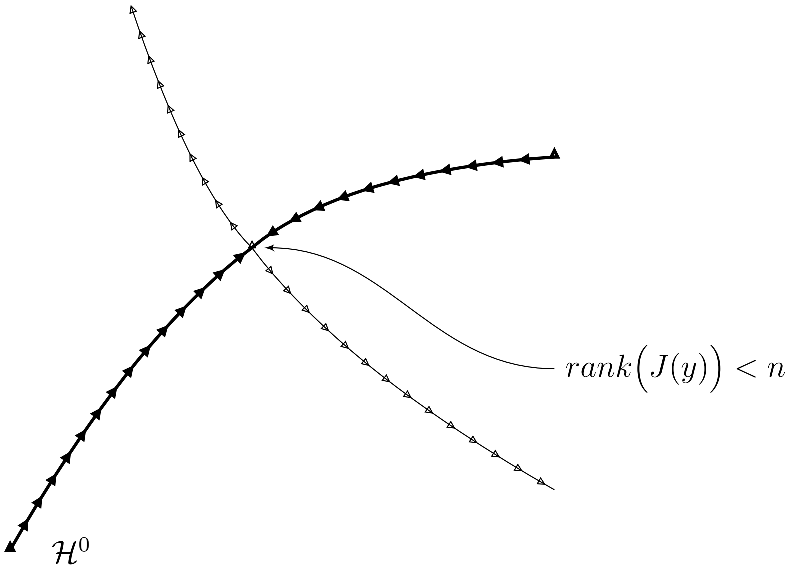


Figure 2.3: Example for a bifurcation point of the graph of the logit Markov QRE correspondence.

Thus, in these points all minor determinants of J must vanish. Because these determinants are polynomials in the derivatives of H, points for which $H(\boldsymbol{y}) = 0$ and rank $(J(\boldsymbol{y})) < n$ are again the zero set of a set of exponential polynomials, so that Khovanski's theorem implies that only a finite number of bifurcations exist.

By Lyapunov-Schmidt reduction one can decompose $H^{-1}(0)$ into the different segments and formally show that the tangents on both sides of these simple bifurcation points point in same or exactly opposite directions, i.e.

$$\lim_{\tau \to \tilde{\tau}^+} t\Big(\boldsymbol{y}(\tau)\Big) \;=\; \pm \lim_{\tau \to \tilde{\tau}^-} t\Big(\boldsymbol{y}(\tau)\Big)$$

(Allgower and Georg, 1990, theorem 8.1.14). This establishes that it is always possible to find a unique smooth continuation of a current path segment across such bifurcation points. We will refer to collections of such segments that continue each other as paths. Note that the paths are real analytic, as they are implicitly defined by a system of real analytic equations.

Theorem 7. ***Finite number of turning points.***
Each path has at most a finite number of turning points in any of the dimensions σ_{sia}, $\boldsymbol{V}_{si}$, or λ.

Proof. Turning points are characterized by one sub-determinant of J crossing 0 (compare equation 2.1). Together with $H = 0$, turning points are thus characterized by a set of exponential polynomials. Khovanski's theorem (Marker, 1996, p. 757) then again implies that their number is finite. ■

Theorem 8. ***Unique and transversal solution at*** $\lambda = 0$**.**
The system $H(\boldsymbol{\sigma}, \boldsymbol{V}, 0) = 0$ *has a unique solution. There, a unique path in* $H^{-1}(0)$ *crosses the hyperplane characterized by* $\lambda = 0$ *transversally.*

Proof. For $\lambda = 0$, equation (2.4a) implies $\sigma_{sia} = \frac{1}{|A_{si}|}$ for all s, i and $a \in A_{si}$. Since any $\boldsymbol{\sigma}$ induces a unique $\boldsymbol{V}$, the solution must be unique.

Regarding transversality, it is sufficient to show that the sub-determinant of the matrix $J_{\text{-}\lambda}$ does not vanish at that point.[6] To that end, consider the Jacobian of H, where H is given by (2.4). $J_{\text{-}\lambda}$ has the following structure:

$$J_{\text{-}\lambda}(\boldsymbol{\sigma}, \boldsymbol{V}, \lambda) = \begin{pmatrix} \frac{\partial H^{\sigma}_{sia}(\boldsymbol{\sigma},\boldsymbol{V},\lambda)}{\partial \sigma_{s'i'a'}} & \frac{\partial H^{\sigma}_{sia}(\boldsymbol{\sigma},\boldsymbol{V},\lambda)}{\partial V_{s',i'}} \\ \frac{\partial H^{V}_{si}(\boldsymbol{\sigma},\boldsymbol{V},\lambda)}{\partial \sigma_{s',i',a'}} & \frac{\partial H^{V}_{si}(\boldsymbol{\sigma},\boldsymbol{V},\lambda)}{\partial V_{s'i'}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{I}_{|A|} & \boldsymbol{0} \\ \frac{\partial H^{V}_{si}(\boldsymbol{\sigma},\boldsymbol{V},\lambda)}{\partial \sigma_{s',i',a'}} & (\boldsymbol{I} - \delta\bar{\boldsymbol{\Phi}}) \end{pmatrix}$$

The blocks in the second expression are derived as follows. First, when setting $\lambda = 0$ in (2.4a), $\frac{\partial H^{\sigma}_{sia}(\boldsymbol{\sigma},\boldsymbol{V},\lambda)}{\partial \sigma_{s'i'a'}}$ is equal to 1 if $s = s', i = i'$ and $a = a'$, and 0 else. The top left block is thus the identity matrix $\boldsymbol{I}_{|A|}$. Similarly, $\frac{\partial H^{\sigma}_{sia}(\boldsymbol{\sigma},\boldsymbol{V},\lambda)}{\partial V_{s',i'}} = 0$ for $\lambda = 0$, so that the top right block consists of zeros only. The bottom left block can be ignored for the present purpose. Finally, in the bottom right block, $\frac{\partial H^{V}_{si}(\boldsymbol{\sigma},\boldsymbol{V},\lambda)}{\partial V_{s'i'}} = 0$ if $i \neq i'$. The block is thus itself block-diagonal, with one block per player. For $i = i'$, $\frac{\partial H^{V}_{si}(\boldsymbol{\sigma},\boldsymbol{V},\lambda)}{\partial V_{s'i'}} = \delta_i \phi_{s\to s'}(\boldsymbol{\sigma}_s)$, unless $s = s'$, in which case $\frac{\partial H^{V}_{si}(\boldsymbol{\sigma},\boldsymbol{V},\lambda)}{\partial V_{s'i'}} = 1 - \delta_i \phi_{s\to s'}(\boldsymbol{\sigma}_s)$. Thus, the sub-blocks are of the form $\boldsymbol{I}_{|S|} - \delta_i \boldsymbol{\Phi}$, where $\boldsymbol{\Phi}$ is a transition matrix. Because $\delta_i < 1$, each of these sub-blocks is invertible. In extension, the complete lower right block is invertible; together with the top two blocks this ensures that the complete matrix has full rank. ■

We will call this solution at $\lambda = 0$ the *starting point*, and can now state the main result of this section.

Theorem 9. ***Existence of the principal branch.***
There exists a unique path that begins at the starting point and converges to a stationary equilibrium of the game as $\lambda \to \infty$*. This path is called the principal branch and can serve as homotopy path.*

[6] $J_{\text{-}\lambda}$ is the square matrix obtained by deleting the λ-column from J. If the determinant is non-zero, the Jacobian must have full rank, so that the implicit function theorem implies that $H^{-1}(0)$ is locally one-dimensional, i.e. a path. The determinant of $J_{\text{-}\lambda}$ then determines the λ-component of the tangent at that point (see equation 2.9). If it is non-zero, the path can not be parallel to the hyperplane $\lambda = 0$.

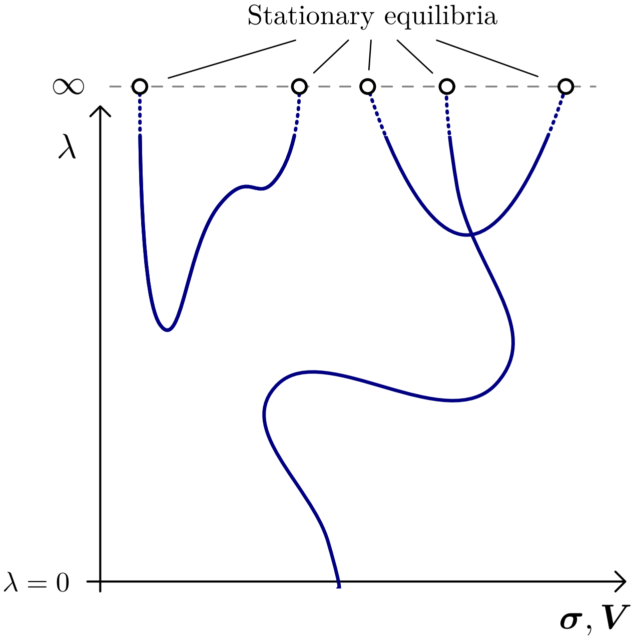


Figure 2.4: Structure of the graph of the logit Markov QRE correspondence. The graph consists of 1-dimensional segments, with at most finitely many transversal bifurcation points. A unique solution exists at $\lambda = 0$, where the principal branch begins. For $\lambda \to \infty$, the branches converge to stationary equilibria.

Proof. As just shown, a unique path starts there transversally to the hyperplane $\lambda = 0$. Next, remember that $H^{-1}(0)$ is bounded in all dimensions except λ (see proof of Theorem 3). This implies that no path can go off to infinity in any of the other directions. Any path stretching to $\lambda = \infty$ must converge; this follows because paths are bounded and have a finite number of turning points, i.e. are eventually monotonic. Any path in $H^{-1}(0)$ must therefore either be a closed loop or have two limit points at $\lambda = \infty$. The only exception is the path emanating from the starting point, which must have as other endpoint a limit point at $\lambda = \infty$. Finally, Theorem 5 already showed that the limit points of logit Markov QRE, i.e. of the set $H^{-1}(0)$, at $\lambda = \infty$ are stationary equilibria. ■

The structure of the graph of the logit Markov QRE correspondence is summarized in Figure 2.4. The existence of the principal branch allows to compute a stationary equilibrium of any finite discounted stochastic game in the following way. First compute the starting point; this is trivial, as $\boldsymbol{\sigma}$ is just the centroid and the corresponding values easily obtained. Next, follow the path numerically (see Chapter 4 for a detailed description of an algorithm to do so). The possible existence of branching points is no impediment, as one can simply continue across them; in principle, they even allow to compute further equilibria by following the secondary branches (Allgower and Georg, 1990, ch. 8). Once the path has converged with the desired accuracy, one has obtained a stationary equilibrium. An implementation of this procedure is publicly available as part of the python package `sgamesolver`, which is introduced in Chapter 4.

In addition to computation, this procedure of course entails a selection from the set of all stationary equilibria. Whether this represents a convincing criterion will certainly depend on whether logit Markov QRE in itself is a reasonable approximate solution concept and whether traversing the principal branch can be given a convincing interpretation. The following two sections establish properties of logit Markov QRE in that regard. First, we show that logit Markov QRE are ε-equilibria, and derive a bound for ε. Consequently, while players do incur some loss relative to perfect play by adopting the logit choice rule, this loss is bounded and decreasing in λ. Then, Section 2.6 shows that logit Markov QRE can arise from a very reasonable learning algorithm, and that the traversal of the principal branch in a sense resembles all players using that algorithm with continuously increasing precision.

2.5 Logit Markov QRE as ε-equilibrium

In this section, we will show that every logit Markov QRE is an ε-equilibrium. We derive a bound for ε that, interestingly, does not depend on the payoffs of the game, but is given by $\varepsilon \leq \frac{J-1}{(1-\delta)\lambda e}$, where $J = \max_{s,i} |A_{si}|$ represents the maximum number of actions held by any player in any state, and $\delta = \max_i \delta_i$. ε-equilibrium is defined as follows.

Definition 6. *ε-equilibrium.*
In a normal-form game with payoff functions u_i, a strategy profile $\boldsymbol{\sigma}$ is an ε-equilibrium if and only if, for all $i \in I$,

$$\max_{\boldsymbol{\sigma}'_i} u_i(\boldsymbol{\sigma}'_i, \boldsymbol{\sigma}_{\text{-}i}) - u_i(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_{\text{-}i}) \leq \varepsilon$$

Likewise, in a stochastic game where total discounted payoffs for player i in state s are given by $U_{si}(\boldsymbol{\sigma})$, a stationary strategy profile $\boldsymbol{\sigma}$ is an ε-equilibrium if and only if, for all s and i,

$$\max_{\boldsymbol{\sigma}'_i} U_{si}(\boldsymbol{\sigma}'_i, \boldsymbol{\sigma}_{\text{-}i}) - U_{si}(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_{\text{-}i}) \leq \varepsilon$$

We begin with a simple, one-shot decision situation where we derive a bound for the maximum loss a decision maker can incur when choosing according to logit probabilities, rather than maximizing. While the derivation is rather straightforward, we have not found this bound in the literature. Consider an agent who has to choose from a finite set of J options, with utilities $\boldsymbol{u} = (u_1, ..., u_J) \in R^J$. Denote $u_{\max} = \max \boldsymbol{u}$. For a given precision parameter $\lambda > 0$, the *logit choice rule* consists of choosing option i with probability

$$\sigma_i(\lambda) := \frac{\exp(\lambda u_i)}{\sum_j \exp(\lambda u_j)}$$

Unless all options have the same utility, this is obviously worse than simply choosing a maximizing option. However, the loss can be bounded as follows. Define the loss incurred from following logit choice rather than maximizing as

$$loss(\lambda) := u_{\max} - \sum_j \sigma_j u_j$$

Theorem 10. ***Bound for the loss incurred from logit choice.***
For any $(u_1, ..., u_J) \in \mathbb{R}^J$ *and any* $\lambda \geq 0$, $loss(\lambda) \leq \frac{J-1}{\lambda e}$.

Proof. We begin by rewriting the loss function as

$$\begin{aligned} loss(\lambda) &= u_{\max} - \sum_j \sigma_j u_j \\ &= \sum_j \frac{\exp(\lambda u_j)}{\sum_k \exp(\lambda u_k)} (u_{\max} - u_j) \\ &= \sum_j \exp\left[\lambda u_j - \log\left(\sum_k \exp\left(\lambda u_k\right)\right)\right] (u_{\max} - u_j) \end{aligned}$$

Note the term $\log\left(\sum_k \exp\left(\lambda u_k\right)\right)$, which can be bounded as follows:

$$\begin{aligned} \lambda u_{\max} = \log\Big(\exp\left(\lambda u_{\max}\right)\Big) &\leq \log\left(\sum_k \exp(\lambda u_k)\right) \\ &\leq \log\Big(J \exp(\lambda u_{\max})\Big) = \lambda u_{\max} + \log J \end{aligned}$$

Using the lower bound in $loss(\lambda)$:

$$\begin{aligned} loss(\lambda) &= \sum_j \exp\left[\lambda u_j - \log\left(\sum_k \exp\left(\lambda u_k\right)\right)\right](u_{\max} - u_j) \\ &\leq \sum_j \exp\left[\lambda u_j - \lambda u_{\max}\right](u_{\max} - u_j) \\ &= \sum_j \frac{(u_{\max} - u_j)}{\exp\left(\lambda u_{\max} - \lambda u_j\right)} \end{aligned}$$

This sum comprises of terms of the form $t = ze^{-\lambda z}$, with $z, \lambda \geq 0$. First and second derivatives of these terms are

$$\frac{\partial t}{\partial z} = e^{-\lambda z} - \lambda z e^{-\lambda z}$$

$$\frac{\partial^2 t}{\partial z^2} = \lambda(\lambda z - 2)e^{-\lambda z}$$

so that their global maximum is attained at $z = \lambda^{-1}$, where $t_{\max} = \frac{1}{\lambda e}$. Finally, at least one of these summands is zero, so that

$$loss(\lambda) \leq \sum_j \frac{(u_{\max} - u_j)}{\exp\left(\lambda u_{\max} - \lambda u_j\right)} \leq \frac{J-1}{\lambda e}$$

as claimed. ■

An interesting feature of this bound is that it does not depend on $\boldsymbol{u}$, which might be counter-intuitive at first glance. The reason for this is that the logit choice rule itself is sensitive to $\boldsymbol{u}$: If the utility difference between good and bad options increases, the rule shifts probability mass to the former. For example, doubling the stakes by doubling all utilities in $\boldsymbol{u}$ has the same effect on choices as doubling the precision parameter λ; thus, while the cost of choosing a bad option increases, this is offset by a decrease in probability. Also note that the bound depends on the number of actions, J. The reason is that adding additional copies of the worst action to the decision problem increases the total probability with which this class of actions is played – a general property of the logit choice rule.

Before moving to intertemporal decision problems, we can already use this to state a result regarding one-shot games.

Theorem 11. ***Logit QRE as ε-equilibrium.***
Consider a finite normal form game $\mathcal{G}$, and suppose $\boldsymbol{\sigma}$ is a logit QRE with precision

parameter λ. Then $\boldsymbol{\sigma}$ is an ε-equilibrium of $\mathcal{G}$ with $\varepsilon \leq \frac{J-1}{\lambda e}$, where J is the maximum number of actions held by any player.

Proof. This follows almost directly from Theorem 10. In a logit QRE, player i with action set A_i randomizes between actions according to the logit rule, with $\boldsymbol{u} = (u(a, \boldsymbol{\sigma}_{\text{-}i}))_{a \in A_i}$. Thus, the maximum loss incurred is bounded by $\frac{|A_i|-1}{\lambda e}$. The theorem then results from taking the maximum over all players. ■

We now extend Theorem 10 to finite Markov decision processes (MDPs). We take as given an MDP with a finite set of states S, finite action sets A_s, instantaneous utilities u_s, transition probabilities $\phi_{s \to s'}$ and discount factor $\delta < 1$. Note that this conforms to Definition 1 for stochastic games, just with a singleton player set. We will consider stationary policies, which assign to each state a probability distribution over the available actions. A *logit policy* with precision parameter λ is defined by choosing action $a \in A_s$ in state s with probability

$$\sigma_{sa}(\lambda) := \frac{\exp\left(\lambda U(a, \boldsymbol{V})\right)}{\sum_{a' \in A_s} \exp\left(\lambda U(a', \boldsymbol{V})\right)}$$

where we again use a shorthand for total discounted utility

$$U_s(a, \boldsymbol{V}) := u_s(a) + \delta \sum_{s' \in S} \phi_{s \to s'}(a) V_{s'}$$

and it is assumed that the continuation values $\boldsymbol{V}$ are implicitly defined by the recursive relation

$$V_s = u_s(\boldsymbol{\sigma}) + \delta \sum_{s' \in S} \phi_{s \to s'}(\boldsymbol{\sigma}) V_{s'}$$

Essentially, a logit policy consists in applying logit choice to total discounted utility, where continuation values are consistent with the policy itself. Note that this simply corresponds to logit Markov QRE applied to a single-player stochastic game.

Theorem 12. ***Bound for the loss incurred from a logit policy.***
Consider a given finite MDP. If $\boldsymbol{\sigma}$ is a logit policy with precision parameter λ, then in any state, the loss in total discounted utility relative to an optimal policy is bounded from above by $\frac{J-1}{(1-\delta)\lambda e}$, where $J = \max_s |A_s|$ is the maximum number of actions in any state.

Proof. The relation to Theorem 10 is probably not surprising, as here, losses are just compounded. However, a bit of careful work is necessary, because Theorem 10 essentially concerns one-shot decisions. But changing the policy in period t affects the decision situation in $t-1$ via continuation values, so that it is not obvious that the bound from Theorem 10 then keeps holding in $t-1$.

We begin introducing some notation. Let $\boldsymbol{\sigma}^*$ be an optimal policy for the MDP, with associated values $\boldsymbol{V}^*$. Denote by $\boldsymbol{u}^* = \Big(u_1(\boldsymbol{\sigma}_1^*), u_2(\boldsymbol{\sigma}_2^*), ..., u_{|S|}(\boldsymbol{\sigma}_{|S|}^*)\Big)^T$ the vector of instantaneous utilities and $\boldsymbol{\Phi}^*$ be the transition matrix induced by it, so that $\boldsymbol{\Phi}^*_{m,n} = \phi_{m \to n}(\boldsymbol{\sigma}^*_m)$. Likewise, denote by $\boldsymbol{V}$, $\boldsymbol{u}$, $\boldsymbol{\Phi}$ the same quantities associated with the logit policy $\boldsymbol{\sigma}$. Finally, let $\varepsilon = \frac{J-1}{\lambda e}$ and let $\boldsymbol{\varepsilon}$ be a vector of length $|S|$ with all entries ε.

Applying the bound from Theorem 10 to the definition of a logit policy yields

$$U_s(\boldsymbol{\sigma}_s^*, \boldsymbol{V}) - U_s(\boldsymbol{\sigma}_s, \boldsymbol{V}) = u_s^* + \delta \sum_{s'} \phi_{s \to s'}(\boldsymbol{\sigma}_s^*)\boldsymbol{V}_{s'} - u_s - \delta \sum_{s'} \phi_{s \to s'}(\boldsymbol{\sigma}_s)\boldsymbol{V}_{s'} \leq \varepsilon$$

in every state, or, in the more parsimonious vector notation

$$\boldsymbol{u}^* + \delta\boldsymbol{\Phi}^*\boldsymbol{V} - \boldsymbol{u} - \delta\boldsymbol{\Phi}\boldsymbol{V} \leq \boldsymbol{\varepsilon}$$

The above essentially states that, starting from the logit policy $\boldsymbol{\sigma}$, the gain from a one-shot deviation – changing the policy today, but returning to $\boldsymbol{\sigma}$ tomorrow – is at most ε. This is also why $\boldsymbol{V}$ appears twice as continuation value, and $\boldsymbol{V}^*$ not at all. However, the goal is to bound a total, rather than a one-shot deviation, that is, bound $\boldsymbol{V}^* - \boldsymbol{V}$. To get there, we use the recursive definitions of $\boldsymbol{V}^*$ and $\boldsymbol{V}$, namely

$$\boldsymbol{u}^* = \boldsymbol{V}^* - \delta\boldsymbol{\Phi}^*\boldsymbol{V}^* \qquad \text{and} \qquad \boldsymbol{u} = \boldsymbol{V} - \delta\boldsymbol{\Phi}\boldsymbol{V}$$

After plugging both into the above inequality, the $\delta\boldsymbol{\Phi}\boldsymbol{V}$ cancel and one obtains

$$\boldsymbol{V}^* - \delta\boldsymbol{\Phi}^*\boldsymbol{V}^* - \boldsymbol{V} + \delta\boldsymbol{\Phi}^*\boldsymbol{V} = (\boldsymbol{I} - \delta\boldsymbol{\Phi}^*)(\boldsymbol{V}^* - \boldsymbol{V}) \leq \boldsymbol{\varepsilon}$$

Then, using the fact that $\boldsymbol{\Phi}^*$ is a transition matrix and $\delta < 1$,

$$\boldsymbol{V}^* - \boldsymbol{V} \leq (\boldsymbol{I} - \delta\boldsymbol{\Phi}^*)^{-1}\boldsymbol{\varepsilon} = \sum_{t=0}^{\infty}(\delta\boldsymbol{\Phi}^*)^t\boldsymbol{\varepsilon}$$

As a transition matrix, $\boldsymbol{\Phi}^*$ has row sums less or equal to one, and thus $\boldsymbol{\Phi}^*\boldsymbol{\varepsilon} \leq \boldsymbol{\varepsilon}$:

$$\boldsymbol{V}^* - \boldsymbol{V} \leq \sum_{t=0}^{\infty}(\delta\boldsymbol{\Phi}^*)^t\boldsymbol{\varepsilon} \leq \sum_{t=0}^{\infty}\delta^t\boldsymbol{\varepsilon} = \frac{1}{1-\delta}\boldsymbol{\varepsilon}$$

which completes the proof. ■

We can now turn to stochastic games and state the main result of this section.

Theorem 13. ***Logit Markov QRE as ε-equilibrium.***
Consider a stochastic game $\mathcal{G}$, and suppose $\boldsymbol{\sigma}$ is a logit Markov QRE with precision parameter λ. Then $\boldsymbol{\sigma}$ is also an ε-equilibrium of $\mathcal{G}$ with $\varepsilon \leq \frac{J-1}{(1-\delta)\lambda e}$, where $J = \max_{s,i} |A_{si}|$ is the maximum number of actions held by any player in any state and $\delta = \max_i \delta_i$ is the maximum discount factor among players.

Proof. This now follows quickly from Theorem 12. In a Markov QRE given by $\boldsymbol{\sigma}$, the strategy $\boldsymbol{\sigma}_i$ of each player i is a logit policy, given $\boldsymbol{\sigma}_{\text{-}i}$. Thus, i can gain at most $\frac{J_i-1}{(1-\delta_i)\lambda e}$ in any state, where $J_i = \max_s |A_{si}|$. Taking the maximum over all players then yields the claim. ■

2.6 Logit Markov QRE and Reinforcement Learning

In this section, we establish a connection between logit Markov QRE and reinforcement learning. Specifically, we will show that the set of logit Markov QRE corresponds to the stationary points of a game dynamic that arises if all players follow the well-established reinforcement learning algorithm SARSA. Before we turn to that, the following subsection establishes some background on reinforcement learning.

2.6.1 Background: Reinforcement Learning

To give unfamiliar readers some context, we will briefly discuss reinforcement learning in general before then introducing the specific algorithm SARSA, which has a close connection to logit Markov QRE. Everything in this introduction owes to the excellent textbook by Sutton and Barto (2018), who characterize the topic as follows (p. 2):

> Reinforcement learning, like many topics whose names end with “ing,” such as machine learning and mountaineering, is simultaneously a problem, a class of solution methods that work well on the problem, and the field that studies this problem and its solution methods.

The solution methods or algorithms under this label deal with the problem of optimal control in Markov decision processes – meaning any decision problem that is defined by a set of states (S), a set of actions per state (A_s), each of which is associated with a distribution over a set of rewards (R) and a distribution over the subsequent state (ϕ).[7] Dynamic programming methods are also geared towards this problem class; but unlike these, reinforcement learning generally does not require the decision maker to have complete knowledge of the MDP. Instead, agents are assumed to have no initial knowledge, but can learn from direct, ongoing experience with the environment – by making decisions and facing their consequences. Thus, one of the central issues of interest is the trade-off between exploration of the *ex ante* unknown environment, and exploitation of the information collected thus far. Furthermore, reinforcement learning methods are commonly designed with limitations in computing power and memory in mind, a consideration again absent in dynamic programming. Similar to dynamic programming, reinforcement learning algorithms typically involve (iterative) approximation of value and/or optimal policy functions.

We consider here the class of finite MDPs (S, A, R finite); for these, the value function approximation is typically a mapping from states (or state-action pairs, see later) to the reals, meaning so-called tabular methods are feasible (or even necessary, if the environment has no further structure). The agent is assumed to undergo episodes of experience as follows: An initial state is drawn according to some distribution. The agent learns the state; he can choose an available action; a reward is drawn from the associated distribution and revealed, as is the subsequent state. This repeats until a terminal state is reached and a new episode starts. The presence of terminal states may be replaced with exponential discounting. The agent is assumed to have no further knowledge of the environment, including prior beliefs on the relevant distributions. Learning is typically "online", meaning the rewards collected during this process matter, and episodes are either limited in number or themselves discounted.

Commonly in reinforcement learning, the quantities estimated are not state values, but state-action values: the (possibly discounted) expected total reward following a specific action in a given state. These are often called q- or q_{sa}-values. In the notation of this paper, q_{sa} corresponds to $U_s(a) = u_s(a) + \delta \sum_{s'} \phi_{s \to s'}(a) V_s$

[7] The reader will immediately recognize the similarity to Defintion 1 of stochastic games; the only difference to a one-player-version of the latter is that rewards are allowed to be randomly distributed. In contexts where only the expected reward matters, one can simply use $u_s(a) = \mathbb{E}(R|s, a)$.

(where the player-index i is omitted for now). The importance of q-values over simple state values V is due to the fact that many methods are *model free*, i.e. stay agnostic with regards to the transitions ϕ throughout the learning process. Without knowledge or an estimate of ϕ, q-values cannot be calculated from V, and V in itself is actually quite useless for guiding behavior.

The conceptually simplest tabular methods are so-called *Monte Carlo* methods, dating back at least as far as dynamic programming. Essentially, these methods store a running average value for each state-action $a \in A_s$. After each completed episode, the value estimate of each state that was visited (or each state-action in model-free algorithms) is updated using the total reward collected following the visit. Variants differ in how a policy is derived from current estimates, how exactly the update is weighted, and so on.

Probably the most important breakthrough in reinforcement learning were the so-called *temporal difference* (TD) methods. Unlike Monte Carlo methods, they do not delay updating the value function until after the episode. Rather, whenever an action is taken, its q-estimate is updated right away, using the sum of the immediate reward and a (discounted) current estimate of the resulting states' value. Basing an estimate upon a previous estimate is called *bootstrapping* and a feature TD methods share with dynamic programming, in contrast with e.g. Monte Carlo methods or solving a known MDP with linear programming. TD methods have proven to be very effective in a wide range of settings; moreover, there is ample evidence for neurological correlates of the prediction error and subsequent correction of value estimates in humans and other organisms.

TD methods are defined by an action selection rule that guides exploration, and an update rule that prescribes how exactly experience is incorporated into value estimates. Their choice is rather free; under the mild restriction that each state-action pair is revisited an unlimited number of times, most actions selection rules typically ensure convergence in the limit. The exact choice of course affects speed of convergence and, in online learning, the degree of loss due to exploration along the way. Logit choice (often called *softmax* in this context) and ϵ-greedy are the most widely studied (both of course with respect to the current q-estimates)[8].

[8] The *greedy* policy always picks the current best estimate, and thus focuses on exploitation only. ϵ-greedy balances exploration and exploitation by choosing the current best estimate with probability $1 - \epsilon$ and a completely random action with ϵ. Logit choice also balances both, but takes into account the estimated value differences – actions estimated to be very bad are still picked, but with probability decreasing in the utility gap. The advantage in online learning should be obvious.

Update rules on the other hand give rise to further conceptual distinctions within the TD class. Generally, after taking action a in state s, the update is performed as follows

$$
\begin{aligned}
q'_{sa} &= q_{sa} - \alpha(u_s(a) + \delta q_{s'a'} - q_{sa}) && \text{(2.6a)}\\
&= (1-\alpha)q_{sa} + \alpha(u_s(a) + \delta q_{s'a'}) && \text{(2.6b)}
\end{aligned}
$$

where s' is the observed subsequent state, and a' the prescribed action in that state – how exactly a' is chosen in updating is the main distinguishing feature between TD methods and will be discussed below. Equation (2.6a) illustrates that these methods work by continuously correcting the estimates by an experienced prediction error: The bracketed term represents realized minus expected reward. TD methods thus fall under the more general mathematical concept of *stochastic approximation processes* (Borkar, 2008). Equation (2.6b) represents the new estimate as a convex combination of the previous estimate and realized reward. α is a gain parameter; typically, it is chosen to be decreasing over time: $\sum_t \alpha_t = \infty$ and $\sum_t \alpha_t^2 < \infty$ together ensure convergence for most algorithms, an obvious choice being $\alpha_t = {}^1\!/t$.

As mentioned, the choice of a' in the update gives rise to further sub-categorization within TD methods. Perhaps the most widely known algorithm *q-learning* is a so-called *off-policy method*, meaning the policy used for updating does not correspond to the action selection rule. Rather, updates are always performed using $q_{s'a'} = \arg\max_{a''} q_{s'a''}$, i.e. updating is done with respect to the greedy policy. This implies that q-values may converge to the true, underlying value function *without* current, explorative behavior necessarily becoming optimal over time (e.g. trivially when using a static action selection rule such as the centroid). This illustrates that one should distinguish different notions of convergence in the setting of TD methods:

1. Convergence of q-values to the optimal values.

2. Convergence of q-values to those in accordance with current behavior.

3. Convergence of current policy to optimal policy.

While exploration is still going on, q-learning always achieves the first, and never the second. Thus, to meet the third – arguably the behavioral goal – exploration has to be decreased towards zero over time. In an ϵ-greedy policy, the ϵ has to be shrunk towards zero, or the λ used in logit choice increased towards infinity.[9] Proofs for 1. and 3. exist for suitably parametrized q-learning.

The contrast are *on-policy* methods that perform the update with respect to the same rule that is used for action selection. The main exemplar is *SARSA* ("state-action-reward-state-action"): Here, a' in updating simply corresponds to the action that is actually selected due to the current policy, i.e. which is in fact chosen next period by the agent. A refinement is *expected SARSA*, which removes the randomness that is under the agent's control from the updating step: Updating is performed using $q_{s'a'} = \sum_{a''} \sigma_{a''} q_{s'a''}$, where $\sigma_{a''}$ is the probability given to a'' in the resulting state under current policy and value estimates. In expectation and in the limit, both variants behave equivalently, but using the expected continuation demonstrably speeds up convergence in applications (van Seijen et al., 2009). For suitable α_t and a static action selection rule (e.g. logit with a fixed λ), SARSA meets 2., but not 1. in terms of convergence. However, if action selection is shifted towards a greedy policy over time (e.g. λ increased without bound), 3. and thus also 1. occur as well.

Arguably, reinforcement learning models are well-suited for an application in stochastic games, which already share the basic structure of an MDP. The literature so far however has focused predominantly on one-shot games. Between on- and off-policy algorithms, the former seem to us the more natural choice when studying dynamics in games; the reason is that off-policy algorithms such as q-learning incur the problem that players' behavior does not necessarily reveal in real time what they have learned so far.

The upcoming section will establish a close connection of logit Markov QRE to expected SARSA. SARSA and expected SARSA are tried and tested methods, and are often used in actual applications to solve problems of the given class (van Seijen et al., 2009; Zhang et al., 2011; Jiang et al., 2019; Kosana et al., 2022). Thus, the close connection of logit Markov QRE to reinforcement learning is not with regard to some obscure algorithm, perhaps even hand-picked to match, but rather to one of the main methods the field suggests for problems that are the single-player equivalent of finite stochastic games.

[9] Of course, this is assuming the environment is known to be static; in a changing environment, maintaining a degree of exploration will be desirable.

2.6.2 Logit Markov QRE and expected SARSA

We now establish a connection of logit Markov QRE and a specific form of expected SARSA. First, we establish that it is possible to reformulate logit Markov QRE in terms of q-values rather than $\boldsymbol{\sigma}$ and $\boldsymbol{V}$. This will make it straightforward to relate the two concepts later on.

Denote as $\boldsymbol{q} \in \mathbb{R}^{|A|}$ a vector of reals, with one entry q_{sia} for each action $a \in A_{si}$ of any player i in any state s. Furthermore, we will need a way to map these into strategies; for a given precision parameter λ, define the logit choice function $\boldsymbol{\sigma} : \mathbb{R}^{|A|} \to \mathbb{R}^{|A|}$ with components

$$\sigma_{sia}(\boldsymbol{q}) = \frac{\exp(\lambda q_{sia})}{\sum_{a' \in A_{si}} \exp(\lambda q_{sia'})}$$

As before with mixed strategies, we denote collections of the components of $\boldsymbol{\sigma}$ by expressions like $\boldsymbol{\sigma}_{si}$, $\boldsymbol{\sigma}_{\text{-}i}$, and so on. We can now obtain the following result.

Theorem 14. ***q-value-representation of logit Markov equilibrium.***
(i) Suppose for given $\lambda \in \mathbb{R}_0^+$, $\boldsymbol{q} \in \mathbb{R}^{|A|}$ is a solution to

$$q_{sia} = u_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}(\boldsymbol{q})) + \delta_i \sum_{s' \in S} \phi_{s \to s'}(a, \boldsymbol{\sigma}_{s,\text{-}i}(\boldsymbol{q})) \sum_{a' \in A_{s'i}} \sigma_{s'ia'}(\boldsymbol{q})\, q_{s'ia'} \tag{2.7}$$

for all s, i, and $a \in A_{si}$. Then $\boldsymbol{\sigma}(\boldsymbol{q})$ is a logit Markov QRE with the same precision parameter λ.
(ii) Conversely, if $\boldsymbol{\sigma}$ is a logit Markov QRE, then there exists a vector $\boldsymbol{q} \in \mathbb{R}^{|A|}$ which satisfies $\boldsymbol{\sigma}(\boldsymbol{q}) = \boldsymbol{\sigma}$ and equations (2.7).

Proof. (i) Set $V_{si} = \sum_{a \in A_{si}} \sigma_{sia}(\boldsymbol{q})\, q_{sia}$ and plug into (2.7) to obtain

$$q_{sia} = u_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}(\boldsymbol{q})) + \delta_i \sum_{s' \in S} \phi_{s \to s'}(a, \boldsymbol{\sigma}_{s,\text{-}i}(\boldsymbol{q}))\, V_{s'i} \tag{2.8}$$

By equation (2.2c), this identifies q_{sia} with $U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}(\boldsymbol{q}))$. From the definition of $\boldsymbol{\sigma}(\boldsymbol{q})$, equation (2.2a) then follows immediately. (2.2b) is obtained by multiplying each equation (2.8) with the corresponding $\sigma_{sia}(\boldsymbol{q})$ and then summing over all $a \in A_{si}$.
(ii) Let $\boldsymbol{V}$ be the state-values associated with $\boldsymbol{\sigma}$. Set $q_{sia} = U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)$. Then,

$\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{q})$ follows from the definition of logit Markov QRE (equation 2.2a). Plugging into (2.7) yields

$$\begin{aligned} U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) &= u_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}) + \delta_i \sum_{s'} \phi_{s \to s'}(a, \boldsymbol{\sigma}_{s,\text{-}i}) \sum_{a' \in A_{s'i}} \sigma_{s'ia'}\, U_{s'i}(a', \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) \\ &= u_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}) + \delta_i \sum_{s'} \phi_{s \to s'}(a, \boldsymbol{\sigma}_{s,\text{-}i}) U_{s'i}(\boldsymbol{\sigma}_{si}, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) \\ &= u_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}) + \delta_i \sum_{s'} \phi_{s \to s'}(a, \boldsymbol{\sigma}_{s,\text{-}i}) V_{s'i} \end{aligned}$$

which holds by equation (2.2c). ■

We now turn to derive a form of game dynamics from expected SARSA. We will assume that all players of the game play and learn according to expected SARSA, using a logit choice rule with a fixed and identical precision parameter λ, which we will denote by $\boldsymbol{\sigma}(\boldsymbol{q})$ as introduced above. Thus, every player tracks a set of q-values, $(q_{sia})_{s \in S, a \in A_{si}}$. Players do this separately, possibly without being aware that other players even exist. As mentioned earlier, SARSA is model-free, and learning does not involve forming a representation or estimate of u or ϕ. In particular, players do not track the strategies nor the learning process of the others in any way.

This induces the following updating process. Suppose that the current state is s, and players happen to choose action profile $\boldsymbol{a}_s$. A subsequent state s' is drawn (with probabilities $\phi_{s \to s'}(a_s)$) and observed by all players. Each player then updates the q-value of the chosen action a_{si} according to the update rule of expected SARSA:

$$q^{t+1}_{sia} = (1 - \alpha_t) q^t_{sia} + \alpha_t \left(u_{si}(\boldsymbol{a}_s) + \delta_i \sum_{a' \in A_{s'i}} \sigma_{s'ia'}(\boldsymbol{q})\, q^t_{s'ia'} \right)$$

All other entries of $\boldsymbol{q}$ are unchanged, i.e. $q^{t+1} = q^t$. The process then repeats, this time in state s'. If the game has terminal states, we assume play to be organized into episodes, meaning the game restarts from an initial state after termination. Otherwise, players will just follow this process indefinitely.

The above process induces a Markov chain in discrete time, with state variables given by $\boldsymbol{q}^t \in \mathbb{R}^{|A|}$, $s^t \in S$, and α_t (if this update parameter is chosen to be time-varying; we will comment on α later). The expectation of q^{t+1}_{sia}, conditional on action a being chosen by i, is

$$\mathbb{E}(q^{t+1}_{sia}|a \text{ chosen}) = (1-\alpha_t)q^t_{sia}+\alpha_t \left(u_{si}(a, \sigma_{s,\text{-}i}) + \delta_i \sum_{s' \in S} \phi_{s \to s'}(a, \sigma_{s,\text{-}i}) \sum_{a' \in A_{s'i}} \sigma_{s'ia'}(\boldsymbol{q})\, q^t_{s'ia'} \right)$$

One can obtain the unconditional expectation by simply weighting the above update with σ_{sia}:

$$\mathbb{E}(q^{t+1}_{sia}) = \sigma_{sia}\left(\boldsymbol{q}^t\right)\, \mathbb{E}\left(q^{t+1}_{sia}|a \text{ chosen}\right) + \left(1 - \sigma_{sia}(\boldsymbol{q}^t)\right)\, q^t_{sia}$$

One can quickly see that $\mathbb{E}(q^{t+1}_{sia}) = q^t_{sia}$ for all $s, i, a \in A_{si}$ if and only if $\boldsymbol{\sigma}(\boldsymbol{q})$ is a logit Markov QRE. After setting $\mathbb{E}(q^{t+1}_{sia}) = q^t_{sia}$ in the equation above, it simplifies to equation (2.7), and the claim follows directly from Theorem 14.

Thus, logit Markov QRE figures as those points of the Markov chain that are stationary in expectation. However, it is often inconvenient to deal with continuous-state, discrete time chains such as the one considered here. To circumvent this, it is standard practice in the study of reinforcement learning algorithms to rely on the so-called *ODE method* (Borkar, 2008). The underlying idea is to establish that the discrete process can be approximated by a continuous one in the limit, and then study the latter, which often allows much stronger results. There are generally two prerequisites. The first concerns the update weights α_t, which must obey $\sum_t \alpha_t^2 < \infty$ and $\sum_t \alpha_t = \infty$. Intuitively, these conditions ensure that the update steps get continuously smaller, but not at a pace that would allow the process to converge simply due to the decrease in step size, rather than reaching an actual stationary point. A typical choice is $\alpha_t = {}^1\!/_t$. It is also possible to use a separate α for every individual state or even action. The other prerequisite is that every state is visited infinitely often and every action is used an infinite number of times. The latter is directly implied by the logit choice rule. For the former, some mild restrictions must be placed on ϕ, in particular on the transition matrix that arises from completely mixed strategy profiles (which logit choice always produces): Either, the game does not involve termination, and all states are connected. Or, the game terminates with probability one in finite time (which then starts a new episode), and every state is reachable from the possible initial states.

If the prerequisites are met, discrete stochastic updating processes can be represented by a system of ordinary differential equations. In our case, the system is given by

$$\dot{q}_{sia} = q_{sia} - \left(u\Big(a, \boldsymbol{\sigma}_{s,\text{-}i}(\boldsymbol{q})\Big) + \delta \sum_{s' \in S} \phi_{s \to s'}\Big(a, \boldsymbol{\sigma}_{s,\text{-}i}(\boldsymbol{q})\Big) \sum_{a' \in A_{s'i}} \sigma_{s'ia'}(\boldsymbol{q})\, q_{s'ia'} \right) \tag{2.9}$$

for every s, i, and $a \in A_{si}$. Note that for normal-form games, this dynamic reduces to the well-known logit dynamics (Alós-Ferrer and Netzer, 2010).

Theorem 15. ***Logit Markov QRE as stationary points.***
The set of logit Markov QRE exactly coincides with the set of stationary points of the dynamic given by (2.9).

Proof. For $\dot{q}_{sia} = 0$, equation (2.9) reduces to equation (2.7), which are necessary and sufficient for logit Markov QRE by Theorem 14. ■

This is the main result in the current section. A natural next step is to consider the stability of logit Markov QRE as stationary points. Unfortunately, we cannot offer such results yet.[10]

Still, the dynamic perspective in our view already gives additional credence to logit Markov QRE as a solution concept. In the usual interpretation of QRE, players still act strategically, i.e. form (accurate) beliefs about others' actions and react to expected payoffs, even if noisily. However, as shown here, logit Markov QRE remains a reasonable solution concept even under much weaker assumptions: The updating process that gives rise to the dynamic in (2.9) is purely mechanical, and players are required to track own actions and realized instantaneous utilities only, without modeling the game or other players at all.

Theorem 15 is also interesting for the homotopy interpretation of logit Markov QRE. In this section, we assumed λ to be a given and fixed parameter. However, one could imaging a meta-process, by which all players start at $\lambda = 0$ and then increase it continuously, while correcting the q-estimates according to (2.9) along the way.[11] If the former happens on a much slower time scale, the resulting

[10] As a minor exception, it is straightforward to show that the starting point at $\lambda = 0$ is always stable: Since the logit response consists uniform mixing independent of $\boldsymbol{q}$, there is essentially no interaction between the players, so that the convergence proofs from the single-agent case apply (van Seijen et al., 2009). By continuity this result then extends also to all logit Markov QRE with sufficiently small λ.

[11] Note that increasing λ is also generally the recommendation when using SARSA in practical applications, as only then the optimal policy is actually reached in the limit (Sutton and Barto, 2018).

trajectory should resemble following the principal branch of the homotopy, as described in Section 2.4 for computational purposes. An important complication is the existence of path segments along which λ has to be decreased to follow it (see Figure 2.1). It seems promising to us to study in detail what happens along these segments in terms of the dynamics. We suspect that generically, stability of the stationary points switches at the turning points in λ (so that the increasing segments are stable, the decreasing ones unstable). Continuously increasing λ would then lead to a mostly continuous evolution of strategies, but involve discontinuous, rapid changes whenever such a turning point is reached.

Finally, if the existence of stable equilibria can be established at least generically, the dynamic interpretation could also be used for computational purposes. An alternative to homotopy continuation from the starting point would then be to start at some higher λ and arbitrary $\boldsymbol{q}$, and either perform iterated discrete updates or integrate the system of ordinary differential equations from (2.9) until a logit Markov QRE is approximated with the desired precision. Once there, one can follow the homotopy branch from that point on in both directions to compute stationary equilibria. This might be more efficient computationally than pure homotopy continuation. Moreover, it would allow to reach branches of the QRE graph that are not connected to the starting point.

2.7 Conclusion

In this paper we have defined Markov quantal response equilibria and shown existence for all finite discounted stochastic games. We then studied the specific variant based on logit response, whose correspondence can be given a homotopy interpretation. As we demonstrate, the set of logit Markov QRE includes a uniquely defined principal branch, connecting the unique solution at $\lambda = 0$ to a specific limiting stationary equilibrium.

This result opens two avenues. First, the uniqueness of the principal branch suggests that it can be used as an equilibrium selection criterion. Second, numerical traversal of the principal branch can be used to efficiently compute at least one stationary equilibrium of any stochastic game. A ready-to-use implementation is available as part of the python package sgamesolver (Chapter 4). Subject to the usual limitations of numerical computation, it can solve any finite game falling under the broad class of finite stochastic games, with no requirements to its specific structure.

We further established two interesting properties of logit Markov QRE. The first is a bound for the maximum loss incurred by any player, compared to an optimal strategy, establishing it as ε-equilibrium. The bound is independent of u and decreasing in λ. Second, we demonstrated a close connection of logit Markov QRE to reinforcement learning, which is readily applicable to the structure of stochastic games. In particular, logit Markov QRE coincide with the stationary points of a dynamic we derived from the tried-and-tested learning algorithm expected SARSA. QRE is in general seen as a behavioral solution concept, where players are assumed to act boundedly rational, but still model their surroundings as rational agents would. The results suggests that it remains interesting under much weaker assumptions, as the outcome of a learning process that has extremely minimal demands in terms of information processing. Studying this dynamic in more detail, in particular with respect to asymptotic stability, seems like a very promising avenue to us.

Appendix

2.A Homotopy Function for Computation

2.A.1 Homotopy Function

The original system of equations according to (2.4) reads

$$H^{\sigma}_{sia}(\boldsymbol{\sigma}, \boldsymbol{V}, \lambda) = \sigma_{sia} - \frac{\exp\left(\lambda U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)\right)}{\sum\limits_{a' \in A_{si}} \exp\left(\lambda U_{si}(a', \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)\right)} = 0 \qquad \forall s, i, a \in A_{si}$$

$$H^{V}_{si}(\boldsymbol{\sigma}, \boldsymbol{V}, \lambda) = V_{si} - u_{si}(\boldsymbol{\sigma}_s) - \delta_i \sum_{s' \in S} \phi_{s \to s'}(\boldsymbol{\sigma}_s)\, V_{s',i} = 0 \qquad \forall s, i,$$

It contains two types of equations: strategy equations and state value equations. The strategy equations are transformed as follows, analogously to Turocy (2005, 2010). We will denote the first action of each agent as a_0 and the according probability by σ_{si0}. First, for each agent (s, i), the strategy equations for actions $a \neq a_0$ are divided by the corresponding equation of action $a = 0$ so that the denominators cancel:

$$\frac{\sigma_{sia>0}}{\sigma_{si0}} \;=\; \exp\left(\lambda U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) - \lambda U_{si}(a_0, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)\right).$$

Secondly, logarithmizing removes the remaining exponential :

$$\log(\sigma_{sia>0}) - \log(\sigma_{si0}) \;=\; \lambda\Big(U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) - U_{si}(a_0, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)\Big)$$

Finally, instead of the by now trivial strategy equations for actions $a = 0$, normalization equations are introduced, ensuring that probabilities sum up to one:

$$\sum_{a \in A_{si}} \sigma^{*}_{sia} \;=\; 1.$$

The state value equations are rewritten using the shorthand

$$U_{si}(\boldsymbol{\sigma}_s) = u_{si}(\boldsymbol{\sigma}_s) + \delta_i \sum_{s' \in S} \phi_{s \to s'}(\boldsymbol{\sigma}_s)\, V_{s',i}$$

Finally, for numerical purposes, it is preferable to work with logarithmized strategies to avoid blow-up in the Jacobian (Turocy, 2005, 2010). Applying the substitution $\beta_{sia} := \log(\sigma_{sia})$, but still writing $\sigma_{sia} = e^{\beta_{sia}}$ for better readability, the transformed system of strategy and state value equations is given by

$$\begin{aligned}
\tilde{H}^{\sigma}_{si0}(\boldsymbol{\beta}, \boldsymbol{V}, \lambda) &= 1 - \sum_{a \in A_{si}} \sigma_{sia} = 0 \\
\tilde{H}^{\sigma}_{sia>0}(\boldsymbol{\beta}, \boldsymbol{V}, \lambda) &= \lambda\Big(U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) - U_{si}(a_0, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)\Big) - \Big(\beta_{sia} - \beta_{si0}\Big) = 0 \\
\tilde{H}^{V}_{si}(\boldsymbol{\beta}, \boldsymbol{V}, \lambda) &= -V_{si} \ + \sum_{a \in A_{si}} \sigma_{sia}\, U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) = 0
\end{aligned}$$

2.A.2 Jacobian Matrix

The components of the Jacobian matrix

$$J(\boldsymbol{\beta}, \boldsymbol{V}, \lambda) = \begin{pmatrix} \frac{\partial H^{\sigma}_{sia}(\boldsymbol{\beta},\boldsymbol{V},\lambda)}{\partial \beta_{s',i',a'}}, & \frac{\partial H^{\sigma}_{sia}(\boldsymbol{\beta},\boldsymbol{V},\lambda)}{\partial V_{s',i'}}, & \frac{\partial H^{\sigma}_{sia}(\boldsymbol{\beta},\boldsymbol{V},\lambda)}{\partial \lambda} \\ \frac{\partial H^{V}_{si}(\boldsymbol{\beta},\boldsymbol{V},\lambda)}{\partial \beta_{s',i',a'}}, & \frac{\partial H^{V}_{si}(\boldsymbol{\beta},\boldsymbol{V},\lambda)}{\partial V_{s',i'}}, & \frac{\partial H^{V}_{si}(\boldsymbol{\beta},\boldsymbol{V},\lambda)}{\partial \lambda} \end{pmatrix}$$

are given as follows.

Partial derivatives of H^{σ}:

$$\frac{\partial H^{\sigma}_{si0}(\boldsymbol{\beta}, \boldsymbol{V}, \lambda)}{\partial \beta_{s'i'a'}} = \begin{cases} -\sigma_{sia'} & \text{if } s' = s \text{ and } i' = i, \\ 0 & \text{else,} \end{cases}$$

$$\frac{\partial H^{\sigma}_{si0}(\boldsymbol{\beta}, \boldsymbol{V}, \lambda)}{\partial V_{s'i'}} = 0,$$

$$\frac{\partial H^{\sigma}_{si0}(\boldsymbol{\beta}, \boldsymbol{V}, \lambda)}{\partial \lambda} = 0,$$

$$\frac{\partial H^{\sigma}_{sia>0}(\boldsymbol{\beta}, \boldsymbol{V}, \lambda)}{\partial \beta_{s',i',a'}} = \begin{cases} 1 & \text{if } s' = s, i' = i, a' = 0, \\ -1 & \text{if } s' = s, i' = i, a' > 0, \\ \lambda \left[\frac{\partial U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)}{\partial \beta_{s'i'a'}} - \frac{\partial U_{si}(a_0, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)}{\partial \beta_{s',i',a'}} \right] & \text{if } i' \neq i, \\ 0 & \text{else,} \end{cases}$$

$$\frac{\partial H^{\sigma}_{sia>0}(\boldsymbol{\beta}, \boldsymbol{V}, \lambda)}{\partial V_{s'i'}} = \lambda \left[\frac{\partial U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)}{\partial V_{s'i'}} - \frac{\partial U_{si}(a_0, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)}{\partial V_{s'i'}} \right],$$

$$\frac{\partial H^{\sigma}_{sia>0}(\boldsymbol{\beta}, \boldsymbol{V}, \lambda)}{\partial \lambda} \;=\; U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) - U_{si}(a_0, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i).$$

Partial derivatives of H^V:

$$\frac{\partial H^V_{si}(\boldsymbol{\beta}, \boldsymbol{V}, \lambda)}{\partial \beta_{s'i'a'}} \;=\; \begin{cases} \sigma_{sia'} \cdot U_{si}(a', \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) & \text{if } s' = s, i' = i, \\ \sum\limits_{a'' \in A_{si}} \sigma_{sia''} \cdot \frac{\partial U_{si}(a'', \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)}{\partial \beta_{si'a'}} & \text{if } s' = s, i' \neq i, \\ 0 & \text{else,} \end{cases}$$

$$\frac{\partial H^V_{si}(\boldsymbol{\beta}, \boldsymbol{V}, \lambda)}{\partial V_{s',i'}} \;=\; \begin{cases} -1 + \sum\limits_{a'' \in A_{si}} \sigma_{sia''} \cdot \frac{\partial U_{si}(a'', \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)}{\partial V_{si}} & \text{if } s' = s, i' = i, \\ \sum\limits_{a'' \in A_{si}} \sigma_{sia''} \cdot \frac{\partial U_{si}(a'', \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)}{\partial V_{s',i}} & \text{if } s' \neq s, i' = i, \\ 0 & \text{else,} \end{cases}$$

$$\frac{\partial H^V_{si}(\boldsymbol{\beta}, \boldsymbol{V}, \lambda)}{\partial \lambda} \;=\; 0.$$

Underlying partial derivatives of expected payoffs:

$$\frac{\partial U_{sia''}(a'', \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)}{\partial \beta_{s'i'a'}} \;=\; \begin{cases} \sum\limits_{\substack{\boldsymbol{a}_{s,\text{-}i} \in A_{s,\text{-}i} \\ a_{si'} = a'}} \prod\limits_{\substack{i'' \in I \\ i'' \neq i}} \sigma_{si'', a_{s,i''}} \cdot U_{si}(a'', \boldsymbol{a}_{s,\text{-}i}, \boldsymbol{V}_i) & \text{if } s' = s, i' \neq i, \\ 0 & \text{else,} \end{cases}$$

$$\frac{\partial U_{si}(a'', \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)}{\partial V_{s',i'}} \;=\; \begin{cases} \sum\limits_{\boldsymbol{a}_{s,\text{-}i} \in A_{s,\text{-}i}} \prod\limits_{\substack{i'' \in I \\ i'' \neq i}} \sigma_{s,i'', a_{s,i''}} \cdot \delta_i \cdot \phi_{s \to s'}(a'', \boldsymbol{a}_{s,\text{-}i}) & \text{for } i' = i, \\ 0 & \text{else.} \end{cases}$$

References

Allgower, E. L. and K. Georg (1990): *Numerical Continuation Methods: An Introduction*, New York: Springer.

Alós-Ferrer, C. and N. Netzer (2010): "The logit-response dynamics," *Games and Economic Behavior*, 68, 413–427.

Basar, T. and G. J. Olsder (1999): *Dynamic Noncooperative Game Theory*, Philadelphia: Society for Industrial and Applied Mathematics.

Battaglini, M. and T. R. Palfrey (2012): "The dynamics of distributive politics," *Economic Theory*, 49, 739–777.

Bellman, R. (1954): "The Theory of Dynamic Programming," *Bulletin of the American Mathematical Society*, 60, 503–515.

Borkar, V. (2008): *Stochastic Approximation: A Dynamical Systems Viewpoint*, Cambridge University Press.

Borkovsky, R. N., U. Doraszelski, and Y. Kryukov (2010): "A User's Guide to Solving Dynamic Stochastic Games Using the Homotopy Method," *Operations Research*, 58, 1116–1132.

Breitmoser, Y., J. H. Tan, and D. J. Zizzo (2010): "Understanding Perpetual R&D Races," *Economic Theory*, 44, 445–467.

Dang, C., P. J.-J. Herings, and P. Li (2022): "An Interior-Point Differentiable Path-Following Method to Compute Stationary Equilibria in Stochastic Games," *INFORMS Journal on Computing*, 34, 1403–1418.

Doraszelski, U. and J. F. Escobar (2010): "A Theory of Regular Markov Perfect Equilibria in Dynamic Stochastic Games: Genericity, Stability, and Purification," *Theoretical Economics*, 5, 369–402.

Filar, J. and K. Vrieze (1997): *Competitive Markov Decision Processes*, New York: Springer.

Fink, A. M. (1964): "Equilibrium in a Stochastic n-Person Game," *Journal of Science of the Hiroshima University, Series A-I (Mathematics)*, 28, 89–93.

Goeree, J. K., C. A. Holt, and T. R. Palfrey (2016): *Quantal Response Equilibrium: A Stochastic Theory of Games*, Princeton, New Jersey: Princeton University Press.

Govindan, S. and R. Wilson (2009): “Global Newton Method for Stochastic Games,” *Journal of Economic Theory*, 144, 414–421.

Harsanyi, J. C. (1975): “The Tracing Procedure: A Bayesian Approach to Defining a Solution for n-Person Noncooperative Games,” *International Journal of Game Theory*, 4, 61–94.

Harsanyi, J. C. and R. Selten (1988): *A General Theory of Equilibrium Selection in Games*, Cambridge, Massachusetts: MIT Press.

Herings, P. J.-J. and R. J. Peeters (2004): “Stationary Equilibria in Stochastic Games: Structure, Selection, and Computation,” *Journal of Economic Theory*, 118, 32–60.

——— (2010): “Homotopy Methods to Compute Equilibria in Game Theory,” *Economic Theory*, 42, 119–156.

Jiang, H., R. Gui, Z. Chen, L. Wu, J. Dang, and J. Zhou (2019): “An Improved Sarsa (λ) Reinforcement Learning Algorithm for Wireless Communication Systems,” *IEEE Access*, 7, 115418–115427.

Kosana, V., M. Santhosh, K. Teeparthi, and S. Kumar (2022): “A novel dynamic selection approach using on-policy SARSA algorithm for accurate wind speed prediction,” *Electric Power Systems Research*, 212, 108174.

Luce, R. D. (1959): *Individual Choice Behavior*, New York: Wiley.

Marker, D. (1996): “Model Theory and Exponentiation,” *Notices of the AMS*, 43, 753–759.

Maskin, E. and J. Tirole (2001): “Markov Perfect Equilibrium: I. Observable Actions,” *Journal of Economic Theory*, 100, 191–219.

McFadden, D. L. (1973): “Conditional Logit Analysis of Qualitative Choice Behavior,” in *Frontiers in Econometrics*, ed. by P. Zarembka, New York: Academic Press.

McKelvey, R. D. and T. R. Palfrey (1995): “Quantal Response Equilibria for Normal Form Games,” *Games and Economic Behavior*, 10, 6–38.

——— (1998): "Quantal Response Equilibria for Extensive Form Games," *Experimental Economics*, 1, 9–41.

Mertens, J.-F., S. Sorin, and S. Zamir (2015): *Repeated Games*, Cambridge: Cambridge University Press.

Pakes, A. and P. McGuire (1994): "Computing Markov Perfect Nash Equilibria: Numerical Implications of a Dynamic Differentiated Product Model," *RAND Journal of Economics*, 25, 555–589.

——— (2001): "Stochastic Algorithms, Symmetric Markov Perfect Equilibrium, and the Curse of Dimensionality," *Econometrica*, 69, 1261–1281.

Shapley, L. S. (1953): "Stochastic Games," *Proceedings of the National Academy of Sciences*, 39, 1095–1100.

Sobel, M. J. (1971): "Non-Cooperative Stochastic Games," *Annals of Mathematical Statistics*, 42, 1930–1935.

Sutton, R. S. and A. G. Barto (2018): *Reinforcement learning: An introduction*, Cambridge, MA: MIT press.

Takahashi, M. (1964): "Equilibrium Points of Stochastic, Noncooperative n-Person Games," *Journal of Science of the Hiroshima University, Series A-I (Mathematics)*, 28, 95–99.

Turocy, T. L. (2005): "A Dynamic Homotopy Interpretation of the Logistic Quantal Response Equilibrium," *Games and Economic Behavior*, 51, 243–263.

——— (2010): "Computing Sequential Equilibria Using Agent Quantal Response Equilibria," *Economic Theory*, 42, 255–269.

van Seijen, H., H. van Hasselt, S. Whiteson, and M. Wiering (2009): "A theoretical and empirical analysis of Expected Sarsa," in *2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, 177–184.

Zangwill, W. I. and C. B. Garcia (1981): *Pathways to Solutions, Fixed Points, and Equilibria*, Upper Saddle River, New Jersey: Prentice-Hall.

Zhang, Z., L. Zheng, F. Hou, and N. Li (2011): "Semiconductor final test scheduling with Sarsa(λ,k) algorithm," *European Journal of Operational Research*, 215, 446–458.

Chapter 3

The Logarithmic Stochastic Tracing Procedure

This chapter is based on joint work with Steffen Eibelshäuser, Victor Klockmann, and Alicia von Schenk. It has been published in the Informs Journal on Computing (online August 2023, print in preparation).

Abstract: We introduce the logarithmic stochastic tracing procedure, a homotopy method to compute stationary equilibria for finite and discounted stochastic games. We build on the linear stochastic tracing procedure (Herings and Peeters 2004), but introduce logarithmic penalty terms as a regularization device, which brings two major improvements. First, the scope of the method is extended: it now has a convergence guarantee for all games of this class, rather than just generic ones. Second, by ensuring a smooth and interior solution path, computational performance is increased significantly. A ready-to-use implementation is publicly available. As demonstrated here, its speed compares quite favorable to other available algorithms, and it allows to solve games of considerable size in reasonable times. Because the method involves the gradual transformation of a prior into equilibrium strategies, it is possible to search the prior space and uncover potentially multiple equilibria and their respective basins of attraction. This also connects the method to established theory of equilibrium selection.

3.1 Introduction

Many economic decisions are both strategic and dynamic: Agents interact repeatedly, and their present decisions not only jointly determine immediate payoffs, but also shape the strategic situation in future periods. Examples include dynamic pricing, the accumulation and depletion of resources, savings or capacities, entering of legal contracts, and the development of new technologies.

The interplay of strategic and dynamic considerations is adequately captured in the class of stochastic games, first introduced by Shapley (1953). Stochastic games generalize both Markov decision processes (by introducing multiple players) and repeated games (by introducing different states of the world), and thus provide a rich framework for the analysis of dynamic interaction. Following Shapley, the most prominent solution concept for discounted stochastic games is stationary equilibrium, which is a subgame-perfect equilibrium in history- and time-independent (but state-dependent) strategies. It is well-known that every finite discounted stochastic game admits a stationary equilibrium (Shapley, 1953; Fink, 1964; Takahashi, 1964).

Solan and Vieille (2015) recently appraised Shapley's contribution and the subsequent theoretical developments. In closing, the authors remark: "Although our understanding of dynamic situations has improved, the questions that we can answer are still limited, and the models that are analyzed are still very stylistic. New tools must be developed so that we can treat models that are closer to real-life situations and provide better predictions." We agree, and argue in particular that stochastic games hold great potential for applied economics. Situations involving dynamic-strategic interplay are ubiquitous, in diverse areas such as pricing and contracting in industrial organization, competition in research and development, or the market microstructure of institutions such as limit order books. Yet, so far stochastic games have found only limited use as a modeling device.[1] A likely reason is that they come with two major difficulties, even when focusing on stationary strategies. First, equilibria are generally hard to compute (Gilboa and Zemel, 1989). Finding Nash equilibria even in normal-form games has been shown to be PPAD-complete (Daskalakis et al., 2009). In stochastic games in particular, even a moderate number of states quickly lead to a large system of nonlinear equations which characterize equilibria. Markov decision problems, which exhibit a similar

[1] Some notable exceptions include Cournot competition with renewable resources (Levhari and Mirman, 1980), dynamic price competition (Maskin and Tirole, 1988a,b), industry dynamics (Ericson and Pakes, 1995), financial limit order markets (Goettler et al., 2005), and learning by doing (Besanko et al., 2010).

curse of dimensionality, are typically solved using iterative methods. Unfortunately, these are generally not applicable to stochastic games: In the presence of strategic interaction their convergence is not guaranteed. Second, equilibria are typically not unique. In this paper, we address both issues by proposing a solution method which, first, allows to compute a stationary equilibrium of any finite discounted stochastic game and, second, has ties to established equilibrium selection theory.

Our method has its roots in the linear tracing procedure for finite normal-form games by Harsanyi and Selten (1988), which is the basis of their theory of equilibrium selection. The linear tracing procedure augments the game in question with a set of prior beliefs as additional primitive, a strategy profile which can be understood as a first expectation of players' likely actions. The procedure then performs a gradual transformation of these priors into equilibrium beliefs. Specifically, a set of auxiliary games is defined using a homotopy parameter $t \in [0, 1]$. For $t = 0$, players maximize solely against their prior. For $t \in (0, 1)$, players maximize against a convex combination of priors and the best responses of other players. At $t = 1$, players maximize solely against others' best responses: Beliefs are consistent and an equilibrium of the original game is reached. Harsanyi and Selten (1988) interpret the linear tracing procedure as a form of Bayesian strategic reasoning, in which all players start from a shared first expectation regarding others' behavior and then gradually feed in second order information on others' rational response, until beliefs are in equilibrium.

Mathematically, the linear tracing procedure falls into the general class of homotopy methods, in which a complex problem, here finding an equilibrium, is continuously transformed into a related, but much simpler problem, here finding solutions to a set of decision problems at $t = 0$.[2] This transformation is then gradually reversed, while tracing a so-called homotopy path of solutions until a solution for the original problem at $t = 1$ is reached. In the linear tracing procedure and its descendants, this path is constructed from equilibria of the auxiliary games.

The linear tracing procedure was generalized to finite discounted stochastic games by Herings and Peeters (2003, 2004), on which our method is directly based. Both the original and the stochastic linear tracing procedures share an important limitation: They are guaranteed to be well-defined only for generic

[2] A general, thorough, and accessible introduction to homotopy methods is given by Zangwill and Garcia (1981). Eaves and Schmedders (1999) offer a concise introduction specifically geared towards economists.

games. This is a considerable restriction since most, if not virtually all games studied in economics are non-generic.[3] For these games, the solution set of the linear methods may include multiple starting points, branching points, or manifolds of dimension higher than one. The method then fails to define a unique, isolated, one-dimensional path, which poses a problem both for numerical computations and interpretation as a selection criterion. In light of this, Harsanyi (1975) and Harsanyi and Selten (1988) also discuss a logarithmic variant for normal-form games that addresses these issues, but agrees with the linear method in a limiting sense whenever the latter is well-defined. Moreover, the path used by the logarithmic procedure is smooth and interior, which eases numerical path following.

The present paper develops the logarithmic stochastic tracing procedure, which similarly extends the linear tracing procedure for stochastic games by Herings and Peeters (2004). We show that our logarithmic variant, in contrast to the linear procedure, is guaranteed to induce an isolated path for any finite discounted stochastic game, placing it in the class of *probability-one homotopies* (Watson, 2002) and overcoming the limitation to generic games. Nevertheless, it retains a close connection to the original, linear method. Whenever the linear method is well-behaved, both select the same equilibrium. When the solution set of the linear procedure fails to define a unique, one-dimensional path without branching points, the path induced by the logarithmic procedure is still *contained* in that set, and can thus be considered a selection from the multiple paths suggested by the linear procedure.

In principle, our approach of regularization via logarithmic costs which are then faded out resembles that of Harsanyi and Selten (1988) for normal form games. However, the resulting mathematical system for stochastic games differs in important respects, making the extension non-trivial and our proofs of convergence rely on quite different mathematical instruments.[4] At the same time, the proofs

[3] Almost all games are generic, in the sense that the set of generic games has full Lebesgue measure in the space of games. If payoff and transition matrices of a finite stochastic game were to be chosen at random from a continuous distribution, the resulting game would be generic with probability one. Nonetheless, restriction to generic games is a serious limitation, since games studied by economists are usually not picked at random, but constructed in some regular fashion. Consequently, games of interest almost always have properties that make them non-generic, for example symmetries between players, states or certain actions, payoffs that are regularly spaced, or transition matrices that contain zeros. If these properties are deemed important, having methods for non-generic games are essential: While adding a small perturbation makes a non-generic game generic, it also results in the loss of such symmetries and regularities.

[4] Notably, Harsanyi and Selten themselves were in many respects not rigorous in their treatment of the logarithmic procedure for normal-form games. Schanuel et al. (1991) address this,

turn out to be quite general, and depend only on very general properties of the resulting equations.[5] In consequence, they should be easily adaptable to economic systems other than stochastic games (e.g. games of incomplete information or general equilibrium) or other forms of regularization penalties besides logarithmic ones.

The path traced by our method can be understood as the continuous transformation of prior beliefs into a specific equilibrium, as Harsanyi and Selten (1988) already discussed in the context of their theory of equilibrium selection in normal form games. While we cannot offer an analogous, complete theory of selection for stochastic games, the method could likewise be used as a building block of one (see also the discussion by Herings and Peeters (2004) regarding the linear stochastic procedure, whose properties concerning selection are preserved by ours). For example, it allows to compute the equilibrium arising from a specific prior, e.g. uniform mixing, or some other prior that is particularly salient for the game in question.[6] Alternatively, by applying the method repeatedly over points sampled from the prior space, one can potentially uncover multiple equilibria and map out their basins of attraction.

The logarithmic regularization does not only guarantee regularity, but also has direct computational advantages: It ensures that the path is smooth and interior, so that it can easily be traced numerically, allowing to compute stationary equilibria in finite stochastic games even of considerable size. An implementation is publicly available as part of the python package sgamesolver by Eibelshäuser and Poensgen (2019, code at `github.com/davidpoensgen/sgamesolver`). Benchmark timings are reported in Section 3.6 and compare quite favorably to available alternatives. The fastest algorithm with comparable scope which we could identify is Dang et al. (2022, Table 4), who in turn report to outperform the linear tracing procedure. The largest games solved by them contain 5 states, 5 players, and 8

using theory of semi-algebraic sets. This approach is not feasible here: For stochastic games, the logarithmic penalty terms do not drop out in the first order conditions, due to their presence in the continuation values. Thus, the solution curve is not algebraic (see equation 3b). Our approach is therefore quite different: A set of weights is used to guarantee a one-dimensional, isolated path, making the method a *probability one homotopy* in the sense of Watson (2002).

[5] Our proofs of convergence predominantly require that the resulting system of equations allows a representation in exponential polynomials, enabling repeated use of Khovanski's theorem (see Propositions 5 and 8). This will hold for virtually all objective functions typically used in economic modeling, and a wide range of possible penalty functions.

[6] Understood in this way, the method shares some similarity with the level-k framework (Nagel, 1995; Stahl and Wilson, 1995), with the prior playing the same role as the level-0 strategy. A notable difference is of course that the present method is guaranteed to converge to an equilibrium, which is not the case for level-k reasoning.

actions per state and player, with average running times of over 27 000 seconds, or 7.5 hours. In contrast, our implementation of the logarithmic procedure requires only 49 seconds for this size – making it over 500 times faster – and also allows to solve much larger games in still reasonable time.[7]

As mentioned, computing stationary equilibria involves solving a high-dimensional, nonlinear system of equations. It is therefore not surprising that all currently available, practically suitable algorithms for arbitrary finite stochastic games are homotopy-based, to the best of our knowledge.[8] Apart from the linear tracing procedure by Herings and Peeters (2004), we are aware of three further such methods. First, Govindan and Wilson (2009) propose a global Newton method based on the structure theorem by Kohlberg and Mertens (1986). Second, Eibelshäuser and Poensgen (2020) propose a homotopy method based on quantal response equilibrium (McKelvey and Palfrey, 1995). Finally, Dang et al. (2022) propose an interior-point method from an arbitrary starting point. The latter two methods also work for all games. As mentioned earlier, two advantages of the logarithmic tracing procedure developed in this paper are its roots in equilibrium selection theory and its computational performance.

3.2 Stochastic Games, Stationary Strategies, and Equilibria

A stochastic game is played as follows. The initial state is determined, possibly according to a random distribution. At the beginning of each stage, all players learn the current state of the world and then choose one of their available actions in that state. (If a player has no decision to make in a certain state, the respective action set is a singleton.) Action profile and current state jointly determine instantaneous utilities for each player and a probability distribution from which a state for the next period is drawn. The next stage begins accordingly. A game

[7] Of course, this is a joint comparison of both algorithm and implementation. Unfortunately, it is not straightforward to compare the computational burden of different homotopy methods: The running times are mainly determined by how long, winded, and well-conditioned the solution paths are – properties for which it is hard to establish any general results.

[8] There also exist iteration-based algorithms following Pakes and McGuire (1994), but they come with no guarantee of convergence (which will thus depend on the specific game) and are in principle only suited to find pure-strategy equilibria (but see e.g. Doraszelski and Satterthwaite (2010) who use a purification technique to find equilibrium mixtures for certain decisions). On the other hand, these algorithms are able to handle quite large state spaces (Doraszelski and Judd, 2012). An extensive literature on industry dynamics has made use of these algorithms (see e.g. Ericson and Pakes, 1995; Doraszelski and Pakes, 2007; Abbring et al., 2018).

may involve terminal states, meaning it will end once such a state is reached; otherwise, the game will continue indefinitely. Players discount exponentially from period to period.

This class of games is quite general and nests for example normal form games, dynamic games with finite time horizon, repeated games (where the state space is a singleton), and Markov decision processes (which are one-player stochastic games). Formally, a stochastic game $\mathcal{G}$ is defined as a tuple $\left(S, I, A, \boldsymbol{u}, \boldsymbol{\Phi}, \boldsymbol{\Phi}_0, \boldsymbol{\delta}\right)$ with

S **:** set of states.

I**:** set of players.

A_{si}**:** action set of player i in state s. $A_s = \bigtimes_{i\in I} A_{si}$ is the set of action profiles in state s. $A = \bigcup_{s\in S, i\in I} A_{si}$ denotes the set of all actions of any player in any state (understood as a disjoint union). Thus, $|A|$ represents the total number of actions of the game. We often use the index ${}_{sia}$ to refer to an action a that belongs to player i in state s.

$\boldsymbol{u} = \left(u_{si}(\boldsymbol{a}_s)\right)_{\boldsymbol{a}_s\in A_s, s\in S, i\in I}$**:** instantaneous payoff functions $u_{si} : A_s \to \mathbb{R}$.

$\boldsymbol{\Phi} = \left(\phi_{s \to s'}(\boldsymbol{a}_s)\right)_{\boldsymbol{a}_s\in A_s, s, s'\in S}$**:** state transition probabilities, where $\phi_{s \to s'}(\boldsymbol{a}_s)$ denotes the probability of transitioning from state s to s', if action profile $\boldsymbol{a}_s$ is played.

$\boldsymbol{\Phi}_0 \in \Delta(S)$**:** probability distribution over the initial state s_0.

$\boldsymbol{\delta} = \left(\delta_i\right)_{i\in I}$**:** discount factors for all players.

Our method applies to all finite discounted stochastic games. A stochastic game is finite if S, I, and A are finite (while the time horizon is generally still infinite), and discounted if $\delta_i < 1$ for all i.

Throughout, we will limit all discussion to stationary behavior strategies. Such a strategy assigns to each pair (s, i), called the *agent* of player i in state s, a mixture $\boldsymbol{\sigma}_{si}$ over the available actions.[9] This means that strategies may condition on the current state, but neither on history of play beyond what is reflected in the

[9] We will denote by $\boldsymbol{\sigma}_{si} \in \Delta(A_{si})$ the mixture of an agent, by σ_{sia} the probability placed on a specific action $a \in A_{si}$, by $\boldsymbol{\sigma}_i$ the complete behavior strategy of player i, by $\boldsymbol{\sigma}_s$ the mixture of all agents in state s, and by $\boldsymbol{\sigma}$ the strategy profile of all players. As usual, index -i denotes a profile for all players but i.

current state, nor on time.[10] Limiting attention to stationary equilibria is quite conventional in the study of stochastic games, as the set of all equilibria is generally vast. Importantly, there always exist stationary best responses to stationary strategies. Placing a stationarity constraint on players' strategies therefore does not induce additional equilibria, but simply acts as a selection criterion. In addition, it is well-known that every finite discounted stochastic game has at least one stationary equilibrium. Herings and Peeters (2004) offer a more detailed exposition of behavior strategies, stationarity, and related matters in stochastic games. Note that by the conventional definition, stationary equilibria (e.g. Shapley, 1953; Takahashi, 1964; Fink, 1964) require optimality in all states (even those not actually reached in equilibrium), making them subgame-perfect. The well-known one-shot deviation principle applies to stochastic games, allowing the following characterization of stationary equilibria. A stationary strategy profile $\boldsymbol{\sigma}$ and an associated vector of state-player-values $\boldsymbol{V} \in \mathbb{R}^{|S \times I|}$ form an equilibrium if and only if, for all $(s, i) \in S \times I$:[11]

$$V_{si} = u_{si}(\boldsymbol{\sigma}_s) + \delta_i \sum_{s' \in S} \phi_{s \to s'}(\boldsymbol{\sigma}_s)\ V_{s'i} \tag{1a}$$

$$\sum_{a \in A_{si}} \sigma_{sia} = 1 \tag{1b}$$

$$\sigma_{sia} \geq 0 \qquad \forall a \in A_{si} \tag{1c}$$

$$\begin{aligned} & u_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}) + \delta_i \sum_{s' \in S} \phi_{s \to s'}(a, \boldsymbol{\sigma}_{s,\text{-}i})\ V_{s'i} \geq \\ & \qquad u_{si}(a', \boldsymbol{\sigma}_{s,\text{-}i}) + \delta_i \sum_{s' \in S} \phi_{s \to s'}(a', \boldsymbol{\sigma}_{s,\text{-}i})\ V_{s'i} \quad \vee \quad \sigma_{sia} = 0 \quad \forall a, a' \in A_{si} \end{aligned} \tag{1d}$$

Condition (a) simply requires that the values are consistent with $\boldsymbol{\sigma}$.[12] Conditions (b) and (c) are the usual constraints on mixed strategies. Finally, (d) rules out

[10] Of course it is always possible to model history-dependent strategies as stationary by introducing additional states. For example, augmenting the repeated prisoner's dilemma with the states "no defection yet" and "defection has occurred" with according transitions makes trigger a stationary strategy.

[11] Equilibrium is more commonly defined in terms of $\boldsymbol{\sigma}$ alone. However, including $\boldsymbol{V}$ is innocuous, as any $\boldsymbol{\sigma}$ uniquely determines $\boldsymbol{V}$ (see footnote 12), so that one could simply write $\boldsymbol{V}(\boldsymbol{\sigma})$ in place of $\boldsymbol{V}$. We use $\boldsymbol{\sigma}$ and $\boldsymbol{V}$ throughout because it greatly simplifies first-order conditions and also keeps the resulting equations closer to a numerical implementation of the algorithm.

[12] An expression for $\boldsymbol{V}$ as a function of $\boldsymbol{\sigma}$ can be recovered from conditions (1a) in vector notation. Enumerate states as $1, 2, ..., |S|$. Let $\boldsymbol{V}_i = (V_{1i}, V_{2i}, ..., V_{|S|i})^\top$, $\boldsymbol{u}_i = (u_{1i}(\boldsymbol{\sigma}), u_{2i}(\boldsymbol{\sigma}), ...u_{|S|i}(\boldsymbol{\sigma}))^\top$, and let $\boldsymbol{\Phi}$ be the matrix of state transition probabilities under $\boldsymbol{\sigma}$, so that $[\boldsymbol{\Phi}]_{s,s'} = \phi_{s \to s'}(\boldsymbol{\sigma})$. Now, equations (1a) for all agents of player i together read

$$\boldsymbol{V}_i = \boldsymbol{u}_i + \delta_i \boldsymbol{\Phi} \boldsymbol{V}_i$$

profitable one-shot deviations: It allows positive probability only on such actions which maximize total utility, given continuation values $\boldsymbol{V}$ for the next period. It can be expressed more succinctly by introducing the following notation

$$U_{si}(\boldsymbol{\sigma}_s, \boldsymbol{V}_i) := u_{si}(\boldsymbol{\sigma}_s) + \delta_i \sum_{s' \in S} \phi_{s \to s'}(\boldsymbol{\sigma}_s)\, V_{s'i}$$

so that U reflects total expected discounted utility when $\boldsymbol{\sigma}_s$ is played in the current period, and continuation values are given by $\boldsymbol{V}$. The condition then reads

$$U_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) \geq U_{si}(a', \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) \quad \vee \quad \sigma_{sia} = 0 \qquad \forall a, a' \in A_{si} \tag{1d'}$$

As mentioned before, this definition requires optimal play in all states, even those never reached in equilibrium. Stationary equilibria are therefore subgame-perfect and independent of the distribution over the initial state.

3.3 The Homotopy Function

3.3.1 Auxiliary Games

We take as given a finite stochastic game $\mathcal{G}$ and a prior vector $\boldsymbol{\rho} \in \bigtimes_{(s,i) \in S \times I} \Delta(A_{si})$, a mixed strategy profile that can be chosen freely.[13]A homotopy parameter $t \in [0, 1]$ is introduced to define a family $\mathcal{G}^t$ of auxiliary stochastic games, each of same dimensions as $\mathcal{G}$. In these games, all players i choose their strategy $\boldsymbol{\sigma}_i$ to maximize against a belief that others play according to $\boldsymbol{\rho}_{\text{-}i}$ with probability $(1-t)$ and according to $\boldsymbol{\sigma}_{\text{-}i}$ with probability t. The correlation structure of this belief is as follows. First, we assume correlation across opponents, so that the belief entails that either all or none of the other players follow $\boldsymbol{\rho}_{\text{-}i}$. (The alternative would have each individual opponent follow $\boldsymbol{\rho}_{\text{-}i}$ with probability $(1-t)$. Assuming correlation is in line with Harsanyi and Selten (1988), but nothing rests on it.) In stochastic games, one also needs to specify correlation across periods: One possibility is that opponents either follow or do not follow $\boldsymbol{\rho}_{\text{-}i}$ in all future stages, another

which is equivalent to

$$\boldsymbol{V}_i = (I - \delta_i \boldsymbol{\Phi})^{-1} \boldsymbol{u}_i = \sum_{t=0}^{\infty} (\delta_i \boldsymbol{\Phi})^t \boldsymbol{u}_i$$

The inverse $(I - \delta_i \boldsymbol{\Phi})^{-1}$ always exists, since $\boldsymbol{\Phi}$ is a transition matrix and $\delta_i < 1$.

[13] The assumption that priors $\boldsymbol{\rho}$ are shared by all players is in line with Harsanyi and Selten (1988); it also eases exposition. However, none of our results rely on it, and all could be obtained with one set of priors per player.

that this is resolved independently, period by period, making beliefs uncorrelated across time. We will assume the latter, as otherwise best responses in stationary strategies generally do not exist, as Herings and Peeters (2003) already argued convincingly.

According to these beliefs, one obtains from $\mathcal{G}$ and $\boldsymbol{\rho}$ transition probabilities

$$\bar{\phi}^t_{s \rightarrow s'}(\boldsymbol{\sigma}_s) := t\phi_{s \rightarrow s'}(\boldsymbol{\sigma}_{si}, \boldsymbol{\sigma}_{s,\text{-}i}) + (1-t)\phi_{s \rightarrow s'}(\boldsymbol{\sigma}_{si}, \boldsymbol{\rho}_{s,\text{-}i})$$

and instantaneous payoff functions

$$\bar{u}^t_{si}(\boldsymbol{\sigma}_s) := tu_{si}(\boldsymbol{\sigma}_{si}, \boldsymbol{\sigma}_{s,\text{-}i}) + (1-t)u_{si}(\boldsymbol{\sigma}_{si}, \boldsymbol{\rho}_{s,\text{-}i})$$

which so far correspond exactly to the auxiliary games used in the linear stochastic tracing procedure by Herings and Peeters (2004). Our logarithmic variant then further adds a logarithmic penalty term to instantaneous utilities:

$$\hat{u}^t_{si}(\boldsymbol{\sigma}_s) := \bar{u}^t_{si}(\boldsymbol{\sigma}_s) + (1-t)\eta \sum_{a \in A_{si}} \nu_{sia} \log(\sigma_{sia})$$

The penalties are weighted by $(1-t)$, and also by the scalar η and vector (ν_{sia}); the significance of these additional parameters is discussed below. To ensure the logarithmic terms are well-defined, players will be restricted to completely mixed strategies (with $\sigma_{sia} > 0$ for all s, i, a) as long as $t \in [0, 1)$ and $\eta > 0$. Note that for $t = 1$, penalties and terms depending on the prior drop out, so that $\hat{u}^1 = u$ and $\bar{\phi}^1 = \phi$, which corresponds to the original game. For $\eta = 0$, only the penalties drop out, so that $\hat{u}^t|_{\eta=0} = \bar{u}^t$, which again corresponds to the linear stochastic procedure. In both limiting cases, utilities are well-defined for pure strategies, and the restriction is not necessary.

One may interpret the logarithmic penalties as a form of control cost, which is minimized when mixing all actions and increases as a pure strategy is approached. Their main purpose however is to ensure that the system defines a homotopy path that is smooth and interior, and that all agents always have a unique best response.

Each penalty term is weighted by ν_{sia}, where $(\nu_{sia}) = \boldsymbol{\nu} \in \mathbb{R}^{|A|}_{>0}$ is a vector of parameters with one entry for each action of $\mathcal{G}$. The purpose behind these weights is to guarantee that the path is indeed regular, which is the case for any generic $\boldsymbol{\nu}$ as shown later on.[14] Multiplication by $(1-t)$ ensures that the penalty smoothly

[14] In the logarithmic procedure of Harsanyi and Selten (1988) for normal form games, the corresponding terms are given by $\nu_{ia} = \nu_i = \max u_i - \min u_i$ and simply serve to normalize the

fades out as t approaches 1, just as the influence of $\boldsymbol{\rho}$ does: It is easily seen that $\lim_{t\to 1}(\bar{\phi}^t, \hat{u}^t) = (\phi, u)$ pointwise (on the domain of completely mixed strategies, where $\hat{u}$ is well-defined).[15] The penalties are further weighted by η, a positive real number; once we have discussed properties for any given $\eta > 0$, we will consider behavior of the system as η goes to zero. Note that $\lim_{\eta\to 0}(\bar{\phi}^t, \hat{u}^t) = (\bar{\phi}^t, \bar{u}^t)$ pointwise. This already suggests that the method of Herings and Peeters (2004) arises as a limiting case of the logarithmic procedure, a relationship that will be established formally in Section 3.5.

These auxiliary games $\mathcal{G}^t$ with $t < 1$ are not strictly speaking finite stochastic games as defined in the preceding section, because different players face different transition probabilities. Moreover, instantaneous utilities of the auxiliary games are not defined for pure strategies and are not linear in $\boldsymbol{\sigma}_{si}$. Nevertheless, one can just as well consider equilibria in stationary strategies.

3.3.2 Auxiliary Equilibria and the Homotopy Function H

In a stationary equilibrium of the auxiliary game $\mathcal{G}^t$, each player chooses $\boldsymbol{\sigma}_i$ such that in each state, total discounted utility is maximized given beliefs constructed from $(\boldsymbol{\sigma}_{\text{-}i}, \boldsymbol{\rho}_{\text{-}i}, t)$. Put formally, a set of strategies $\boldsymbol{\sigma}$ and an associated vector of state-player values $\boldsymbol{V}$ are an equilibrium of $\mathcal{G}^t$ if and only if they solve, for each agent $(s, i) \in S \times I$:

$$\underset{\boldsymbol{\sigma}_{si}}{\text{maximize}} \quad V_{si} \tag{2a}$$

$$\text{s.t.} \qquad V_{si} = \bar{u}^t_{si}(\boldsymbol{\sigma}_s) + \delta_i \sum_{s'\in S} \bar{\phi}^t_{s\to s'}(\boldsymbol{\sigma}_s)\; V_{s'i} + (1-t)\eta \sum_{a\in A_{si}} \nu_{sia} \log(\sigma_{sia}) \tag{2b}$$

$$\sum_{a\in A_{si}} \sigma_{sia} = 1 \tag{2c}$$

$$\sigma_{sia} > 0 \qquad \forall a \in A_{si} \tag{2d}$$

penalty. However, with such a specification the path is not necessarily well-defined for all games and priors (contrary to the claims of the authors).

[15] As a technical side note, $\hat{u}$ is not jointly continuous at points with at least one $\sigma_{sia} = 0$ and $(1-t)\eta = 0$: When approaching these points, there is no guarantee that $(1-t)\eta\log(\sigma_{sia}) \to 0$. However, for the algorithm this is unproblematic; in particular, it is perfectly capable to compute equilibria involving some $\sigma_{sia} = 0$. First, on the equilibrium sets used by the method, σ_{sia} are bounded below by a function of $(1-t)\eta$ which ensures convergence; see Appendix 3.A in the Online Supplement for details. In addition, the logarithmic terms actually cancel out in all expressions in the proofs of Propositions 7 and 10, which concern the behavior at $t = 1$ and $\eta = 0$ respectively.

Equilibrium exists for all t. As this will follow naturally as corollary of Proposition 6 later on, we omit direct proof here.[16]

The logarithmic stochastic tracing procedure consists of tracing a curve of equilibria starting at $t = 0$ until an equilibrium of the original game is reached at $t = 1$. This is done using the homotopy function $H(\boldsymbol{\sigma}, \boldsymbol{V}, t)$, which is derived from equation (2); see Appendix 3.B in the Online Supplement. Its domain reflects the restrictions $\sigma_{sia} > 0$ for $t < 1$ and $\sigma_{sia} \geq 0$ at $t = 1$:

$$H : \Big((0,1]^{|A|} \times \mathbb{R}^{|S\times I|} \times [0,1)\Big) \cup \Big([0,1]^{|A|} \times \mathbb{R}^{|S\times I|} \times \{1\}\Big) \to \mathbb{R}^{|A|+|S\times I|}$$

H has one component representing a sum-to-one-condition for each agent $(s, i) \in S\times I$:

$$H^{\sigma}_{si}(\boldsymbol{\sigma}, \boldsymbol{V}, t) \ := \sum_{a\in A_{si}} \sigma_{sia} \ - \ 1 \tag{3a}$$

For each agent $(s, i) \in S\times I$, H further has one component for each action $a \in A_{si}$:

$$\begin{aligned} H^{V}_{sia}(\boldsymbol{\sigma}, \boldsymbol{V}, t) \ &:= \ \sigma_{sia} \left(-V_{si} + \bar{u}^t_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}) + \delta_i \sum_{s'\in S} \bar{\phi}^t_{s\rightarrow s'}(a, \boldsymbol{\sigma}_{s,\text{-}i}) \ V_{s'i} \right) \\ &\qquad + \ (1-t)\eta \Big(\nu_{sia} + \sigma_{sia} \sum_{a'\in A_{si}} \nu_{sia'} \Big[\log(\sigma_{sia'}) - 1\Big] \Big) \\ &= \ \sigma_{sia} \Big(-V_{si} + \bar{U}^t_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) \Big) \\ &\qquad + \ (1-t)\eta \Big(\nu_{sia} + \sigma_{sia} \sum_{a'\in A_{si}} \nu_{sia'} \Big[\log(\sigma_{sia'}) - 1\Big] \Big) \end{aligned} \tag{3b}$$

The last equality above simply introduces the following shorthand notation:

$$\bar{U}^t_{si}(\boldsymbol{\sigma}_s, \boldsymbol{V}_i) := \bar{u}^t_{si}(\boldsymbol{\sigma}_s) + \delta_i \sum_{s'\in S} \bar{\phi}^t_{s\rightarrow s'}(\boldsymbol{\sigma}_s) \ V_{s'i}$$

H is chosen so that its zero set exactly coincides with the set of equilibria of all auxiliary games:

Proposition 1. *For $t \in [0, 1)$, $(\boldsymbol{\sigma}, \boldsymbol{V})$ is a stationary equilibrium of $\mathcal{G}^t$ if and only if $H(\boldsymbol{\sigma}, \boldsymbol{V}, t) = \boldsymbol{0}$.*

[16] Such proof could proceed as follows: (i) show that for some $\epsilon > 0$, any strategy with any $\sigma_{sia} < \epsilon$ is dominated, as the logarithmic penalty outweighs any direct utility from a. (ii) Apply Brouwer's fixed point theorem to the now compact strategy space $[\epsilon, 1]^{|A|}$.

Proof. Detailed proof is found in Appendix 3.B. There, H is derived from the maximization problems given by equation (2), so that $H = \mathbf{0}$ represents their first order conditions. The logarithmic penalty terms make the problems strictly concave. Thus, $H = \mathbf{0}$ is both necessary and sufficient for equilibrium. ■

At $t = 1$, $H = \mathbf{0}$ is still necessary but no longer sufficient for an equilibrium. As the logarithmic penalty terms vanish, the resulting equations (3b) require that all actions a played by an agent (s, i) with strictly positive probability ($\sigma_{sia} > 0$) yield the same total discounted payoff, namely V_{si}. This is clearly necessary for equilibrium, but not sufficient, as it allows strictly better actions to be played with zero probability. In fact, $H|_{t=1} = \mathbf{0}$ characterizes stationary points under replicator dynamics for the game $\mathcal{G}$. In addition to the actual equilibria, these include for example all pure strategy profiles. The existence of additional solutions at $t = 1$ is unproblematic for the method: The endpoint of the homotopy path will always be an equilibrium, as will be shown in Proposition 7.

Before discussing the homotopy path induced by H in the next section, we will briefly show that the equilibria of all $\mathcal{G}^t$ are bounded.

Proposition 2. *The set of equilibria of the games $\mathcal{G}^t$ for all $t \in [0, 1]$ is bounded. Equivalently, the zero set $H^{-1}(0)$ is bounded.*

Proof. Because $\boldsymbol{\sigma}$ is bounded, it suffices to establish a bound for $\boldsymbol{V}$. From a vector representation of (2b) one obtains (compare footnote 12):

$$\boldsymbol{V}_i \ = \ \hat{\boldsymbol{u}}_i^t(\boldsymbol{\sigma}) + \delta_i \boldsymbol{\Phi}^t(\boldsymbol{\sigma})\boldsymbol{V}_i \ = \ \Big(\boldsymbol{I} - \delta_i \boldsymbol{\Phi}^t(\boldsymbol{\sigma})\Big)^{-1} \hat{\boldsymbol{u}}_i^t(\boldsymbol{\sigma})$$

Recall that $\hat{u}$ includes both linear and logarithmic parts of instantaneous utility. Clearly, $\boldsymbol{V}_i$ must be bounded from above since $\hat{\boldsymbol{u}}_i^t(\boldsymbol{\sigma})$ is bounded from above and $\delta_i < 1$. An upper bound for all entries of $\boldsymbol{V}_i$ is then

$$V_{si} \leq \frac{1}{1 - \delta_i} \max_{s, \boldsymbol{\sigma}_s} \hat{u}_{si}^t(\boldsymbol{\sigma}_s)$$

While $\hat{\boldsymbol{u}}_i^t$ does not have a lower bound due to the logarithmic penalty terms, in equilibrium $\boldsymbol{V}_i$ will nevertheless be bounded from below. Intuitively, the players can always guarantee some finite utility for themselves. To see this, pick an arbitrary interior strategy $\boldsymbol{c}_i$ for player i, e.g. the centroid strategy with $c_{sia} = \frac{1}{|A_{si}|}$. Any solution to (2) must then have as lower bound

$$\begin{aligned}\boldsymbol{V}_i &= \max_{\boldsymbol{\sigma}_i} \Big(\boldsymbol{I} - \delta_i \boldsymbol{\Phi}^t(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_{\text{-}i})\Big)^{-1} \hat{\boldsymbol{u}}_i^t(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_{\text{-}i}) \\ &\geq \Big(\boldsymbol{I} - \delta_i \boldsymbol{\Phi}^t(\boldsymbol{c}_i, \boldsymbol{\sigma}_{\text{-}i})\Big)^{-1} \hat{\boldsymbol{u}}_i^t(\boldsymbol{c}_i, \boldsymbol{\sigma}_{\text{-}i}) > -\infty\end{aligned}$$

To obtain uniform bounds for $\boldsymbol{V}_i$ in any stationary equilibrium of all auxiliary games $\mathcal{G}^t$, simply take the maximum of the upper bound over $t \in [0,1]$, and likewise take the minimum of the lower bound over $t \in [0,1]$ and $\boldsymbol{\sigma}_{\text{-}i} \in [0,1]^{|A_{\text{-}i}|}$. Continuity and compactness ensure that both exist. ■

3.4 The Solution Path for Given $\eta > 0$

We will now show that the set of stationary equilibria of $\mathcal{G}^t$ always contains a unique, smooth, and isolated path connecting the unique equilibrium of $\mathcal{G}^0$ to an equilibrium of the original game $\mathcal{G}$.

It will be helpful to define the sets

$$Y := (0,1]^{|A|} \times \mathbb{R}^{|S \times I|} \times [0,1) \quad \text{and} \quad Y^1 := [0,1]^{|A|} \times \mathbb{R}^{|S \times I|} \times \{1\}$$

The solution set discussed in this section is then

$$Z := \Big\{(\boldsymbol{\sigma}, \boldsymbol{V}, t) \in Y \mid H(\boldsymbol{\sigma}, \boldsymbol{V}, t) = \boldsymbol{0}\Big\}$$

We will show that if $\boldsymbol{\nu}$ is suitably chosen, $\boldsymbol{0}$ is a regular value of H on Y, so that Z consists of isolated, smooth arcs only. Moreover, there is always a unique solution at $t = 0$, which is connected by one such path to an equilibrium of $\mathcal{G}$, which we will call *distinguished.* Tracing this path allows to compute the distinguished equilibrium numerically. Because the path depends on the priors, repeating the process with different priors usually allows to compute additional equilibria.

Proposition 3. *$H|_{t=0} = \boldsymbol{0}$ has a unique solution.*

Proof. Detailed proof is found in Online Appendix 3.C.1. Its idea is as follows: In $\mathcal{G}^0$, there is no strategic interaction, and each player faces a discounted Markov decision problem. It is straightforward to show that a unique vector $\boldsymbol{V}_i^0 \in \mathbb{R}^{|S|}$ of state values exists for each of these problems. Due to the logarithmic penalty terms, utility in each state is strictly concave in $\boldsymbol{\sigma}_{si}$, so that optimal policies of all players must also be unique. ■

This solution will be called the starting point, $(\boldsymbol{\sigma}^0, \boldsymbol{V}^0)$. Appendix 3.C.2 in the Online Supplement details how it can be computed efficiently.

Proposition 4. *For generic $\boldsymbol{\nu}$, $\mathbf{0}$ is a regular value of $H|_Y$.*

Proof. Detailed proof is found in Online Appendix 3.E. It proceeds as follows: The Jacobian of $H(\boldsymbol{\sigma}, \boldsymbol{V}, t, \boldsymbol{\nu})$ has full rank everywhere in Y. This allows application of a parametrized version of Sard's theorem (Chow et al., 1978, Theorem 2.1, p. 891). Consequently, the set of $\boldsymbol{\nu}$ for which $H|_Y = \mathbf{0}$ is regular has full Lebesgue measure in $\mathbb{R}^{|A|}_{>0}$. ■

In all that follows we will assume that $\boldsymbol{\nu}$ is generic.

Corollary 4.1. Due to the regularity of $H = \mathbf{0}$, the implicit function theorem is applicable at any point contained in Z, so that Z must consist of a collection of isolated paths and loops. In effect, each path or loop can be represented as a function in a single variable; and since H is real analytic, so are these functions. Regularity further allows application of the route-loop-theorem (Eaves and Schmedders, 1999, Theorem 1, p. 1264), which implies that these paths can not form spirals or have endpoints in the interior of Y.

Because Z is bounded (Proposition 2), no path can go off to infinity. The following proposition establishes that the paths cannot oscillate indefinitely either, but must eventually reach the boundary of Y at one of its two endpoints when followed in either direction.

Proposition 5. *For generic $\boldsymbol{\nu}$, all paths and loops contained in Z are of finite arc length.*

Proof. If $H = \mathbf{0}$ is regular on $Y \cup Y^1$, this follows immediately from Watson (2002, Theorem 2.3 (4), p. 788). However, if the game is not generic, it is well possible that H is not regular on Y^1. Nevertheless, arc lengths will still be finite by the following argument.

We proceed by showing that any path in Z has a finite number of turning points in any dimension. Let k represent any σ_{sia}, any V_{si}, or t; a turning point in variable k is a point in which a path changes direction in that dimension. A necessary condition for a turning point is $\det(J_{\text{-}k}) = 0$, where $J_{\text{-}k}$ is the square matrix obtained from the Jacobian $J(\boldsymbol{\sigma}, \boldsymbol{V}, t)$ by deleting the column which contains the partial derivatives with respect to k.[17] Remember that $\boldsymbol{\sigma}$, $\boldsymbol{V}$, and t along any path can be represented as a real analytic function of a single variable, namely path length (see Corollary 4.1). As compositions of real analytic functions, all sub-determinants are also real analytic in path length, so that the real analytic identity theorem applies: Each sub-determinant is either zero along the complete path (in which case the respective variable is constant on that path, and no turning points exist), or its zero set along the path consists of isolated points.

Applying a change of variables, it can be shown that in the latter case, the zero set must be finite. Namely, replace $\sigma_{sia} =: \exp(\beta_{sia})$ in H and J. Because $\sigma_{sia} > 0$, this transformation is a homeomorphism, and the zero sets we are interested in are topologically unchanged. It is readily seen that all components of H and all sub-determinants of J after this substitution are exponential polynomials in $(\boldsymbol{\beta}, \boldsymbol{V}, t)$.[18] Khovanski's theorem (Marker, 1996, p. 757) states that the zero set of any set of exponential polynomials consists of finitely many connected components. This applies to the systems given by $H = \mathbf{0}$ and $\det(J_{\text{-}k}) = 0$ for any k. Since turning points are elements of these zero sets, and they are isolated as shown above, Z contains at most a finite number of turning points in any direction k. Thus, any path in Z must have two endpoints on the boundary of Y. Furthermore, each path can be partitioned into a finite number of segments joining one of these endpoints, a finite sequence of turning points (possibly none), and the other endpoint. Likewise, all closed loops consist of a finite number of segments joining its turning points. Each of these segments is bounded in length, because Z is bounded (Proposition 2) and the segments themselves contain no turning points. It follows that total arc length of each path or loop is finite. ■

Proposition 6. *For generic $\boldsymbol{\nu}$, Z contains exactly one path that starts transversally at $(\boldsymbol{\sigma}^0, \boldsymbol{V}^0)$ and ends in a point in Y^1, called the distinguished path. Any other path contained in Z either connects two points in Y^1, or is a closed loop.*

Proof. By the previous proposition, any component of Z that is not a loop must be a path with two endpoints on the boundary of Y. Since H is in particular also regular at the starting point $(\boldsymbol{\sigma}^0, \boldsymbol{V}^0, 0)$, the route-loop theorem (Eaves and

[17] Online Appendix 3.F offers a brief sketch of the significance of these sub-determinants to readers less familiar with homotopy methods.

[18] Exponential polynomials in a set of variables may be defined recursively as follows: (i) All polynomials in these variables are exponential polynomials. (ii) Furthermore, if x, y are exponential polynomials, then xy, $x + y$, and e^x are also exponential polynomials. (iii) Only expressions obtainable from (i) and (ii) are exponential polynomials.

Schmedders, 1999, Theorem 1, p. 1264) ensures that a single path reaches the boundary at $t = 0$ transversally: This path cannot be a loop. Since that point is unique, the path cannot return to $t = 0$ and no other path can hit the boundary at $t = 0$. All σ_{sia} are either bound to $(0, 1)$ or constant and equal to 1 (if the respective action set is a singleton). Thus, no path can have an endpoint on these boundaries. Furthermore, all V_{si} are bounded in equilibrium, which was shown in Proposition 2. The path starting at $(\boldsymbol{\sigma}^0, \boldsymbol{V}^0)$ must therefore eventually reach Y^1. Any other path must be a closed loop, or eventually reach Y^1 when followed in either direction. ■

Corollary 6.1. Equilibrium exists for all $\mathcal{G}^t$. By Proposition 1, $(\boldsymbol{\sigma}, \boldsymbol{V})$ is an equilibrium if and only if $H(\boldsymbol{\sigma}, \boldsymbol{V}, t) = \mathbf{0}$, and the existence of the distinguished path implies that H has at least one zero for any $t \in [0, 1)$.

Corollary 6.2. Together, Propositions 5 and 6 imply that the number of paths and loops contained in Z is finite, by the following argument. Each path or loop must contain at least one turning point in t (the only exception may be the distinguished path), and it was shown using Khovanski's theorem that all paths together contain a finite number of turning points in t.

Finally, we establish that the distinguished path indeed leads to an equilibrium of $\mathcal{G}$. For $t < 1$, $H = \mathbf{0}$ is both necessary and sufficient for an equilibrium of $\mathcal{G}^t$. At $t = 1$, it is no longer sufficient (see the discussion following Proposition 1). However, the following proposition shows that all the endpoints at $t = 1$ of paths in Z are in fact equilibria.

Proposition 7. *If $(\boldsymbol{\sigma}^n, \boldsymbol{V}^n, t^n)$ is a sequence in Z with limit $(\boldsymbol{\sigma}^N, \boldsymbol{V}^N, 1)$, then $(\boldsymbol{\sigma}^N, \boldsymbol{V}^N)$ is an equilibrium of $\mathcal{G}$. Therefore, any path in Z that reaches the boundary at $t = 1$ must do so at an equilibrium of $\mathcal{G}$.*

Proof. Conditions for stationary equilibria of $\mathcal{G}$ were stated in equation (1) and are repeated here for better readability: $\boldsymbol{\sigma}^N, \boldsymbol{V}^N$ are an equilibrium if, for each $(s, i) \in S \times I$,

$$V_{si}^N = u_{si}(\boldsymbol{\sigma}_s^N) + \delta_i \sum_{s' \in S} \phi_{s \to s'}(\boldsymbol{\sigma}_s^N)\, V_{s'i}^N = U_{si}(\boldsymbol{\sigma}_s^N, \boldsymbol{V}_i^N) \tag{4a}$$

$$\sum_{a \in A_{si}} \sigma_{sia}^N = 1 \tag{4b}$$

$$\sigma_{sia}^N \geq 0 \qquad \forall a \in A_{si} \tag{4c}$$

$$U_{si}(a, \boldsymbol{\sigma}_{s,-i}^N, \boldsymbol{V}_i^N) \geq U_{si}(a', \boldsymbol{\sigma}_{s,-i}^N, \boldsymbol{V}_i^N) \quad \vee \quad \sigma_{sia}^N = 0 \qquad \forall a, a' \in A_{si} \tag{4d}$$

(4a) follows from each $(\boldsymbol{\sigma}^n, \boldsymbol{V}^n, t^n)$ satisfying constraint (2b), which is continuous, with $\lim_{t\to 1} \bar{U}^t = U$ pointwise on Y. Likewise, (4b) holds for all $\boldsymbol{\sigma}^n_{si}$ by (2c). $\boldsymbol{\sigma}^n > \mathbf{0}$ ensures (4c). Regarding (4d), for any pair of actions a, a' of any agent (s, i), consider the equation

$$\sigma^n_{sia'} H^V_{sia}(\boldsymbol{\sigma}^n, \boldsymbol{V}^n, t^n) - \sigma^n_{sia} H^V_{sia'}(\boldsymbol{\sigma}^n, \boldsymbol{V}^n, t^n) = 0$$

which must hold for all n because $(\boldsymbol{\sigma}^n, \boldsymbol{V}^n, t^n) \in Z$ implies $H(\boldsymbol{\sigma}^n, \boldsymbol{V}^n, t^n) = \mathbf{0}$. Spelling out this equation gives:

$$\sigma^n_{sia}\sigma^n_{sia'} \left(\bar{U}^{t^n}_{si}(a, \boldsymbol{\sigma}^n_{s,\text{-}i}, \boldsymbol{V}_i) - \bar{U}^{t^n}_{si}(a', \boldsymbol{\sigma}^n_{s,\text{-}i}, \boldsymbol{V}_i)\right) + (1 - t^n)\eta(\nu_{sia}\sigma^n_{sia'} - \nu_{sia'}\sigma^n_{sia}) = 0 \tag{5}$$

$t^n \to 1$ then implies

$$\lim_{n\to\infty} \sigma^n_{sia}\sigma^n_{sia'} \left(\bar{U}^{t^n}_{si}(a, \boldsymbol{\sigma}^n_{s,\text{-}i}, \boldsymbol{V}^n_i) - \bar{U}^{t^n}_{si}(a', \boldsymbol{\sigma}^n_{s,\text{-}i}, \boldsymbol{V}^n_i)\right) = 0$$

and at least one factor must go to zero. If

$$\lim_{n\to\infty} \left(\bar{U}^{t^n}_{si}(a, \boldsymbol{\sigma}^n_{s,\text{-}i}, \boldsymbol{V}^n_i) - \bar{U}^{t^n}_{si}(a', \boldsymbol{\sigma}^n_{s,\text{-}i}, \boldsymbol{V}^n_i)\right) = 0,$$

then (4d) holds immediately. If conversely and without loss of generality

$$\lim_{n\to\infty} \left(\bar{U}^{t^n}_{si}(a, \boldsymbol{\sigma}^n_{s,\text{-}i}, \boldsymbol{V}^n_i) - \bar{U}^{t^n}_{si}(a', \boldsymbol{\sigma}^n_{s,\text{-}i}, \boldsymbol{V}^n_i)\right) > 0, \tag{6}$$

(4d) is still satisfied unless

$$\lim_{n\to\infty} \sigma^n_{sia'} > 0 = \lim_{n\to\infty} \sigma^n_{sia} \tag{7}$$

However, if (6) and (7) both hold, the first and the second summand in (5) must be strictly positive for n sufficiently large. This is a contradiction. ■

Summary of the first main result. For any given game $\mathcal{G}$, any prior $\boldsymbol{\rho}$, parameter $\eta > 0$, and generic $\boldsymbol{\nu} \in \mathbb{R}^{|A|}_{>0}$, the solution set $Z = H^{-1}(0) \cap Y$ is well-structured in the sense that it consists solely of smooth, isolated paths of finite length. In particular, it always contains a specific curve L which connects the unique solution at $t = 0$, $(\boldsymbol{\sigma}^0, \boldsymbol{V}^0, 0)$, to a point $(\boldsymbol{\sigma}^*, \boldsymbol{V}^*, 1) \in Y^1$, called the distinguished equilibrium of $\mathcal{G}$. L is a real analytic curve, as it is implicitly defined by $H = \mathbf{0}$, which is real analytic and regular. Appendix 3.F in the Online Supplement gives an explicit construction of L.

Notably, following this path allows to efficiently compute at least one stationary equilibrium of any finite stochastic game. Since the path and thereby the distinguished equilibrium depend on the choice of $\boldsymbol{\rho}$, which is free, additional equilibria can potentially be found by searching the prior space. (However, there is no guarantee that all equilibria of a game can be found in this manner.) Computationally, the path can be traced using standard numerical continuation methods. Timings are reported in Section 3.6.

3.5 The Limiting Solution Curve as $\eta \to 0$

We now discuss the behavior of the solution curve as the logarithmic penalty terms in the auxiliary games are faded out. As before, a game $\mathcal{G}$, an arbitrary prior $\boldsymbol{\rho}$, and a generic vector of weights $\boldsymbol{\nu}$ are taken as given. In the previous section we have shown that fixing some $\eta > 0$ gives rise to a well-defined curve L^η. We will now study their limit[19]

$$L^0 := \lim_{\eta \to 0} L^\eta$$

As outlined in the introduction, the present method has a close relation to the linear stochastic tracing procedure (Herings and Peeters, 2004), which resembles the relation of the logarithmic to the linear tracing procedure for normal form games (Harsanyi and Selten, 1988). This will now be established formally.

The linear stochastic procedure constructs auxiliary games from a prior just as described in Section 3.3.1, however without the logarithmic penalties (corresponding to $\eta = 0$). For generic $\mathcal{G}$ and $\boldsymbol{\rho}$, it also guarantees existence of a unique starting point for $t = 0$, and of a piecewise algebraic path connecting this point to a stationary equilibrium of $\mathcal{G}$. However, if game or prior are not generic, the solution set of its defining equations may not be well-behaved: It may contain multiple starting points or uncountable sets thereof; its paths may contain branching points; and the solution set may contain manifolds of dimension higher than one. In these cases, the linear procedure is not well-defined: It will fail to select a unique equilibrium, and in particular the application of numerical continuation methods is generally not feasible.

In the preceding section it was shown that the present method removes these problems and is always well-defined. Nevertheless, our method retains a close

[19] To be technically precise, Proposition 8 will show convergence of the curves L^η up to parametrization (i.e. convergence of their images in Hausdorff distance). From a practical perspective, this subtlety is without importance.

connection to the linear procedure. In particular, the limiting curve of the logarithmic procedure, L^0, will always be contained in the solution set of the linear procedure. If the linear method is well-defined, meaning it contains a unique solution path, L^0 will be identical to this path. If on the other hand the linear solution set includes multiple starting points, branching points, or higher-dimensional sets, there will be a multitude of paths connecting $t = 0$ and $t = 1$. In this case L^0 can be considered a unique selection from these.

To proceed, we first establish existence and some properties of L^0.

Proposition 8. *For generic $\boldsymbol{\nu}$, $L^0 = \lim_{\eta \to 0} L^\eta$ exists.*

Proof. Because this proof is somewhat lengthy, it is presented in steps 8.1–8.5. We will consider H again, but this time taking η as an argument rather than a fixed parameter. Because an open domain will be needed, $Y \times (0, \infty)$ is padded by some $\epsilon > 0$ in the relevant directions to obtain

$$D := \ (0, 1+\epsilon)^{|A|} \times \mathbb{R}^{|S \times I|} \times (-\epsilon, 1) \times (0, \infty)$$

Now consider the zero set of the homotopy function H on D,

$$\tilde{Z} := \{(\boldsymbol{\sigma}, \boldsymbol{V}, t, \eta) \in D \mid H(\boldsymbol{\sigma}, \boldsymbol{V}, t, \eta) = 0\}$$

which again is well-structured in the following sense:

Lemma 8.1. *For generic $\boldsymbol{\nu}$, $\tilde{Z}$ is a smooth, 2-dimensional manifold.*

Detailed proof for this lemma is given in Online Appendix 3.E. It is based once more on parametrized Sard's theorem (Chow et al., 1978, Theorem 2.1, p. 891), which is applicable since the Jacobian of $H(\boldsymbol{\sigma}, \boldsymbol{V}, \boldsymbol{\nu}, t, \eta)$ is of full rank on D. (This follows almost immediately from Proposition 4, insofar adding a column for η cannot reduce the rank of the Jacobian.)

Lemma 8.2. *For any $\eta > 0$, $H|_{t=0}(\boldsymbol{\sigma}, \boldsymbol{V}) = \mathbf{0}$ defines a unique starting point $(\boldsymbol{\sigma}^0(\eta), \boldsymbol{V}^0(\eta))$. All starting points lie on a single connected component of $\tilde{Z}$.*

Existence and uniqueness were shown in Proposition 3. Connectedness follows from continuity in η, which can be shown as follows. Recall that for $t = 0$, each player simply solves a Markov decision problem with total discounted utilities $U_i(\boldsymbol{\sigma}_i, \boldsymbol{\rho}_{\text{-}i})$ continuous in η. Because the optimal policies $\boldsymbol{\sigma}^0$ are always unique, they must also depend continuously on η. The same then holds for $\boldsymbol{V}^0$, which is

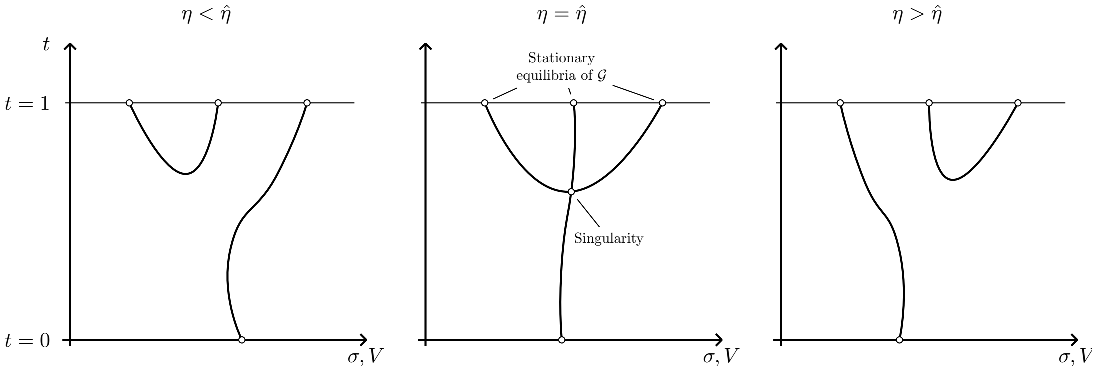


Figure 3.1: Example of a Pitchfork Bifurcation at a Critical Value of η. A singular point exists at $\hat{\eta}$, so that for this value, the curve L^η is not well-defined; for η smaller or larger, the singularity disappears. Note that the point in question is *not* a singularity of the 2-dimensional manifold $\tilde{Z}$, which by Lemma 8.1 is regular for generic $\boldsymbol{\nu}$.

a continuous function of $\boldsymbol{\sigma}^0$. (Note that this specific result does not require $\boldsymbol{\nu}$ to be generic.)

Lemma 8.3. *For all but finitely many $\eta \in (0, \infty)$, a well-defined curve L^η exists, with all properties discussed in Section 3.4. Hence, there exists $\bar{\eta} > 0$ so that L^η is well-defined for all $\eta \in (0, \bar{\eta})$.*

Before proving Lemma 8.3., we should briefly point out why it is necessary at all. In Section 3.4, it was shown that when fixing any η and generic $\boldsymbol{\nu}$, the curve L^η is always well-defined. However, if one then were to vary η continuously, while keeping $\boldsymbol{\nu}$ fixed – as this section intends – one may encounter some values for which the restricted zero set

$$Z^\eta := \left\{ (\boldsymbol{\sigma}, \boldsymbol{V}, t) \mid (\boldsymbol{\sigma}, \boldsymbol{V}, t, \eta) \in \tilde{Z} \right\}$$

is not regular (i.e. not a 1-dimensional manifold). For these, L^η need not be well-defined. The singular points generally have the character of pitchfork bifurcations, as sketched in Figure 3.1. However, the current lemma implies that this poses no problem to the question of convergence for $\eta \to 0$, as L^η is well-defined for all η sufficiently small.

We now turn to proving Lemma 8.3. Denote by $J_{\text{-}\eta,\text{-}k}$ the square matrix obtained from the Jacobian of H by removing those two columns corresponding to η and k, where k represents any σ_{sia}, any V_{si}, or t. Points in whose neighborhood

Z^η is not necessarily a 1-dimensional manifold are characterized by $H = \mathbf{0}$ and $\det(J_{\text{-}\eta,\text{-}k}) = 0$ for all k. The set of all such points will be denoted $S \subset \tilde{Z}$.

We will first show that the manifold $\tilde{Z}$ must be orthogonal to the η-axis in any point $(\hat{\boldsymbol{x}}, \hat{\eta}) := (\hat{\boldsymbol{\sigma}}, \hat{\boldsymbol{V}}, \hat{t}, \hat{\eta}) \in S$, or equivalently, the tangent space of $\tilde{Z}$ at that point is entirely contained in the hyperplane characterized by $\eta = \hat{\eta}$, which we will denote E. The tangent space is simply the kernel of the Jacobian J of H, so that the statement translates to $\ker(J(\hat{\boldsymbol{x}}, \hat{\eta})) \subseteq E$. Note that the restriction of the kernel to E must have dimension at least 2, i.e. $\dim(\ker(J(\hat{\boldsymbol{x}}, \hat{\eta}) \cap E)) \geq 2$ (if it had dimension 1, $\hat{\boldsymbol{x}}$ could not be a singular point of $Z^{\hat{\eta}}$). At the same time, $\tilde{Z}$ is a smooth 2-dimensional manifold everywhere on D (Lemma 8.1), so that $\dim(\ker(J(\hat{\boldsymbol{x}}, \hat{\eta}))) = 2$. Together, these dimensional requirements imply $\ker(J(\hat{\boldsymbol{x}}, \hat{\eta})) \subset E$, proving orthogonality.

Since $\tilde{Z}$ is a manifold, there exists a neighborhood of $(\hat{\boldsymbol{x}}, \hat{\eta})$ on $\tilde{Z}$ which is path-connected. By the preceding argument, the tangent space at any point in S is orthogonal to the η-axis; thus, any path on $\tilde{Z}$ joining $(\hat{\boldsymbol{x}}, \hat{\eta})$ to another point in S must either remain in the hyperplane characterized by $\eta = \hat{\eta}$, or leave S. In consequence, each connected component of S must be confined to one such hyperplane. Since S is the zero set of the exponential polynomials H and the subdeterminants of J, the number of connected components is finite by Khovanski's theorem (Marker, 1996, p. 757), so that L^η must be well-defined for all but a finite number of values of η. This completes proof of Lemma 8.3.

Lemma 8.4. *The curves L^η are uniformly bounded for $\eta \in (0, \bar{\eta})$. Furthermore, a uniform bound $\bar{\ell}$ exists for the arc lengths ℓ^η of these curves.*

The curves consist of equilibria of the respective auxiliary games. At the end of Section 3.3.2, we showed that values are bounded in equilibrium for given η. Continuity allows to obtain a uniform bound by simply taking the minimum of the lower and the maximum of the upper bound over $[0, \bar{\eta}]$.

We will denote by ℓ^η the arc length of each curve L^η. Proposition 5 showed that ℓ^η is always finite; a similar argument establishes a uniform bound $\bar{\ell}$ for $\eta \in (0, \bar{\eta})$. Recall that turning points of L^η are characterized by $H = \mathbf{0}$ and one sub-determinant of J crossing 0; both conditions can be expressed as exponential polynomials (see proof of Proposition 5 for details). By Khovanski's theorem, the number of connected components of these zero sets is bounded from above by a function of the complexity of the system (Hovanskii, 1980, Theorem 4; Marker, 1996, p. 757). Because the complexity does not vary with η, the number of turning points is uniformly bounded from above. Together with all L^η being uniformly bounded, this implies that total arc length of the curves has an upper bound $\bar{\ell}$.

Bounding total arc length allows to parametrize the curves L^η, $\eta \in (0, \bar{\eta})$, as a family of functions on the real interval $[0, \bar{\ell}]$ in the following way. For $s \in [0, \ell^\eta]$, let $L^\eta(s)$ be the point with distance s from the starting point, measured along the curve, so that $L^\eta(0)$ is its starting point and $L^\eta(\ell^\eta)$ its distinguished equilibrium (see Online Appendix 3.F for a more explicit definition). For $s \in [\ell^\eta, \bar{\ell}]$, simply set $L^\eta(s) := L^\eta(\ell^\eta)$. Convergence can now be shown as follows.

Lemma 8.5. *Let $(\eta_1, \eta_2, ...)$ be a decreasing sequence in $(0, \bar{\eta})$ with limit 0. Then the sequence $(L^{\eta_1}, L^{\eta_2}, ...)$ converges uniformly to some continuous function L^0.*

By Lemma 7.4, the family of functions $L^\eta(s)$ is uniformly bounded. Being parametrized in arc length, each L^η is Lipschitz continuous with constant 1; thus, the the family is uniformly equicontinuous. By Arzelà–Ascoli (Rudin, 1976, Theorem 7.25), each sequence from L^η must have a subsequence converging uniformly to some continuous function.

It remains to be shown that the subsequential limit is in fact unique as $\eta \to 0$. Consider a sequence $(L^{\eta_1}, L^{\eta_2}, ...)$ with $\eta_1 > \eta_2 > ...$, and suppose without loss of generality that the odd terms converge to some path L^0_{odd}, while the even terms converge to L^0_{even}. The following shows that these paths must have identical images, i.e. parametrize the same curve. Assume to the contrary that there exists a point $x = (\boldsymbol{\sigma}, \boldsymbol{V}, t) \in \operatorname{Im} L^0_{\text{even}}$ which is not in $\operatorname{Im} L^0_{\text{odd}}$. Then there exists an open neighborhood of x which does not intersect $\operatorname{Im} L^0_{\text{odd}}$ (the latter is closed due to continuity). Any such neighborhood intersects only finitely many odd members of the sequence, but an infinite number of even members; this allows to find a hypersphere S_x centered at x for which the same holds. We now consider $(\boldsymbol{\sigma}, \boldsymbol{V}, t, \eta)$-space, where the curves $L^\eta \times \{\eta\}$ are contained in $\tilde{Z}$. The set $\tilde{Z} \cap (S_x \times (0, \bar{\eta}))$ must have an infinite number of connected components, because for i sufficiently large, the intersection is empty at all odd η_i, and nonempty at all even η_i. Since both $\tilde{Z}$ and S_x are the zero set of exponential polynomials, this is a contradiction by Khovanski's theorem (Marker, 1996, p. 757). This completes the proof of Proposition 8. ■

Now that existence of the limiting curve is established, we turn to its properties. By the following proposition, its endpoint is an equilibrium of the original game.

Proposition 9. *$L^0(\bar{\ell})$ is a stationary equilibrium of $\mathcal{G}$.*

Proof. $L^0(\bar{\ell})$ is the limit of a sequence $\left(L^\eta(\bar{\ell})\right)$. Since each $L^\eta(\bar{\ell})$ is an equilibrium of $\mathcal{G}$ and the set of equilibria is closed, $L^0(\bar{\ell})$ must also be an equilibrium of $\mathcal{G}$. ■

Finally, we show that the curve L^0 is contained in the solution set of the linear procedure. This set is simply the set of stationary equilibria for the auxiliary games $\mathcal{G}^t$ for $t \in [0,1]$ as defined in Section 3.3.1, but with $\eta = 0$. Call these games $\mathcal{G}_0^t$.

Proposition 10. *For any t, let $(\boldsymbol{\sigma}^n, \boldsymbol{V}^n, t, \eta^n) \to (\boldsymbol{\sigma}^N, \boldsymbol{V}^N, t, 0)$ be a sequence with $(\boldsymbol{\sigma}^n, \boldsymbol{V}^n, t) \in L^{\eta^n}$. Then $(\boldsymbol{\sigma}^N, \boldsymbol{V}^N)$ is an equilibrium of $\mathcal{G}_0^t$.*

Proof. Similar to Proposition 7. The pair $(\boldsymbol{\sigma}^N, \boldsymbol{V}^N)$ is an equilibrium of $\mathcal{G}_0^t$ if and only if for all $(s,i) \in S \times I$:

$$V_{si}^N = \bar{u}_{si}^t(\boldsymbol{\sigma}_s^N) \;+\; \delta_i \sum_{s' \in S} \bar{\phi}_{s \to s'}^t(\boldsymbol{\sigma}_s^N)\; V_{s'i}^N = \bar{U}_{si}^t(\boldsymbol{\sigma}_s^N, \boldsymbol{V}_i^N) \tag{8a}$$

$$\sum_{a \in A_{si}} \sigma_{sia}^N = 1 \tag{8b}$$

$$\sigma_{sia}^N \geq 0 \qquad \forall a \in A_{si} \tag{8c}$$

$$\bar{U}_{si}^t(a, \boldsymbol{\sigma}_{s,\text{-}i}^N, \boldsymbol{V}_i^N) \geq \bar{U}_{si}^t(a', \boldsymbol{\sigma}_{s,\text{-}i}^N, \boldsymbol{V}_i^N) \quad \vee \quad \sigma_{sia}^N = 0 \qquad \forall a, a' \in A_{si} \tag{8d}$$

Since $(\boldsymbol{\sigma}^n, \boldsymbol{V}^n, t) \in L^{\eta^n}$, each such point satisfies the constraints (2b–d), which correspond to (8a–c) as $\eta \to 0$. Finally, (8d) can be established similarly as in Proposition 7. For any pair of actions $a, a' \in A_{si}$, we must have

$$\sigma_{sia'}^n H_{sia}^V(\boldsymbol{\sigma}^n, \boldsymbol{V}^n, t^n) - \sigma_{sia}^n H_{sia'}^V(\boldsymbol{\sigma}^n, \boldsymbol{V}^n, t^n) = 0$$

for all n and therefore

$$\sigma_{sia}^n \sigma_{sia'}^n \left(\bar{U}_{si}^t(a, \boldsymbol{\sigma}_{s,\text{-}i}^n, \boldsymbol{V}_i^n) - \bar{U}_{si}^t(a', \boldsymbol{\sigma}_{s,\text{-}i}^n, \boldsymbol{V}_i^n)\right) + (1-t)\eta^n(\nu_{sia}\sigma_{sia'}^n - \nu_{sia'}\sigma_{sia}^n) = 0 \tag{9}$$

$\eta^n \to 0$ then implies

$$\lim_{n \to \infty} \sigma_{sia}^n \sigma_{sia'}^n \left(\bar{U}_{si}^t(a, \boldsymbol{\sigma}_{s,\text{-}i}^n, \boldsymbol{V}_i^n) - \bar{U}_{si}^t(a', \boldsymbol{\sigma}_{s,\text{-}i}^n, \boldsymbol{V}_i^n)\right) = 0$$

and at least one factor must go to zero. If

$$\lim_{n \to \infty} \left(\bar{U}_{si}^t(a, \boldsymbol{\sigma}_{s,\text{-}i}^n, \boldsymbol{V}_i^n) - \bar{U}_{si}^t(a', \boldsymbol{\sigma}_{s,\text{-}i}^n, \boldsymbol{V}_i^n)\right) = 0,$$

then (8d) holds immediately. If conversely and without loss of generality

$$\lim_{n\to\infty}\left(\bar{U}^t_{si}(a,\boldsymbol{\sigma}^n_{s,\text{-}i},\boldsymbol{V}^n_i)-\bar{U}^t_{si}(a',\boldsymbol{\sigma}^n_{s,\text{-}i},\boldsymbol{V}^n_i)\right)>0,$$

(8d) is still satisfied unless

$$\lim_{n\to\infty}\sigma^n_{sia'}>0=\lim_{n\to\infty}\sigma^n_{sia}$$

However, if both limits are positive, both summands in equation (9) must eventually be positive. This is a contradiction. ■

Corollary 10.1. The limit curve L^0 is contained in the solution set of the linear stochastic tracing procedure. In particular, whenever the latter defines a unique, isolated path connecting $t=0$ and $t=1$, then L^0 must be identical to that path. Otherwise, it will be a selection from among the multitude of paths connecting $t=0$ and $t=1$ in the linear solution set. Note that this selection may depend on $\boldsymbol{\nu}$.

Summary of the second main result. As this section has shown, the solution curve of the stochastic logarithmic tracing procedure smoothly approaches a limiting curve L^0 as $\eta\searrow 0$. This curve exists for any game and any prior, and its endpoint is a stationary equilibrium of $\mathcal{G}$. In this manner, the procedure always selects a specific equilibrium for any given $\mathcal{G}$ and $\boldsymbol{\rho}$. In case of non-generic games, selection may also depend on $\boldsymbol{\nu}$.

L^0 is always contained in the solution set used by the linear tracing procedure for stochastic games. In consequence, both methods agree in their selection whenever the linear variant is well-defined. On the other hand, for games and priors where the linear procedure does not induce a unique, isolated solution curve and thus also fails to provide a unique selection, the logarithmic procedure circumvents these issues. Still, it stays as close in spirit as possible: Its limiting path is then one of the many paths suggested by the linear variant.

3.6 Timings

This section demonstrates the performance of the logarithmic tracing procedure in the computation of stationary equilibria. A numerical implementation of the procedure is publicly available as part of the python package `sgamesolver` (see Chapter 4, code at `github.com/davidpoensgen/sgamesolver`), alongside other homotopy methods for finite discounted stochastic games. Required inputs are simply $\boldsymbol{u}$, $\boldsymbol{\phi}$, and $\boldsymbol{\rho}$, which can be passed as arrays or as a table. (If desired also a vector $\boldsymbol{\nu}$, although in our experience, simply setting $\boldsymbol{\nu} = \mathbf{1}$ works well.) The software includes routines that evaluate H and J, as well as a path tracking algorithm.

For the timings reported here, games were drawn randomly according to the following specifications. Discount factors are fixed to $\delta = 0.95$ for all players. Payoffs are independently and identically drawn from a uniform distribution over $[0, 1]$. In order sample transition probabilities uniformly from the unit simplex, probabilities are drawn independently and identically from an exponential distribution and then normalized such that transition probabilities sum up to one for each state and action profile (Rubinstein and Melamed, 1998, algorithm 2.7.1). As priors $\boldsymbol{\rho}$, we take the centroid strategy profile, i.e. uniform mixing over available actions. The remaining parameters are $\eta = 0.1$ and $\boldsymbol{\nu} = \mathbf{1}$. Game sizes and timings are listed in Table 3.1.

The way games are randomized here is comparable to e.g. Dang et al. (2022); note that the resulting games are generic. We repeated the timings with a slightly different specification yielding similar, but non-generic games. Timings for those are largely comparable, but about 11% slower on average. Details and a table are given in Online Appendix 3.G, alongside instructions on how to repeat both sets of computations.

All computations were done on a typical desktop computer with an Intel i5 3.0 GHz processor and 16 GB working memory. The runs reported here were unattended, using the same default parameters for all game sizes. Less than 1% of all games were not solved successfully with these defaults. However, in these cases, the same game could be solved by re-running the solver with just slightly changed path tracking settings. We expect that the algorithm can tackle still larger games, but this should ideally be done under attendance and while tuning the parameters to the specific game – which was not economical for this timing exercise.

Regarding the dependency of computation time on game size, three factors are at play. The first is the linear algebra involved in the algorithm: Each step requires computing a QR-decomposition and a pseudo-inverse of J. Both operations scale at $O(n^3)$ in the size of the system $H = 0$, which generally is $\sum_{(s,i)\in S\times I}(|A_{si}| + 1)$, as each agent contributes an equation for each action and a sum-to-one condition. In the games reported here, it equals $|S|\,|I|\,(|A_{si}| + 1)$, since all agents have the

$\lvert S\rvert$	$\lvert A_{si}\rvert$	$\lvert I\rvert$			
		2	3	4	5
1	2	0:00 0:00	0:00 0:00	0:00 0:00	0:00 0:00
	4	0:00 0:00	0:00 0:00	0:00 0:00	0:00 0:00
	8	0:00 0:00	0:00 0:00	0:00 0:00	0:02 0:01
2	2	0:00 0:00	0:00 0:00	0:00 0:00	0:00 0:00
	4	0:00 0:00	0:00 0:00	0:00 0:00	0:00 0:00
	8	0:00 0:00	0:00 0:00	0:01 0:00	0:05 0:02
5	2	0:00 0:00	0:00 0:00	0:00 0:00	0:00 0:00
	4	0:00 0:00	0:00 0:00	0:00 0:00	0:01 0:01
	8	0:00 0:00	0:01 0:00	0:04 0:02	0:49 0:26
10	2	0:00 0:00	0:00 0:00	0:00 0:00	0:00 0:00
	4	0:00 0:00	0:01 0:00	0:02 0:01	0:05 0:03
	8	0:01 0:00	0:04 0:04	0:23 0:20	4:08 3:00
20	2	0:00 0:00	0:00 0:00	0:01 0:00	0:01 0:01
	4	0:01 0:00	0:03 0:02	0:13 0:08	0:56 0:45
	8	0:03 0:02	0:42 0:32	5:03 2:46	1:08:10 46:59
50	2	0:01 0:00	0:03 0:01	0:09 0:05	0:37 0:28
	4	0:07 0:05	1:16 1:16	11:19 9:24	48:45 26:57
	8	1:58 1:54	37:17 17:56		
100	2	0:04 0:03	0:23 0:14	2:03 2:05	9:42 9:19
	4	1:09 0:50	24:16 27:03		
	8	1:00:40 1:07:46			
200	2	0:25 0:07	3:55 2:28	41:31 33:19	4:17:00 1:46:05
	4	24:20 23:02			
400	2	3:20 1:07			
800	2	30:12 19:12			

Table 3.1: Computation times to solve random (generic) games with $|S|$ states, $|I|$ players and $|A_{si}|$ actions for each player in each state. Listed are average times as well as standard deviations (in small print) in *m*:*ss* or *h*:*mm*:*ss*. All timings under 15:00 are based on 100 independently drawn games of the respective size; all others on 10 games per size.

same number of actions $|A_{si}|$. The costs related to linear algebra dominate in games with large state space, but few players and actions per player, i.e. towards the lower left of the table.

The second factor is the evaluation of u and ϕ and their derivatives with respect to strategies σ_{sia}, which appear in H and J (compare Online Appendix 3.D). For example, to compute $u_{si}(\boldsymbol{\sigma}_s)$ for each player and each state, one has to evaluate a product sum over the outer product of $\boldsymbol{\sigma}_s$ and all entries of u_{si} (i.e. over all action profiles). This scales only linearly in the number of states (or rather quadratically, if one considers the need to multiply in continuation values). But it scales quite unfavorably if the game contains a high number of action profiles per state. In the evenly shaped games reported here, there are $|A_{si}|^{|I|}$ action profiles per state, which explains why computation times go up dramatically if both $|A_{si}|$ and $|I|$ are high. On the other hand, this cost is rather negligible if either number is low; and it is essentially absent in games with a sequential structure.

Finally, increasing the size of the game generally leads to a longer and potentially more winded path, so that more predictor-corrector steps are necessary to trace it.

It is important to note that the randomized games used for a benchmark here do not involve any symmetries between states or players. However, whenever a game involves symmetrical agents, the rows of all but one of them can be removed from H, resulting in a far smaller system and significant decreases in computation time.

3.7 Conclusion

This paper introduces the logarithmic tracing procedure for stochastic games, a homotopy method for the computation of at least one stationary equilibrium of any finite discounted stochastic game. Because the solution path is guaranteed to be isolated and smooth, the method is well-suited for numerical application.

The homotopy function is constructed from the equilibrium conditions of auxiliary games, in which players maximize against a convex combination of a prior $\boldsymbol{\rho}$ and other players' best responses. Payoffs in the auxiliary games additionally include logarithmic regularization penalties, which ensure that the path is isolated, interior, and smooth. By varying the homotopy parameter t from 0 to 1, the methods proceeds by tracing a path of auxiliary equilibria, until finally an equilibrium of the original game is reached. Harsanyi and Selten (1988), who proposed a similar procedure for normal form games, interpret the traversal of this path as a form of Bayesian strategic reasoning in which priors are gradually transformed into equilibrium beliefs. Consequently, they suggest to use their tracing procedure as a tool for equilibrium selection. This interpretation of course carries over to our procedure for stochastic games, as does the possibility to use it as part of a selection criterion. The prior $\boldsymbol{\rho}$ used in the construction of auxiliary games can be chosen freely. The method can therefore be used to find the equilibrium that results from a particularly focal prior. To obtain an actual selection criterion, one of course needs in addition a rule on which prior to use, as suggested by Harsanyi and Selten (1988). By performing a grid search on the prior space, the method alternatively allows to approximate the basins of attraction of different equilibria in size and shape, which could be used as another potential basis for a selection criterion, if desired.

The present method is a generalization of the linear procedure for stochastic games by Herings and Peeters (2004), which is guaranteed to be well-defined only for generic games and priors. Our method makes the same use of priors, but the addition of logarithmic penalties ensures regularity, so that it is applicable to any finite discounted stochastic game.

The close relation between these methods can be shown formally when considering the limiting curve L^0, which is obtained by letting the weight on the penalty terms go to zero. As we have shown, this curve is identical to the curve of the linear method whenever the latter is well-defined. Consequently, the selected equilibria of both methods then also coincide. In cases where the linear variant is not well-defined, L^0 is always contained in its solution graph, so that it can be understood as a unique selection from all paths and equilibria consistent with the linear method.

Beyond these theoretical considerations, the present method allows the efficient numerical computation of stationary equilibria for stochastic games even of considerable size. A ready-to-use implementation of the algorithm is publicly available as part of the python package `sgamesolver` (Eibelshäuser and Poensgen, 2019). Section 3.6 reports timings, which compare quite favorably to available alternatives.

Appendix

3.A Continuity of $\hat{u}$ and H

This section expands on the discussion of footnote 15 regarding the continuity of $\hat{u}$ (Section 3.3.1) and H (equation 3). Generally, these functions are not jointly continuous at points where one or more $\sigma_{sia} = 0$ and $(1-t)\eta = 0$, because it is not ensured that

$$\lim_{(1-t)\eta \searrow 0} (1-t)\eta \log(\sigma_{sia}) = 0$$

As an example, consider the sequence $\Big((1-t)\eta, \sigma_{sia}\Big)_n = \left(\frac{1}{n}, \frac{1}{e^n}\right)$, for which the above expression is constant and does not approach 0 as would be required for continuity. Intuitively, the example requires σ to decrease much faster than $(1-t)\eta$. However, the following will show this can not occur in the equilibrium sets (i.e. the zero set of H) used by the algorithm. The reason is that in equilibrium, σ_{sia} is bounded from below by a function of $(1-t)\eta$ which ensures convergence to 0. Thus, one could construct a domain that includes all equilibria while also ensuring joint continuity of $\hat{u}$ and H .

The following argument establishes existence of said bound. Clearly, only sequences where $\sigma_{sia} \to 0$ are potentially problematic. For each such action a, there must be another action $a' \in A_{si}$ of the same agent (s, i) that converges to a number strictly greater than 0. Consider then the equation

$$\frac{H^V_{sia'}(\boldsymbol{\sigma}, \boldsymbol{V}, t)}{\sigma_{sia'}} - \frac{H^V_{sia}(\boldsymbol{\sigma}, \boldsymbol{V}, t)}{\sigma_{sia}} = \\ \bar{U}^t_{si}(a', \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) - \bar{U}^t_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) + (1-t)\eta \left(\frac{\nu_{sia'}}{\sigma_{sia'}} - \frac{\nu_{sia}}{\sigma_{sia}} \right) = 0$$

derived from equation (3b) which must hold for all points in $H^{-1}(0)$. Rearranging terms yields

$$\sigma_{sia} = \frac{\nu_{sia}}{\frac{\bar{U}^t_{si}(a', \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) - \bar{U}^t_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i)}{(1-t)\eta} + \frac{\nu_{sia'}}{\sigma_{sia'}}}$$

Here, ν_{sia} and $\nu_{sia'}$ are positive constants. By assumption, $\sigma_{sia'} \not\to 0$ so that $\frac{\nu_{sia'}}{\sigma_{sia'}}$ is bounded. Because V_{si} are bounded in equilibrium (see Proposition 2), $\bar{U}^t_{si}$ are also bounded for all $\boldsymbol{\sigma}, \boldsymbol{V}, t$ in equilibrium. In addition, the fact that σ_{sia} goes to 0, while $\sigma_{sia'}$ does not, implies $\bar{U}^t_{si}(a', \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) - \bar{U}^t_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) \geq 0$ for $(1-t)\eta$ sufficiently small (action a' must be at least as good as action a in equilibrium; compare proofs of Propositions 7 and 10). Therefore, for small $(1-t)\eta$ it holds that $M \geq \bar{U}^t_{si}(a', \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) - \bar{U}^t_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) \geq 0$ for some $M > 0$. Using all of the preceding, one can bound σ_{sia} by two expressions of the form $\frac{1}{\frac{q}{(1-t)\eta}+r}$ with $q \geq 0, r > 0$. Because

$$\lim_{(1-t)\eta\searrow 0} (1-t)\eta \log\left(\frac{1}{\frac{q}{(1-t)\eta}+r}\right) = 0$$

for all $q \geq 0, r > 0$, the sandwich theorem then gives

$$\lim_{(1-t)\eta\searrow 0} (1-t)\eta \log(\sigma_{sia}) = 0$$

as claimed.

3.B Derivation of the Homotopy Function H

This section provides the proof of Proposition 1, which states that the zero set of $H(\boldsymbol{\sigma}, \boldsymbol{V}, t)$ coincides with the set of equilibria of the auxiliary stochastic games $\mathcal{G}^t$. To this end, we derive the homotopy function H from the maximization problems stated in equation (2). $H = 0$ corresponds to the problems' first order conditions, which are not only necessary, but due to concavity also sufficient for an equilibrium.

The Lagrangeans corresponding to the maximization problems stated in equation (2) are given by

$$\begin{aligned}\mathcal{L}^t_{si} \;=\; V^t_{si} \;+\; \alpha_{si}\Bigg[&- V^t_{si} \;+\; \bar{u}^t_{si}(\boldsymbol{\sigma}_s) \;+\; \delta_i \sum_{s'\in S} \bar{\phi}^t_{s\to s'}(\boldsymbol{\sigma}_s)\; V^t_{s'i} \\ &+ (1-t)\eta \sum_{a\in A_{si}} \nu_{sia} \log(\sigma_{sia}) \Bigg] \;+\; \beta_{si}\Bigg[\sum_{a\in A_{si}} \sigma_{sia} - 1 \Bigg]\end{aligned}$$

where $\alpha_{si}, \beta_{si} \neq 0$ are the Lagrange multipliers of the two constraints (2b) and (2c). Since the logarithmic penalty terms are strictly concave in $\boldsymbol{\sigma}_{si}$ and all other terms are linear in $\boldsymbol{\sigma}_{si}$, the Karush-Kuhn-Tucker conditions are both necessary and sufficient. For each agent $(s,i) \in S\times I$, they consist of the two constraints, as well as one equation for each action $a \in A_{si}$:

$$\frac{\partial \mathcal{L}^t_{si}}{\partial \sigma_{sia}} = \alpha_{si}\Bigg[\; \bar{u}^t_{si}(a, \boldsymbol{\sigma}_{si}) + \delta_i \sum_{s'\in S} \bar{\phi}^t_{s\to s'}(a, \boldsymbol{\sigma}_{s,\text{-}i})\; V^t_{s'i} + (1-t)\frac{\eta\nu_{sia}}{\sigma_{sia}} \;\Bigg] \;+\; \beta_{si} = 0$$

Multiplying each of these equations by the corresponding σ_{sia} and summing up over $a \in A_{si}$ yields

$$\alpha_{si}\underbrace{\left[\,\bar{u}^t_{si}(\boldsymbol{\sigma}_{si},\boldsymbol{\sigma}_{s,\text{-}i}) + \delta_i \sum_{s'\in S} \bar{\phi}^t_{s\to s'}(\boldsymbol{\sigma}_{si},\boldsymbol{\sigma}_{s,\text{-}i})\; V^t_{s'i} + (1-t)\eta \sum_{a\in A_{si}} \nu_{sia}\right]}_{=V^t_{si}-(1-t)\eta\sum_{a\in A_{si}}\nu_{sia}\left[\log(\sigma_{sia})-1\right]} + \beta_{si}\underbrace{\sum_{a\in A_{si}}\sigma_{sia}}_{=1} = 0$$

and thus

$$\frac{\beta_{si}}{\alpha_{si}} \;=\; -V^t_{si} \;+\; (1-t)\eta \sum_{a\in A_{si}} \nu_{sia}\Big[\log(\sigma_{sia})-1\Big]$$

Replacing $\frac{\beta_{si}}{\alpha_{si}}$ in the first order conditions $\frac{\partial \mathcal{L}^t_{si}}{\partial \sigma_{sia}} = 0$ gives the following necessary and sufficient conditions for all agents $(s,i) \in S \times I$:

$$\begin{aligned} 0 \;&=\; -V^t_{si} + \bar{u}^t_{si}(a,\boldsymbol{\sigma}_{s,\text{-}i}) \;+\; \delta_i \sum_{s'\in S} \bar{\phi}^t_{s\to s'}(a,\boldsymbol{\sigma}_{s,\text{-}i})\; V^t_{s'i} \\ &\qquad\qquad +\; (1-t)\eta\left(\frac{\nu_{sia}}{\sigma_{sia}} + \sum_{a'\in A_{si}} \nu_{sia'}\Big[\log(\sigma_{sia'})-1\Big]\right) \qquad \forall\; a \in A_{si} \\ 0 \;&=\; \sum_{a\in A_{si}} \sigma_{sia} - 1 \end{aligned}$$

which characterize the set of stationary equilibria of the game $\mathcal{G}^t$.

The homotopy function H is obtained by multiplying the right hand sides of the former set of equations by the corresponding σ_{sia}, and then collecting all conditions for all agents.

3.C The Starting Point (σ^0, V^0)

3.C.1 Existence and Uniqueness

This section provides the proof of Proposition 3, which states that the auxiliary stochastic game $\mathcal{G}^0$ always has a unique equilibrium $(\boldsymbol{\sigma}^0, \boldsymbol{V}^0)$. At $t = 0$, players maximize solely against their prior $\boldsymbol{\rho}_{\text{-}i}$, so that strategic interaction is completely absent. Formally, each player faces a discounted Markov decision problem with finite state space S, action spaces $\Delta(A_{si})$, and state transition functions $\bar{\phi}^0_{s\to s'}(\boldsymbol{\sigma}_{si}) = \phi_{s\to s'}(\boldsymbol{\sigma}_{si}, \boldsymbol{\rho}_{s,\text{-}i})$. Instantaneous utilities are given by:

$$\begin{aligned} \hat{u}^0_{si} : \Delta(A_{si}) &\quad\rightarrow\quad \{-\infty\} \cup \mathbb{R} \\ \boldsymbol{\sigma}_{si} &\quad\mapsto\quad u_{si}(\boldsymbol{\sigma}_{si}, \boldsymbol{\rho}_{s,\text{-}i}) + \eta \sum_{a \in A_{si}} \nu_{sia} \log(\sigma_{sia}) \end{aligned}$$

For the purpose of proving existence, it is helpful to set $\log(0) := -\infty$ and take the extended real line $\{-\infty\} \cup \mathbb{R}$ as range for $\hat{u}$, as indicated above. Note that still $\hat{u} < \infty$, so that total expected discounted utilities are always well-defined. Denote as $\hat{U}^0_{si}(\boldsymbol{\sigma}_i)$ total utility under $\boldsymbol{\sigma}_i$ and beginning in state s, so that one obtains in vector notation:

$$\hat{\boldsymbol{U}}^0_i(\boldsymbol{\sigma}_i) = \Big(\hat{U}^0_{1i}(\boldsymbol{\sigma}_i), \hat{U}^0_{2i}(\boldsymbol{\sigma}_i), \dots\Big)^\top = \Big(I - \delta_i \Phi(\boldsymbol{\sigma}_i)\Big)^{-1} \hat{\boldsymbol{u}}^0_i(\boldsymbol{\sigma}_i)$$

Any solution to the Markov decision problem of player i must then satisfy

$$V_{si} = \max_{\boldsymbol{\sigma}_i \in \times_{s \in S} \Delta(A_{si})} \hat{U}^0_{si}(\boldsymbol{\sigma}_i) \qquad \forall s \in S$$

Because $\hat{U}^0_{si} : \bigtimes_{s \in S} \Delta(A_{si}) \rightarrow \{-\infty\} \cup \mathbb{R}$ is upper semi-continuous over a compact domain, the respective version of the extreme value theorem guarantees that this maximum exists (Bourbaki, 1966, Ch. IV, § 6, Theorem 3, p. 361). Moreover it must be that $V_{si} > -\infty$, because an arbitrary interior strategy guarantees some finite total discounted utility in every state, giving a lower bound for the maximized values. Thus there exists a unique vector $\boldsymbol{V}^0_i \in \mathbb{R}^{|S|}$ of state values for each player. Given this vector, optimal strategies $\boldsymbol{\sigma}^0$ are the solutions to

$$\boldsymbol{\sigma}^0_{si} = \underset{\boldsymbol{\sigma}_{si} \in \Delta(A_{si})}{\arg\max} \left\{ u_{si}(\boldsymbol{\sigma}_{si}, \boldsymbol{\rho}_{s,\text{-}i}) + \delta_i \sum_{s' \in S} \phi_{s \to s'}(\boldsymbol{\sigma}_{si}, \boldsymbol{\rho}_{s,\text{-}i})\; V^0_{s'i} + \eta \sum_{a \in A_{si}} \nu_{sia} \log(\sigma_{sia}) \right\}$$

Because the first two terms are linear in $\boldsymbol{\sigma}_{si}$, while the logarithmic term is strictly concave, optimal strategies are also unique.

3.C.2 Computation

To compute initial values $\boldsymbol{V}^0$ and strategies $\boldsymbol{\sigma}^0$, one can use standard value function iteration on the Bellman equation

$$\begin{aligned} V^{(k+1)}_{si} &= \max_{\boldsymbol{\sigma}_{si}} \left[u_{si}(\boldsymbol{\sigma}_{si}, \boldsymbol{\rho}_{s,\text{-}i}) + \delta_i \sum_{s' \in S} \phi_{s \to s'}(\boldsymbol{\sigma}_{si}, \boldsymbol{\rho}_{s,\text{-}i})\; V^{(k)}_{s'i} + \eta \sum_{a \in A_{si}} \nu_{sia} \log(\sigma_{sia}) \right] \\ &=: \max_{\boldsymbol{\sigma}_{si}} \left[U^k_{si}(\boldsymbol{\sigma}_{si}) + \eta \sum_{a \in A_{si}} \nu_{sia} \log(\sigma_{sia}) \right] \end{aligned}$$

where k counts iterations and U^k is simply a shorthand for the first two terms: The linear part of instantaneous utility plus expected discounted future value under the current estimate $V^{(k)}$. Introducing a multiplier γ_{si} for the constraint $\sum_a \sigma_{sia} = 1$, necessary and sufficient first order conditions for each (s,i) are then

$$\frac{\partial U^k_{si}\left(\boldsymbol{\sigma}_{si}\right)}{\partial \sigma_{sia}} + \frac{\eta \nu_{sia}}{\sigma_{sia}} = U^k_{sia} + \frac{\eta \nu_{sia}}{\sigma_{sia}} = \gamma_{si} \qquad \forall a \in A_{si}$$

$$\sum_{a \in A_{si}} \sigma_{sia} = 1$$

Dropping indices s, i, and k, and labeling the strategies $\sigma_1, ..., \sigma_N$ with $N = |A_{si}|$, the above implies

$$U_1 + \frac{\eta \nu_1}{\sigma_1} \;=\; U_n + \frac{\eta \nu_n}{\sigma_n} \qquad\qquad \text{for } n = 1, \ldots, N$$

and thus

$$\sigma_n = \frac{\nu_n}{\frac{U_1 - U_n}{\eta} + \frac{\nu_1}{\sigma_1}}$$

Plugging into the constraint gives an equation only in σ_1:

$$f(\sigma_1) \;:=\; \sum_{n=1}^{N} \frac{\nu_n}{\frac{U_1 - U_n}{\eta} + \frac{\nu_1}{\sigma_1}} - 1 \;=\; 0$$

If $N = 1$, then $\sigma_1 = 1$. If $N > 1$, order strategies such that $U_1 - U_n \geq 0$ for all n, without loss of generality. The following shows that the equation then always has a unique solution $\sigma_1 \in (0,1)$. First, f is continuous and monotonously increasing on the open unit interval:

$$f'(\sigma_1) \;=\; \sum_{n=1}^{N} \frac{\eta^2 \nu_1 \nu_n}{\Big[(U_1 - U_n)\sigma_1 + \eta \nu_1\Big]^2} \;>\; 0$$

Secondly, behavior at the boundaries of $(0,1)$ is given by

$$\begin{aligned} \lim_{\sigma_1 \to 0} f(\sigma_1) \;&=\; -1 \\ \lim_{\sigma_1 \to 1} f(\sigma_1) \;&=\; \sum_{n=1}^{N} \frac{\nu_n}{\frac{U_1 - U_n}{\eta} + \nu_1} - 1 \;=\; \sum_{n=2}^{N} \frac{\nu_n}{\frac{U_1 - U_n}{\eta} + \nu_1} \;>\; 0 \end{aligned}$$

By application of the intermediate value theorem, there exists a unique solution for σ_1 in the unit interval $(0,1)$, which can be found by standard root-finding algorithms. All other σ_n are then also uniquely determined by the first order conditions. Plugging the optimal policy into the Bellman equation yields the next value iterate $\boldsymbol{V}^{(k+1)}$. Starting from an arbitrary $\boldsymbol{V}^{(0)}$, e.g. the zero vector, this process can be repeated until $\boldsymbol{V}$ has converged.

3.D Jacobian of H

The Jacobian matrix of $H(\boldsymbol{\sigma}, \boldsymbol{V}, \boldsymbol{\nu}, \eta, t)$ is

$J : [0,1]^{|A|} \times \mathbb{R}^{|S\times I|} \times \mathbb{R}_+^{|A|} \times [0,\infty) \times [0,1] \to \mathbb{R}^{|A|+|S\times I|} \times \mathbb{R}^{|A|+|S\times I|+|A|+2}$,

$$J(\boldsymbol{\sigma}, \boldsymbol{V}, \boldsymbol{\nu}, \eta, t) \quad = \quad \begin{pmatrix} \frac{\partial H^V_{sia}}{\partial \sigma_{s'i'a'}} & \frac{\partial H^V_{sia}}{\partial V_{s'i'}} & \frac{\partial H^V_{sia}}{\partial \nu_{s'i'a'}} & \frac{\partial H^V_{sia}}{\partial \eta} & \frac{\partial H^V_{sia}}{\partial t} \\ \frac{\partial H^\sigma_{si}}{\partial \sigma_{s'i'a'}} & \frac{\partial H^\sigma_{si}}{\partial V_{s'i'}} & \frac{\partial H^\sigma_{si}}{\partial \nu_{s'i'a'}} & \frac{\partial H^\sigma_{si}}{\partial \eta} & \frac{\partial H^\sigma_{si}}{\partial t} \end{pmatrix}$$

with

$$\frac{\partial H^V_{sia}(\boldsymbol{\sigma}, \boldsymbol{V}, \boldsymbol{\nu}, \eta, t)}{\partial \sigma_{s'i'a'}} = \begin{cases} -V_{si} + \bar{u}^t_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}) + \delta_i \sum\limits_{s''\in S} \bar{\phi}^t_{s\to s''}(a, \boldsymbol{\sigma}_{s,\text{-}i})\; V_{s''i} & \text{if } s'=s, i'=i \\ \qquad + (1-t)\eta\left(\nu_{sia} + \sum\limits_{a''\in A_{si}} \nu_{sia''}\Big[\log(\sigma_{sia''}) - 1\Big]\right) & \text{and } a'=a \\ (1-t)\eta\nu_{sia'}\dfrac{\sigma_{sia}}{\sigma_{sia'}} & \text{if } s'=s, i'=i \\ & \text{and } a'\neq a \\ t\sigma_{sia}\Bigg[u_{si}(a_{si}, a_{s,i'}, \boldsymbol{\sigma}_{s,\text{-}\{i,i'\}}) & \text{if } s'=s \\ \qquad + \delta_i \sum\limits_{s''\in S} \bar{\phi}^t_{s\to s''}(a_{si}, a_{s,i'}, \boldsymbol{\sigma}_{s,\text{-}\{i,i'\}})\; V_{s'',i}\Bigg] & \text{and } i'\neq i \\ 0 & \text{else} \end{cases}$$

$$\frac{\partial H^V_{sia}(\boldsymbol{\sigma}, \boldsymbol{V}, \boldsymbol{\nu}, \eta, t)}{\partial V_{s'i'}} = \begin{cases} \sigma_{sia}\left(\delta_i \bar{\phi}^t_{s\to s'}(a, \boldsymbol{\sigma}_{s,\text{-}i}) - 1\right) & \text{if } i'=i \text{ and } s'=s \\ \sigma_{sia}\; \delta_i \bar{\phi}^t_{s\to s'}(a, \boldsymbol{\sigma}_{s,\text{-}i}) & \text{if } i'=i \text{ and } s'\neq s \\ 0 & \text{if } i'\neq i \end{cases}$$

$$\frac{\partial H^V_{sia}(\boldsymbol{\sigma},\boldsymbol{V},\boldsymbol{\nu},\eta,t)}{\partial \nu_{s'i'a'}} = \begin{cases} (1-t)\eta\Big(1+\sigma_{sia}\Big[\log(\sigma_{sia})-1\Big]\Big) & \text{if } s'=s, i'=i \text{ and } a'=a \\ (1-t)\eta\sigma_{sia}\Big[\log(\sigma_{sia})-1\Big] & \text{if } s'=s, i'=i \text{ and } a'\neq a \\ 0 & \text{if } s'\neq s \text{ or } i'\neq i \end{cases}$$

$$\frac{\partial H^V_{sia}(\boldsymbol{\sigma},\boldsymbol{V},\boldsymbol{\nu},\eta,t)}{\partial \eta} = (1-t)\Bigg(\nu_{sia}+\sigma_{sia}\sum_{a'\in A_{si}}\nu_{sia'}\Big[\log(\sigma_{sia'})-1\Big]\Bigg)$$

$$\begin{aligned}\frac{\partial H^V_{sia}(\boldsymbol{\sigma},\boldsymbol{V},\boldsymbol{\nu},\eta,t)}{\partial t} = \sigma_{sia}\Bigg(& u_{si}(a,\boldsymbol{\sigma}_{s,\text{-}i})-u_{si}(a,\boldsymbol{\rho}_{s,\text{-}i}) \\ & +\ \delta_i\sum_{s'\in S}\Big[\phi_{s\to s'}(a,\boldsymbol{\sigma}_{s,\text{-}i})-\phi_{s\to s'}(a,\boldsymbol{\rho}_{s,\text{-}i})\Big]V_{s'i}\Bigg) \\ & -\ \eta\Bigg(\nu_{sia}+\sigma_{sia}\sum_{a'\in A_{si}}\nu_{sia'}\Big[\log(\sigma_{sia'})-1\Big]\Bigg)\end{aligned}$$

$$\frac{\partial H^\sigma_{si}(\boldsymbol{\sigma},\boldsymbol{V},\boldsymbol{\nu},\eta,t)}{\partial \sigma_{s'i'a'}} = \begin{cases} 1 & \text{if } s'=s \text{ and } i'=i \\ 0 & \text{else} \end{cases}$$

$$\frac{\partial H^\sigma_{si}(\boldsymbol{\sigma},\boldsymbol{V},\boldsymbol{\nu},\eta,t)}{\partial V_{s'i'}} = \frac{\partial H^\sigma_{si}(\boldsymbol{\sigma},\boldsymbol{V},\boldsymbol{\nu},\eta,t)}{\partial \nu_{s'i'a'}} = \frac{\partial H^\sigma_{si}(\boldsymbol{\sigma},\boldsymbol{V},\boldsymbol{\nu},\eta,t)}{\partial \eta} = \frac{\partial H^\sigma_{si}(\boldsymbol{\sigma},\boldsymbol{V},\boldsymbol{\nu},\eta,t)}{\partial t} = 0$$

3.E Regularity: Parametrized Sard's Theorem, Full Rank of Jacobian

This section provides the proofs for Proposition 4 and Lemma 8.1. To that end, we will show that a generalization of Sard's theorem, known as parametrized Sard's theorem, applies to H. This proves that for generic $\boldsymbol{\nu}$, $\mathbf{0}$ is a regular value of H.

To give an intuitive idea of this result, suppose one picked $\boldsymbol{\nu}$ such that the solution set to $H|_{\boldsymbol{\nu}} = \mathbf{0}$ contained singularities; then these singularities must be unstable and disappear with probability 1 when using a slightly perturbed vector $\boldsymbol{\nu}+\boldsymbol{\epsilon}$ instead. In consequence, it is sufficient to pick an appropriately randomized

$\boldsymbol{\nu}$ to ensure regularity. (For the purpose of numerical computation of equilibria, this may not even be necessary. In our experience, simply setting all ν_{sia} to 1 poses no problem for numerical continuation. The only singularities that we then encountered in extensive testing were transversal bifurcations. These are unproblematic from a numerical perspective, as the path can simply be continued across such points. We failed to create genuinely problematic singularities such as higher-dimensional subsets contained in the solution set.)

We now turn to proving regularity. Parametrized Sard's theorem (Chow et al., 1978, Theorem 2.1, p. 891) reads:

> Let $\mathcal{Y} \subset \mathbb{R}^m$, $\mathcal{V} \subset \mathbb{R}^q$ be open and let $H : \mathcal{Y} \times \mathcal{V} \to \mathbb{R}^p$ be C^r, $r > \max\{0, m-p\}$. If $\mathbf{0} \in \mathbb{R}^p$ is a regular value of H, i.e. if the Jacobian J satisfies $\operatorname{rank}(J(\boldsymbol{y}, \boldsymbol{\nu})) = p$ for all $(\boldsymbol{y}, \boldsymbol{\nu}) \in H^{-1}(\mathbf{0})$, then for almost every $\boldsymbol{\nu} \in \mathcal{V}$, $\mathbf{0}$ is a regular value of $H_{\boldsymbol{\nu}}(\cdot) = H(\boldsymbol{\nu}, \cdot)$.

The theorem applies to H on Y. First, H is smooth (C^∞), so that the differentiability requirement is met. Second, take $\boldsymbol{\nu} \in \mathcal{V} := \mathbb{R}^{|A|}_{>0}$ (or an arbitrary open subset thereof). Third, while Y is itself not open as required, one can easily extend it to an open domain $\mathcal{Y}$ for $H(\boldsymbol{\sigma}, \boldsymbol{V}, t)$, for example by setting

$$\mathcal{Y} := (0, 1{+}\varepsilon)^{|A|} \times \mathbb{R}^{|S \times I|} \times (-\varepsilon, 1)$$

for some $\varepsilon > 0$. Finally, the following will show that the Jacobian J has full rank on $(\mathcal{Y} \times \mathcal{V}) \cap H^{-1}(\mathbf{0})$.

The Jacobian J of H is written out in Appendix 3.D. It has the following block structure:

$$J(\boldsymbol{\sigma}, \boldsymbol{V}, \boldsymbol{\nu}, t) \quad = \quad \begin{pmatrix} \frac{\partial H^V_{sia}}{\partial \sigma_{s'i'a'}} & \frac{\partial H^V_{sia}}{\partial V_{s'i'}} & \frac{\partial H^V_{sia}}{\partial \nu_{s'i'a'}} & \frac{\partial H^V_{sia}}{\partial t} \\ \frac{\partial H^\sigma_{si}}{\partial \sigma_{s'i'a'}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

For detailed contents of the blocks, again refer to Appendix 3.D. To establish full row rank, first consider the block

$$\frac{\partial H^\sigma_{si}(\boldsymbol{\sigma}, \boldsymbol{V}, \boldsymbol{\nu}, t)}{\partial \sigma_{s'i'a'}} = \begin{pmatrix} 1 \ldots 1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & 1 \ldots 1 \end{pmatrix} \in \mathbb{R}^{|S \times I| \times |A|}$$

whose rows clearly are independent. Next, consider the block

$$\frac{\partial H^V_{sia}(\boldsymbol{\sigma}, \boldsymbol{V}, \boldsymbol{\nu}, t)}{\partial \nu_{s'i'a'}} = \begin{pmatrix} B_{1,1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & B_{|S|,|I|} \end{pmatrix} \in \mathbb{R}^{|A|\times|A|}$$

which is itself comprised of quadratic blocks B_{si}. If an agent (s, i) has only a single action, then $B_{si} = \begin{pmatrix} 0 \end{pmatrix}$: These cases will be covered later. If otherwise $|A_{si}| \geq 2$, then

$$B_{si} = (1-t)\eta \begin{pmatrix} 1 + \sigma_{si1}(\log \sigma_{si1} - 1) & \sigma_{si2}(\log \sigma_{si2} - 1) & \dots & \sigma_{si|A_{si}|}(\log \sigma_{si|A_{si}|} - 1) \\ \sigma_{si1}(\log \sigma_{si1} - 1) & 1 + \sigma_{si2}(\log \sigma_{si2} - 1) & \dots & \sigma_{si|A_{si}|}(\log \sigma_{si|A_{si}|} - 1) \\ \vdots & & \ddots & \vdots \\ \sigma_{si1}(\log \sigma_{si1} - 1) & \sigma_{si2}(\log \sigma_{si2} - 1) & \dots & 1 + \sigma_{si|A_{si}|}(\log \sigma_{si|A_{si}|} - 1) \end{pmatrix}$$

After subtracting the first row from each other row, one obtains the following arrowhead matrix:

$$\begin{aligned} B_{si} &= (1-t)\eta \left(\begin{array}{c|c} D & E \\ \hline F & G \end{array}\right) \\ &= (1-t)\eta \left(\begin{array}{c|ccc} 1 + \sigma_{si1}(\log \sigma_{si1} - 1) & \sigma_{si2}(\log \sigma_{si2} - 1) & \dots & \sigma_{si|A_{si}|}(\log \sigma_{si|A_{si}|} - 1) \\ \hline -1 & 1 & & \mathbf{0} \\ \dots & & \ddots & \\ -1 & \mathbf{0} & & 1 \end{array}\right) \end{aligned}$$

Because G is invertible, the determinant of each such block can be computed using the Schur complement:

$$\begin{aligned} \det(B_{si}) &= (1-t)\eta \ \det\Big(G\Big) \ \det\Big(D - EG^{-1}F\Big) \\ &= (1-t)\eta \left(1 + \sigma_{si1}(\log \sigma_{si1} - 1) - \sum_{k=2}^{|A_{si}|} -\sigma_{sik}\left(\log \sigma_{sik} - 1\right)\right) \\ &= (1-t)\eta \sum_{k=1}^{|A_{si}|} \sigma_{sik} \log \sigma_{sik} \ < \ 0 \end{aligned}$$

The inequality follows from $t \in [0, 1)$, $\eta > 0$, and $\sigma_{sia} \in (0, 1)$. The blocks B_{si} taken together thus provide a basis for all rows that do *not* correspond to singleton actions.

To complete the proof for these, we use $\frac{\partial H^V_{sia}(\boldsymbol{\sigma}, \boldsymbol{V}, \boldsymbol{\nu}, t)}{\partial V_{s'i'}}$. Note that all entries are zero for $i \neq i'$, so that one can treat players separately. Fix any i and consider only those rows and columns corresponding to a state in which i has only a single action, enumerated as $s_1, s_2, \ldots, s_n$. In these cases, $\sigma_{sia} = 1$, so that the resulting submatrix is

$$\delta_i \bar{\boldsymbol{\phi}}_i - \boldsymbol{I}_n = \begin{pmatrix} \delta_i \bar{\phi}_{s_1 \to s_1} - 1 & \delta_i \bar{\phi}_{s_1 \to s_2} & \ldots & \delta_i \bar{\phi}_{s_1 \to s_n} \\ \delta_i \bar{\phi}_{s_2 \to s_1} & \delta_i \bar{\phi}_{s_2 \to s_2} - 1 & & \ldots \\ \ldots & & \ddots & \\ \delta_i \bar{\phi}_{s_n \to s_1} & \ldots & & \delta_i \bar{\phi}_{s_n \to s_n} - 1 \end{pmatrix}$$

We have $\delta_i < 1$, and because $\bar{\boldsymbol{\phi}}_i$ is part of a transition matrix, all entries are between 0 and 1, with row sums less or equal to 1. This implies $||\delta_i \bar{\boldsymbol{\phi}}_i||_\infty < 1$ in maximum absolute row sum norm. Therefore, the above matrix is invertible, with

$$\left(\delta_i \bar{\boldsymbol{\phi}}_i - \boldsymbol{I}_n\right)^{-1} = -\sum_{m=0}^{\infty} \left(\delta_i \bar{\boldsymbol{\phi}}_i\right)^m$$

For each player, this gives a basis for the rows corresponding to singleton action sets. Since all rows of J are covered now, the proof is complete.

Note that all preceding arguments apply just as well if one considers η as an argument of H, rather than a fixed parameter, provided $\eta > 0$. The Jacobian $J(\boldsymbol{\sigma}, \boldsymbol{V}, \boldsymbol{\nu}, \eta, t)$ likewise has full row rank for all $(\boldsymbol{\sigma}, \boldsymbol{V}, t, \eta) \in \mathcal{Y} \times (0, \infty)$. Application of parametrized Sard's theorem implies that for generic $\boldsymbol{\nu}$, the zero set of $H(\boldsymbol{\sigma}, \boldsymbol{V}, \eta, t)$ is a smooth, 2-dimensional manifold in the interior of $Y \times (0, \infty)$. This proves Lemma 8.2, which in turn is part of the proof of Proposition 8.

3.F ODE Representation of L^η

Here we give an explicit representation of the curve L^η, parametrized in arc length. This is done in the form of an ordinary differential equation (ODE); for a general discussion of the results used here, see e.g. Zangwill and Garcia (1981).

To simplify notation for this purpose, we collect all the variables σ_{sia}, V_{si}, and t in a vector $\boldsymbol{x} \in \mathbb{R}^{N+1}$, where $N = |A| + |S \times I|$. Similarly, enumerate the components of H as $H_1, ..., H_N$. The Jacobian of H with respect to $\boldsymbol{x}$ is then a matrix J of dimensions $N \times (N+1)$. We denote by $J_{\text{-}n}$ the square submatrix obtained by dropping the nth column from J.

By a well known application of the implicit function theorem, one can traverse any solution path contained in $H^{-1}(\mathbf{0})$ by means of a system of ordinary differential equations constructed from J, provided that J is of full rank N along this path. This system is given by $N+1$ equations

$$\dot{x}_n = (-1)^n \det\Big(J_{\text{-}n}(\boldsymbol{x})\Big)$$

Note that $\dot{\boldsymbol{x}}$ is simply a tangent vector of the path in point $\boldsymbol{x}$. For a detailed exposition and deductions, see Zangwill and Garcia (1981, equation 2.1.2, p. 26).

Since $\mathbf{0}$ is a regular value of $H|_Y$, this applies to all paths contained in Z, and in particular to the distinguished path starting at $(\boldsymbol{\sigma}^0, V^0, 0) =: \boldsymbol{x}^0$. All points on this path can be represented as the solution $\boldsymbol{x}(S)$ to an initial value problem, given by $\boldsymbol{x}(0) := \boldsymbol{x}^0$, and

$$x_n(S) = x_n^0 \pm \int_0^S \frac{\dot{x}_n(s)}{\|\dot{\boldsymbol{x}}(s)\|} ds$$

where the sign before the integral is chosen such that t initially increases. The vector field is normalized, so that the curve is parametrized in arc length, and S simply represents distance traveled along the curve.

This ODE can be used to compute the distinguished equilibrium numerically. Finding a stationary equilibrium of some game $\mathcal{G}$ is generally a hard task, as it involves solving a high-dimensional system of nonlinear equations. Standard methods to do so require a good initial guess, which will usually not be available. Homotopy methods such as the present one circumvent this problem: By construction, the starting point is easy to compute. All that is left to do then is to calculate a sequence of points along the distinguished path, a task that is much easier, because each such step can start from a solution that is very close by. In effect, the global task of finding an equilibrium of $\mathcal{G}$ is transformed into a sequence of local tasks.

Note that the prior vector can be chosen freely, for example as the centroid or some other focal strategy. Alternatively, one can perform the procedure on a collection of priors, e.g. by constructing a grid over the prior space. The procedure will potentially find different equilibria for different priors, which allows to assess the respective sizes of the basins of attraction of different equilibria.

3.G Timings

3.G.1 Timings for Non-Generic Randomized Games

Here, we report timings for randomized non-generic games, to complement the timings of similar, but generic games in Section 3.6. General procedures and the computer used where as described there, except for the following changes. Values of u were drawn from a discrete uniform distribution with support $\{0, 0.1, \ldots, 1\}$. For each action profile a_s in any state, the vector of probabilities for the resulting states, $\Big(\phi_{s \to s_1}(a_s), \phi_{s \to s_2}(a_s), ..., \phi_{s \to s_{|S|}}(a_s)\Big)$, was created using a uniform multinomial distribution with $2|S|$ trials; this vector was then normalized so that the result sums to 1. To obtain generic (ν_{sia}), values were drawn from a uniform distribution over the interval $[0.75, 1.25]$.

Results are listed in Table 3.2. In comparison to the timings for generic games in Table 3.1, computation times are 11% slower on average (the difference is statistically significant with $p < 0.001$, in a regression of logarithmized running time on game type, using fixed effects for the respective game sizes). As for the generic games, less than 1% of all runs are initially not successful when using a set of default path tracking parameters.

3.G.2 Instructions for Replication

Both sets of timings can be replicated by following these steps:

1. Install the anaconda python distribution (`anaconda.com`), if you haven't yet. Then set up a virtual environment and install the required packages by running in a system terminal:

   ```
   conda create -name logtracing-env python=3.9
   conda activate logtracing-env
   conda install scipy numpy cython openpyxl matplotlib pandas
   pip install sgamesolver==1.0.2
   ```

2. From the online supplement to this article, download the files `LogTracing_Generic.xlsx` and `LogTracing_NonGeneric.xslx`. These files contain all the raw data from the runs performed by us, for generic and non-generic games respectively. To repeat all runs from either file, open it in Microsoft Excel or a compatible program and delete all rows but the first from the sheet named *Runs*. (Each set of timings took about 2 weeks on the computer

$\lvert S\rvert$	$\lvert A_{si}\rvert$	$\lvert I\rvert$			
		2	3	4	5
1	2	0:00 0:00	0:00 0:00	0:00 0:00	0:00 0:00
	4	0:00 0:00	0:00 0:00	0:00 0:00	0:00 0:00
	8	0:00 0:00	0:00 0:00	0:00 0:00	0:02 0:00
2	2	0:00 0:00	0:00 0:00	0:00 0:00	0:00 0:00
	4	0:00 0:00	0:00 0:00	0:00 0:00	0:00 0:00
	8	0:00 0:00	0:00 0:00	0:01 0:00	0:05 0:02
5	2	0:00 0:00	0:00 0:00	0:00 0:00	0:00 0:00
	4	0:00 0:00	0:00 0:00	0:00 0:00	0:02 0:01
	8	0:00 0:00	0:01 0:01	0:04 0:02	0:53 0:29
10	2	0:00 0:00	0:00 0:00	0:00 0:00	0:00 0:00
	4	0:00 0:00	0:01 0:00	0:02 0:01	0:06 0:07
	8	0:01 0:00	0:04 0:02	0:21 0:11	4:04 2:45
20	2	0:00 0:00	0:00 0:00	0:01 0:01	0:02 0:01
	4	0:00 0:00	0:03 0:03	0:15 0:11	0:59 0:48
	8	0:04 0:03	0:47 0:43	5:53 2:54	51:24 35:21
50	2	0:01 0:00	0:04 0:02	0:11 0:04	0:39 0:24
	4	0:10 0:07	1:45 1:35	12:30 9:57	57:50 29:02
	8	2:42 2:28	1:03:38 33:57		
100	2	0:05 0:01	0:29 0:19	3:00 4:09	11:53 8:42
	4	1:57 1:42	33:17 26:03		
	8	55:40 48:32			
200	2	0:34 0:11	6:47 6:21	52:59 41:24	6:40:27 4:01:59
	4	50:39 35:59			
400	2	5:06 1:59			
800	2	49:14 22:34			

Table 3.2: Computation times to solve random non-generic games with $|S|$ states, $|I|$ players and $|A_{si}|$ actions for each player in each state. Listed are average times as well as standard deviations (in small print) in *m*:*ss* or *h*:*mm*:*ss*. All timings under 15:00 are based on 100 independently drawn games of the respective size; all others on 10 runs per size.

used by us.) To repeat computation only for specific games, delete only the according rows from the sheet. Save the file and close Excel.

3. To start computation, open a system terminal and navigate to the location of the files. Then run

   ```
   conda activate logtracing-env
   sgamesolver-timings FILENAME.xlsx
   ```

   replacing the filename accordingly. The program will begin solving the games and keep you updated about its progress. Every 5 minutes, all finished runs will be saved to the Excel file; make sure that it is not opened in Excel while computation is in progress, as this will lock writing access. Computations can be canceled by pressing `CTRL+C` at any time. Restarting from the last save is possible by running the commands again.

4. You can check the new results in the *Summary* sheet, which is updated whenever the file is saved. Note that the summary always includes all rows that appear in the *Runs* sheet (e.g. those rows you left from our runs, or from your own previous sessions). If you are interested in individual games only, check the according rows in the *Runs* sheet directly. To obtain a latex table as used in this article, run

   ```
   conda activate logtracing-env
   sgamesolver-timings -l FILENAME.xlsx
   ```

   which will create or update a `FILENAME.tex` file in the current folder.

References

Abbring, J. H., J. R. Campbell, J. Tilly, and N. Yang (2018): “Very Simple Markov-Perfect Industry Dynamics: Theory,” *Econometrica*, 86, 721–735.

Besanko, D., U. Doraszelski, Y. Kryukov, and M. Satterthwaite (2010): “Learning-by-Doing, Organizational Forgetting, and Industry Dynamics,” *Econometrica*, 78, 453–508.

Bourbaki, N. (1966): *Elements of Mathematics. General Topology. Part 1*, Paris and Reading: Hermann and Addison-Wesley.

Chow, S.-N., J. Mallet-Paret, and J. A. Yorke (1978): “Finding Zeroes of Maps: Homotopy Methods That Are Constructive With Probability One,” *Mathematics of Computation*, 32, 887–899.

Dang, C., P. J.-J. Herings, and P. Li (2022): “An Interior-Point Differentiable Path-Following Method to Compute Stationary Equilibria in Stochastic Games,” *INFORMS Journal on Computing*, 34, 1403–1418.

Daskalakis, C., P. W. Goldberg, and C. H. Papadimitriou (2009): “The Complexity of Computing a Nash Equilibrium,” *SIAM Journal of Computing*, 39, 195–259.

Doraszelski, U. and K. L. Judd (2012): “Avoiding the Curse of Dimensionality in Dynamic Stochastic Games,” *Quantitative Economics*, 3, 53–93.

Doraszelski, U. and A. Pakes (2007): “A Framework for Applied Dynamic Analysis in IO,” in *Handbook of industrial organization, Vol. 3*, ed. by M. Armstrong and R. H. Porter, Amsterdam: Elsevier, 1887–1966.

Doraszelski, U. and M. Satterthwaite (2010): “Computable Markov-Perfect Industry Dynamics,” *The RAND Journal of Economics*, 41, 215–243.

Eaves, B. C. and K. Schmedders (1999): “General Equilibrium Models and Homotopy Methods,” *Journal of Economic Dynamics and Control*, 23, 1249–1279.

Eibelshäuser, S. and D. Poensgen (2019): “dsGameSolver: A Python Program for Computing Markov Perfect Equilibria of Dynamic Stochastic Games,” Working paper.

——— (2020): "Markov Quantal Response Equilibrium and a Homotopy Method for Computing and Selecting Stationary Equilibria of Stochastic Games," Working paper.

Ericson, R. and A. Pakes (1995): "Markov-Perfect Industry Dynamics: A Framework for Empirical Work," *Review of Economic Studies*, 62, 53–82.

Fink, A. M. (1964): "Equilibrium in a Stochastic n-Person Game," *Journal of Science of the Hiroshima University, Series A-I (Mathematics)*, 28, 89–93.

Gilboa, I. and E. Zemel (1989): "Nash and correlated equilibria: Some complexity considerations," *Games and Economic Behavior*, 1, 80–93.

Goettler, R. L., C. A. Parlour, and U. Rajan (2005): "Equilibrium in a Dynamic Limit Order Market," *The Journal of Finance*, 60, 2149–2192.

Govindan, S. and R. Wilson (2009): "A Global Newton Method for Stochastic Games," *Journal of Economic Theory*, 144, 414–421.

Harsanyi, J. C. (1975): "The Tracing Procedure: A Bayesian Approach to Defining a Solution for n-Person Noncooperative Games," *International Journal of Game Theory*, 4, 61–94.

Harsanyi, J. C. and R. Selten (1988): *A General Theory of Equilibrium Selection in Games*, Cambridge, Massachusetts: MIT Press.

Herings, P. J.-J. and R. J. Peeters (2003): "Equilibrium Selection in Stochastic Games," *International Game Theory Review*, 5, 307–326.

——— (2004): "Stationary Equilibria in Stochastic Games: Structure, Selection, and Computation," *Journal of Economic Theory*, 118, 32–60.

Hovanskii, A. G. (1980): "On a Class of Systems of transcendental Equations," *Soviet mathematics - Doklady*, 22, 762–765.

Kohlberg, E. and J.-F. Mertens (1986): "On the Strategic Stability of Equilibria," *Econometrica*, 54, 1003–1037.

Levhari, D. and L. J. Mirman (1980): "The Great Fish War: An Example Using a Dynamic Cournot-Nash Solution," *The Bell Journal of Economics*, 11, 322–334.

Marker, D. (1996): “Model Theory and Exponentiation,” *Notices of the AMS*, 43, 753–759.

Maskin, E. and J. Tirole (1988a): “A Theory of Dynamic Oligopoly I: Overview and Quantity Competition with Large Fixed Costs,” *Econometrica*, 56, 549–569.

——— (1988b): “A Theory of Dynamic Oligopoly II: Price Competition, Kinked Demand Curves, and Edgeworth Cycles,” *Econometrica*, 56, 571–599.

McKelvey, R. D. and T. R. Palfrey (1995): “Quantal Response Equilibria for Normal Form Games,” *Games and Economic Behavior*, 10, 6–38.

Nagel, R. (1995): “Unraveling in Guessing Games: An Experimental Study,” *The American Economic Review*, 85, 1313–1326.

Pakes, A. and P. McGuire (1994): “Computing Markov Perfect Nash Equilibria: Numerical Implications of a Dynamic Differentiated Product Model,” *RAND Journal of Economics*, 25, 555–589.

Rubinstein, R. Y. and B. Melamed (1998): *Modern Simulation and Modeling*, New York: Wiley.

Rudin, W. (1976): *Principles of Mathematical Analysis. Third Edition*, New York: McGraw-Hill.

Schanuel, S. H., L. K. Simon, and W. R. Zane (1991): “The Algebraic geometry of Games and the Tracing Procedure,” in *Game Equilibrium Models II: Methods, Morals, and Markets*, ed. by R. Selten, Heidelberg: Springer, 9–43.

Shapley, L. S. (1953): “Stochastic Games,” *Proceedings of the National Academy of Sciences*, 39, 1095–1100.

Solan, E. and N. Vieille (2015): “Stochastic games,” *Proceedings of the National Academy of Sciences*, 112, 13743–13746.

Stahl, D. O. and P. W. Wilson (1995): “On Players’ Models of Other Players: Theory and Experimental Evidence,” *Games and Economic Behavior*, 10, 218–254.

Takahashi, M. (1964): “Equilibrium Points of Stochastic, Noncooperative n-Person Games,” *Journal of Science of the Hiroshima University, Series A-I (Mathematics)*, 28, 95–99.

WATSON, L. T. (2002): "Probability-one homotopies in computational science," *Journal of Computational and Applied Mathematics*, 140, 785–807.

ZANGWILL, W. I. AND C. B. GARCIA (1981): *Pathways to Solutions, Fixed Points, and Equilibria*, Upper Saddle River, New Jersey: Prentice-Hall.

Chapter **4**

sgamesolver: A Python Package to Solve Stochastic Games

This chapter is based on joint work with Steffen Eibelshäuser.

Abstract: We introduce sgamesolver, a python package that uses the homotopy method to compute stationary equilibria of finite discounted stochastic games. A short user guide is complemented with discussion of the homotopy method, the two implemented homotopy functions *logit Markov QRE* and *logarithmic tracing*, and the predictor-corrector procedure and its implementation in sgamesolver. Basic and advanced use cases are demonstrated using several example games. Finally, we discuss the topic of symmetries in stochastic games.

4.1 Introduction

Stochastic games combine two core motifs of modern economic analysis: intertemporal optimization and strategic interaction, and thereby capture the essence of many situations studied in applied economics. They have been used to model, for example, dynamic pricing (Maskin and Tirole, 1988), industry dynamics (Ericson and Pakes, 1995; Abbring et al., 2018), limit order markets (Goettler et al., 2005), research and development (Breitmoser et al., 2010), organizational learning (Besanko et al., 2010), and routing and queueing systems (Altman, 1996).

However, adoption remains slow and there is large, untapped potential (Herings and Peeters, 2004; Solan and Vieille, 2015). The likely reason is that, as easy as it is to find applications for stochastic games, as hard is it to solve the resulting models. Even smaller stochastic games are generally complex enough to defy analytical solution. Unfortunately, computing numerical solutions also comes with difficulties, which we detail below. In particular, there has been a lack of ready-to-use tools and programs that are widely applicable, accessible, and sufficiently fast. In this paper, we introduce sgamesolver, a python package that aims to address this gap.

It is the combination of size and strategic interaction that makes stochastic games hard to solve. Even for games of moderate size, equilibrium conditions quickly consist in hundreds, if not thousands, of nonlinear equations and inequalities that have to be solved simultaneously. This issue is shared by Markov decision problems, which are essentially one-player stochastic games with much of the same structure. Yet, these can be solved using powerful iterative methods. Unfortunately, these methods typically fail to converge once strategic interaction is introduced. Even in games where they do converge, they are only able to find pure, but not mixed strategy equilibria. Likewise, many solution methods classically used in game theory fall short when applied to stochastic games. Without a final period, backward induction is impossible. Support enumeration fails due to curse of dimensionality. For example, consider a game with 20 states, 5 players, and 8 actions per state and player. This size is not unreasonable by any means, but it implies almost 10^{190} potential supports for stationary equilibria. Thus, any technique enumerating supports or complementary slackness conditions is not viable – and would not be even if resolving the equilibrium conditions after guessing a correct support was completely free, which is actually a very hard problem in itself.

In light of this, homotopy methods seem to be the most promising approach to compute stationary equilibria in general stochastic games. This method solves a mathematical problem by first continuously transforming it to a similar, but much simpler problem. The transformation is then gradually reversed while tracking a path of solutions, until the desired solution of the original problem is obtained. Intuitively, this turns a very hard, global problem into a series of much easier, local problems. This is also the method sgamesolver is based on.

The package improves on existing tools in various ways. The first is generality. sgamesolver can in principle solve all games in this class, so that the only limiting factor is size. Crucially, this allows to take modeling decisions without being constrained by the solution technique. The most important alternative to homotopy methods are iteration based algorithms, which have been used with great success in the literature on industry dynamics, going back to Pakes and McGuire (1994). However, as mentioned before, iteration based methods come with no convergence guarantee for stochastic games. Where they do work, they are rather fast and able to handle quite large state spaces; but whether they are able to solve a particular game will depend on its specific structure. Moreover, they are able to find pure strategy equilibria only – which often do not exist.[1] Homotopy methods are able to overcome these limitations; however, existing implementations came with other important limitations. For example, the program provided by Herings and Peeters (2004) requires games to be "rectangular" in that all players must have the same number of actions in all states. Moreover, it may fail for games that are not generic – e.g. games with symmetries or payoffs that are evenly spaced.

The next important improvement is speed: The fastest comparable algorithm for which we could find reported timings is the interior point method by Dang et al. (2022). The largest games solved there contain 5 states, 5 players, and 8 actions per state and player. For these, sgamesolver is faster by a factor of 500, improving average time from 7.5 hours to less than a minute. And due to its speed, it is able to solve much larger games still in reasonable time (see Chapter 3 for details).

Another essential aspect is usability. sgamesolver's target audience are applied researchers; our intention has been to make the program usable without background knowledge in scientific computation. sgamesolver is based on python, a free and very accessible programming language. Furthermore, the most com-

[1] But see Doraszelski and Satterthwaite (2010), who use a purification technique to allow for mixing of at least one class of players, thereby alleviating this restriction somewhat.

plex step in using it is to define the stochastic game to be solved. To make this as convenient as possible, sgamesolver allows to pass the game in form of a table, which can be generated in another language or program of choice, such as Excel, Stata, or R.

Finally, sgamesolver is modular in structure and designed to be easily extensible. For example, it currently includes two distinct homotopy functions – others can easily be added at any time, both by the authors, but also by users. Similarly, the part of the package that implements the actual numerical path tracking is a separate, self-contained sub-module. This allows for easy adaptations and improvements in the future. One could use the homotopy functions provided by sgamesolver with a completely different path tracking algorithm. Or conversely, one can use the path tracking algorithm to solve homotopies completely unrelated to stochastic games.

4.1.1 First Steps and Overview

sgamesolver is written in python and cython. It is free and open source, published under the permissive MIT license. The current version at the time of this writing is 1.0.2. Source code is found on `github.com/davidpoensgen/sgamesolver`, alongside an issue tracker. Documentation is available at `sgamesolver.readthedocs.io`. The package is hosted on the python repository PyPi, so installation is as easy as running

```
pip install sgamesolver
```

in a system terminal (for details, see the online documentation). We recommend to use sgamesolver with the anaconda python distribution (`anaconda.com`), which is often considerably faster than the official distribution for numerical tasks.

The following minimal working example illustrates the core steps in using sgamesolver:

```python
import sgamesolver
game = sgamesolver.SGame.random_game(64, 2, 4, seed=42)
homotopy = sgamesolver.homotopy.QRE(game)
homotopy.solver_setup()
homotopy.solve()
print(homotopy.equilibrium)
homotopy.equilibrium.simulate()
```

We will briefly go over these steps, along which also the rest of the paper is organized. The first is always to define a game [line 2]. To keep things simple, the example uses a randomly generated game (here with 64 states, 2 players, and 4 actions per state and player). Of course, the goal will usually be to solve a specific game; Section 4.2 will cover this in detail, where we formally define stochastic games, discuss examples, and show how they can be passed to sgamesolver. The second step is to pick a homotopy function [3]. sgamesolver currently implements two: *logarithmic tracing* and *QRE*. The homotopy principle, the role of the homotopy function, and these two alternatives are discussed in Section 4.3. Further examples then cover some advanced use cases. The third step is to set up and start the solver [4–5], which computes the equilibrium by numerically tracking the path defined by the homotopy function. sgamesolver uses a predictor-corrector-algorithm to do so; the basic idea of this method and its implementation are laid out in Section 4.4. Ideally, the solver is able to arrive at the desired equilibrium with its default parameters and without further user interaction; however, Section 4.4 also contains some guidance in case of failure. Once the equilibrium is reached, one can output strategies and values [6] or use them for further computations or plotting. sgamesolver also allows to simulate equilibrium play [7]. In Section 4.5, we discuss symmetries in stochastic games, which are not touched upon in the above example, but are of twofold importance: Symmetry is a common selection criterion for equilibria; and it may be used to speed up computations. Section 4.6 concludes.

4.2 Stochastic Games

Stochastic games are a quite general class of games that can be seen to generalize either Markov decision processes to multiple agents, or repeated games by the addition of state transitions. A stochastic game is played as follows: First, an initial state is determined, possibly according to a random distribution. At the beginning of each stage, all players learn the current state of the world and then choose one of their available actions. All actions and the current state jointly determine instantaneous utilities for each player and a probability distribution from which the next state is drawn. The next period begins accordingly. A game may involve terminal states, meaning it will end once such a state is reached; otherwise, the game will continue indefinitely. Players discount exponentially from period to period.

Definition: Stochastic Game. A stochastic game is given by a tuple $\mathcal{G} = \Big(S, I, A, \boldsymbol{u}, \boldsymbol{\Phi}, \boldsymbol{\Phi}_0, \boldsymbol{\delta}\Big)$, with

S : set of states.

I**:** set of players.

A_{si}**:** action set of player i in state s. $A_s = \bigtimes_{i \in I} A_{si}$ is the set of action profiles in state s. $A = \bigcup_{s \in S, i \in I} A_{si}$ denotes the set of all actions of any player in any state (understood as a disjoint union). Thus, $|A|$ represents the total number of actions of the game. We often use the index ${}_{sia}$ to refer to an action a that belongs to player i in state s.

$\boldsymbol{u} = \Big(u_{si}(\boldsymbol{a}_s)\Big)_{\boldsymbol{a}_s \in A_s, s \in S, i \in I}$**:** instantaneous payoff functions $u_{si} : A_s \to \mathbb{R}$.

$\boldsymbol{\Phi} = \Big(\phi_{s \to s'}(\boldsymbol{a}_s)\Big)_{\boldsymbol{a}_s \in A_s, s, s' \in S}$**:** state transition probabilities, where $\phi_{s \to s'}(\boldsymbol{a}_s)$ denotes the probability of transitioning from state s to s', if action profile $\boldsymbol{a}_s$ is played. Note that it may be that $\sum_{s' \in S} \phi_{s \to s'}(\boldsymbol{a}_s) < 1$; the remaining probability mass is then simply the chance of the game to terminate.

$\boldsymbol{\Phi}_0 \in \Delta(S)$**:** a probability distribution over the initial state.

$\boldsymbol{\delta} = \Big(\delta_i\Big)_{i \in I}$**:** discount factors for all players.

A stochastic game is called finite if S, I, and A are finite; the time horizon is of course still infinite. A game is called discounted if all $\delta_i < 1$ or state transitions $\boldsymbol{\Phi}$ are such that that the game eventually terminates with probability one. sgamesolver can solve finite discounted games only. These restrictions reflect fundamental limitations of the homotopy method: First, with continuous action or state spaces, strategies can generally not be represented as finite vectors of real numbers, so that homotopy methods are not applicable.[2] Second, while there exist equilibrium concepts for undiscounted games, the equilibrium strategies are generally not stationary, but depend on history and/or time (Mertens et al., 2015), so that representation by a finite vector is again not possible. At least in applied economics, discounting is a common assumption, so that this limitation should not be overly restrictive.

[2] One can of course use sgamesolver on a discretized version of the game, provided that the size of the resulting state and action sets does not become too large. In special cases, it may alternatively be possible to parameterize strategies by a finite vector of reals, and then solve for these using the homotopy method.

In Section 4.2.1, we will introduce two examples of stochastic games to flesh out the concept, and also to illustrate how to represent games in sgamesolver. Before that, we will introduce the important concepts of stationary strategies and stationary equilibrium.

Definition: Stationary Strategy. A stationary strategy $\boldsymbol{\sigma}$ assigns to each pair $(s, i) \in S \times I$, called the *agent* of player i in state s, a probability distribution $\boldsymbol{\sigma}_{si} \in \Delta(A_{si})$ over the available actions A_{si}.

Thus, stationary strategies are conditional on the current state only, but not on the history of play nor on time. Each stationary strategy profile $\boldsymbol{\sigma}$ induces a unique vector of state-player-values V_{si}, satisfying the recursive relation typical in intertemporal optimization:

$$V_{si} = u_{si}(\boldsymbol{\sigma}_s) + \delta_i \sum_{s' \in S} \phi_{s \to s'}(\boldsymbol{\sigma}_s)\, V_{s'i}$$

One can obtain a vector equation for the values as follows. Suppose the states are labeled $S = \{s_0, s_1, s_2, ..., s_N\}$ For the vector of values of player i, write $\boldsymbol{V}_i = (V_{s_0 i}, V_{s_1 i}, ...,)'$, and for the vector of instantaneous utilities under $\boldsymbol{\sigma}$ write $\boldsymbol{u}_i = (u_{s_0 i}(\boldsymbol{\sigma}_{s_0}), u_{s_1 i}(\boldsymbol{\sigma}_{s_1}), ...)$. Let $\boldsymbol{\Phi}$ be the transition matrix arising from $\boldsymbol{\sigma}$, so that $\boldsymbol{\Phi}_{m,n} = \phi_{s_m \to s_n}(\boldsymbol{\sigma}_{s_m})$. Then one obtains the vector equation

$$\boldsymbol{V}_i = (I - \delta_i \boldsymbol{\Phi})^{-1} \boldsymbol{u}_i = \sum_{t=0}^{\infty} (\delta_i \boldsymbol{\Phi})^t \boldsymbol{u}_i$$

For some purposes, it is helpful to introduce the following notation

$$U_{si}(\boldsymbol{\sigma}_s, \boldsymbol{V}_i) := u_{si}(\boldsymbol{\sigma}_s) + \delta_i \sum_{s' \in S} \phi_{s \to s'}(\boldsymbol{\sigma}_s)\, V_{s'i}$$

where U corresponds to total expected utility when $\boldsymbol{\sigma}_s$ is played in the current period, and continuation values are given by $\boldsymbol{V}$.

The most important solution concept for discounted stochastic games is stationary equilibrium.

Definition: Stationary Equilibrium. A stationary strategy profile $\boldsymbol{\sigma}$ is a stationary equilibrium if and only if for each player i, $\boldsymbol{\sigma}_i$ maximizes $\boldsymbol{V}_i$ given the strategies $\boldsymbol{\sigma}_{\text{-}i}$ of the other players.

We make a few comments. First, stationary equilibrium exists for all finite discounted stochastic games (Shapley, 1953; Fink, 1964; Takahashi, 1964). Second, stationary strategies always admit a stationary best response (Herings and

Peeters, 2004). Therefore, imposing a stationarity restriction on strategies does not introduce additional equilibria, but simply acts as a selection criterion from the set of all subgame perfect equilibria. Third, stationary equilibria obey the one-shot deviation principle, meaning that it is actually sufficient that no single agent (s, i) has a profitable deviation. Fourth, because the definition encompasses all states, including any that might never be reached in equilibrium, they are always subgame perfect.

Fifth, note that history-dependent strategies can always be made stationary by introducing additional states. The set of stationary equilibria may therefore depend on the exact formalization of a situation as a game. For example, consider the classical repeated prisoner's dilemma. Here, the unique stationary equilibrium is "always defect". Now, suppose one models the situation with two states instead, s_0 and s_1. The payoffs in each are exactly as before, and transitions as follows: From s_0, stay in s_0 if both players cooperate, and otherwise transition to s_1. From s_1, always stay in s_1. Here, a stationary equilibrium with cooperation exists, because the strategy "cooperate in s_0 and defect in s_1" – commonly known as trigger – is stationary under this formulation.

In light of this, there exists the refinement *Markov perfect equilibrium*, which tries to better capture the spirit of history-independence (Maskin and Tirole, 2001). It allows to condition strategies on states only insofar these differ in payoff-relevant terms.

Definition: Markov perfect equilibrium. Markov perfect equilibrium is a stationary equilibrium in which symmetric agents play the same strategy.

For the time being, we will leave the notion of symmetry somewhat vague: Essentially, two agents are symmetric if they face the same situation, in terms of payoffs and transitions. In the example above, there is nothing inherently different between s_0 and s_1, so that the unique Markov perfect equilibrium again is "always defect". A formal treatment of symmetry in stochastic games follows in Section 4.5; in the terminology introduced there, Markov perfect equilibrium is a stationary equilibrium that conforms to the game's maximal symmetry structure.

4.2.1 Defining Games in sgamesolver

In sgamesolver, stochastic games are represented by objects of the class `SGame`. To create a game, the relevant quantities $\boldsymbol{u}$, $\boldsymbol{\Phi}$, and $\boldsymbol{\delta}$ can be passed to sgamesolver either in form of a table or as NumPy arrays. In terms of functionality, both formats are equivalent: Anything that can be done in one can also be done in the other. However, they differ a bit in usability. The tabular format is more human-readable and likely more intuitive for many users. It also has the advantage that languages or programs other than python can easily be used to define a game, for example Excel, Stata, R – anything that allows to create data tables. This also makes sgamesolver accessible to users with little experience in python. The array format on the other hand is closer to the internal (and mathematical) representation of stochastic games. It is also more parsimonious and therefore better suited to handle very large games, where the tabular format may quickly result in file sizes in the order of gigabytes. We will focus here on a brief introduction of the tabular format; a more complete description, as well as documentation of the array format can be found online. We begin by introducing an example of a rather simple stochastic game.

Example 1: Rock, Paper, Laser Scissors. This game resembles the classic *Rock, Paper, Scissors*, with one added rule. Whenever a player uses scissors, they become "loaded": If used again in the very next period, they are able to beat the other player's scissors – unless those are loaded too, in which case the usual tie occurs. Performance against rock and paper is unchanged. A win pays 1, a tie 0, and a loss -1. The game runs an infinite number of rounds, and players discount future payoffs with $\delta = 0.95$.

The game can be modeled with three states: *Neutral*, where neither or both scissors are loaded, *Player 0 loaded*, and *Player 1 loaded*.[3] In this game, the number of actions and their labels in all states coincide; this is of course not a general property of stochastic games. The following table summarizes payoff information, with $X = 0$ in state *Neutral*, $X = 1$ in P_0 *loaded*, and $X = -1$ in P_1 *loaded*.

		Player 1		
		Rock	Paper	Scissors
	Rock	$0, 0$	$-1, 1$	$1, -1$
Player 0	Paper	$1, -1$	$0, 0$	$-1, 1$
	Scissors	$-1, 1$	$1, -1$	$X, -X$

[3] Being python-based, sgamesolver uses 0-indexing; we will also adapt this convention throughout the paper.

State transitions $\mathbf{\Phi}$ can be summarized as:

		Player 1		
		Rock	Paper	Scissors
Player 0	Rock	Neutral	Neutral	P1 loaded
	Paper	Neutral	Neutral	P1 loaded
	Scissors	P0 loaded	P0 loaded	Neutral

Table 4.1 illustrates the representation of this game in sgamesolver's tabular format. With the exception of the first row, which specifies δ_i for all players, each row represents exactly one strategy profile in a specific state. Because each of the three states allows 9 action profiles, a complete table for this game would comprise of 27 rows, plus one for the discount factors. One column contains the state label and a set of `a_{player}`-columns the actions of the respective agents. Then, a set of `u_{player}`-columns specifies the instantaneous utilities arising from this action profile. Finally, a single `to_state`-column contains the resulting next state. The case of non-deterministic transitions is discussed below. The table to represent the game can be in a variety of formats, e.g. Excel- or csv-files, or a Pandas dataframe; it can be created by hand or programmatically. Once a file is in place, a single line turns it into a game-object that can then be solved as before:

```
import sgamesolver
rps_game = sgamesolver.SGame.from_table("RPS_table.xlsx")
homotopy = sgamesolver.homotopy.LogTracing(rps_game)
homotopy.solver_setup()
homotopy.solve()
print(homotopy.equilibrium)
```

The solver then quickly computes the stationary equilibrium. The output for this game is:

```
An equilibrium was found via homotopy continuation.
++++++++++ neutral ++++++++++
                rock  paper sciss
p0 : v=0.00, σ=[0.369 0.298 0.333]
p1 : v=0.00, σ=[0.369 0.298 0.333]
+++++++++ p0_loaded +++++++++
                 rock  paper sciss
```

```
p0 : v= 0.11, σ=[0.257 0.409 0.333]
p1 : v=-0.11, σ=[0.480 0.187 0.333]
+++++++++ p1_loaded +++++++++
                 rock  paper sciss
p0 : v=-0.11, σ=[0.480 0.187 0.333]
p1 : v= 0.11, σ=[0.257 0.409 0.333]
```

Note that in *Rock, Paper, Laser Scissors*, the transition function $\mathbf{\Phi}$ has two properties which are not general to stochastic games: First, the resulting state depends on the action profile alone, but not on the current state (formally, $\phi_{s \mapsto s'}(\cdot)$ is identical for all s). Second, state transitions are deterministic: Every action profile implies a single state to follow, rather than a random distribution (formally, $\phi_{s \mapsto s'}(\cdot)$ takes values 0 and 1 only). Again, this may differ for other games. One could imagine a variant of the game where Scissors become loaded only with 50% probability after playing them. Transitions for this variant would be characterized as follows:

			Player 1	
		Rock	Paper	Scissors
Player 0	Rock	Neutral	Neutral	Neutral: 0.5 P1 loaded: 0.5
	Paper	Neutral	Neutral	Neutral: 0.5 P1 loaded: 0.5
	Scissors	Neutral: 0.5 P0 loaded: 0.5	Neutral: 0.5 P0 loaded: 0.5	Neutral: 0.5 P0 loaded: 0.25 P1 loaded: 0.25

It is straightforward to represent such stochastic transitions in the tabular format, by simply listing `state:  probability`-pairs in the `to_state`-column.[4] Table 4.2 demonstrates this. This concludes discussion of our first example – we turn to the second, a simple economic model.

Example 2: Sequential Price Competition. This model by Maskin and Tirole (1988) is a well-known economic application. Two firms produce a homogeneous good and compete by setting prices on the discrete grid $\{0, 0.1, 0.2, ..., 1.1\}$. Prices are somewhat sticky: Firm 0 gets to adjust its price only in even periods (and has to keep this price in the subsequent odd period). Conversely, Firm 1 gets to adjust its price only in odd periods. All consumers buy from the cheapest

[4] An alternative format for random transitions exists; there, the `to_state`-column is replaced by a set of `phi_{state}`-columns, one for each state, which then simply contain the probabilities of the respective state to follow.

state	a_p0	a_p1	u_p0	u_p1	to_state
delta			0.95	0.95	
neutral	rock	rock	0	0	neutral
neutral	rock	paper	-1	1	neutral
neutral	rock	scissors	1	-1	p1_loaded
neutral	paper	rock	1	-1	neutral
neutral	paper	paper	0	0	neutral
neutral	paper	scissors	-1	1	p1_loaded
neutral	scissors	rock	-1	1	p0_loaded
neutral	scissors	paper	1	-1	p0_loaded
neutral	scissors	scissors	0	0	neutral
p0_loaded	rock	rock	0	0	neutral
p0_loaded	rock	paper	-1	1	neutral
p0_loaded	rock	scissors	1	-1	p1_loaded
		...			
p1_loaded	rock	rock	0	0	neutral
p1_loaded	rock	paper	-1	1	neutral
p1_loaded	rock	scissors	1	-1	p1_loaded
		...			
p1_loaded	scissors	scissors	-1	1	neutral

Table 4.1: The game *Rock, Paper, Laser Scissors* in tabular format.

state	a_p0	a_p1	u_p0	u_p1	to_state
delta			0.95	0.95	
neutral	rock	rock	0	0	neutral
neutral	rock	paper	-1	1	neutral
neutral	rock	scissors	1	-1	neutral: 0.5, p1_loaded: 0.5
neutral	paper	rock	1	-1	neutral
neutral	paper	paper	0	0	neutral
neutral	paper	scissors	-1	1	neutral: 0.5, p1_loaded: 0.5
neutral	scissors	rock	-1	1	neutral: 0.5, p0_loaded: 0.5
neutral	scissors	paper	1	-1	neutral: 0.5, p0_loaded: 0.5
neutral	scissors	scissors	0	0	neutral: 0.5, p0_loaded: 0.25, p1_loaded: 0.25
p0_loaded	rock	rock	0	0	neutral
p0_loaded	rock	paper	-1	1	neutral
p0_loaded	rock	scissors	1	-1	neutral: 0.5, p1_loaded: 0.5
					...

Table 4.2: Variant of *Rock, Paper, Laser Scissors* with modified, non-deterministic transitions.

firm; in case of a tie, demand is split evenly. Total demand in each period is $q(p) = 2 - \min(p_0, p_1)$. For simplicity, marginal costs are assumed to be zero. Firms maximize total discounted profit, and share a discount factor $\delta = 0.95$. In this model, the monopoly price is 1. It is however important to include an additional, higher grid point, namely 1.1, because firms will sometimes use that price to invite the other to set the monopoly price in the next period.

Again, the easiest way to pass this game to sgamesolver is to create a tabular representation. Two properties characterize the state: Which firm is active, and which of the 12 possible prices was previously chosen by the other firm. This results in a total of 24 states. The passive firm has no choice to make; this is reflected by assigning a singleton action to it. The actions of the active firm correspond to the 12 possible prices, so that there are 12 action profiles per state, thus 288 in total. Table 4.3 shows the game in tabular format. The exact choice of labels for actions and states is largely up to the user; they only serve readability. With a total of 289 lines, it is certainly wise to let a computer generate the table; in Appendix 4.A, we list exemplary code to do this.

As it turns out, the game has a rather large set of equilibria. We will postpone discussion to Section 4.3.2.1, where we demonstrate how to compute multiple equilibria with sgamesolver and discuss some selective properties of the homotopy method.

4.3 Homotopy Methods

Homotopy methods are a general mathematical tool to solve high-dimensional systems of equations; Zangwill and Garcia (1981) offer an excellent introduction. Suppose one wants to find a solution to $F(x) = 0$, where $F : \mathbb{R}^n \to \mathbb{R}^n$ is highly non-linear. Most solution methods, such as Newton's, are local in nature, and will converge only if one already has a good approximation of the solution to begin with (Miranda and Fackler, 2004). The core advantage of homotopy methods is that they are globally convergent. Intuitively speaking, they approach the problem by first continuously transforming it to a similar, but much easier one. This transformation is then reversed while keeping track of the solution, until a solution of the original problem is recovered.

More formally, the system is relaxed by introducing a homotopy parameter t and a homotopy function $H(x, t)$, $H : \mathbb{R}^{n+1} \to \mathbb{R}^n$. H is a suitable function if it has the following three properties. First, $H(x, 1) = F(x)$, so that a solution at

state	a_firm0	a_firm1	u_firm0	u_firm1	to_state
delta			0.95	0.95	
firm0 active; p1=0.0	0.0	0.0 (inactive)	0.0	0.0	firm1 active; p0=0.0
firm0 active; p1=0.0	0.1	0.0 (inactive)	0.0	0.0	firm1 active; p0=0.1
firm0 active; p1=0.0	0.2	0.0 (inactive)	0.0	0.0	firm1 active; p0=0.2
firm0 active; p1=0.0	0.3	0.0 (inactive)	0.0	0.0	firm1 active; p0=0.3
firm0 active; p1=0.0	0.4	0.0 (inactive)	0.0	0.0	firm1 active; p0=0.4
firm0 active; p1=0.0	0.5	0.0 (inactive)	0.0	0.0	firm1 active; p0=0.5
firm0 active; p1=0.0	0.6	0.0 (inactive)	0.0	0.0	firm1 active; p0=0.6
firm0 active; p1=0.0	0.7	0.0 (inactive)	0.0	0.0	firm1 active; p0=0.7
firm0 active; p1=0.0	0.8	0.0 (inactive)	0.0	0.0	firm1 active; p0=0.8
firm0 active; p1=0.0	0.9	0.0 (inactive)	0.0	0.0	firm1 active; p0=0.9
firm0 active; p1=0.0	1.0	0.0 (inactive)	0.0	0.0	firm1 active; p0=1.0
firm0 active; p1=0.0	1.1	0.0 (inactive)	0.0	0.0	firm1 active; p0=1.1
firm0 active; p1=0.1	0.0	0.1 (inactive)	0.0	0.0	firm1 active; p0=0.0
firm0 active; p1=0.1	0.1	0.1 (inactive)	0.095	0.095	firm1 active; p0=0.1
firm0 active; p1=0.1	0.2	0.1 (inactive)	0.0	0.19	firm1 active; p0=0.2
firm0 active; p1=0.1	0.3	0.1 (inactive)	0.0	0.19	firm1 active; p0=0.3
		...			
firm0 active; p1=1.1	0.6	1.1 (inactive)	0.84	0.0	firm1 active; p0=0.6
firm0 active; p1=1.1	0.7	1.1 (inactive)	0.91	0.0	firm1 active; p0=0.7
firm0 active; p1=1.1	0.8	1.1 (inactive)	0.96	0.0	firm1 active; p0=0.8
firm0 active; p1=1.1	0.9	1.1 (inactive)	0.99	0.0	firm1 active; p0=0.9
firm0 active; p1=1.1	1.0	1.1 (inactive)	1.0	0.0	firm1 active; p0=1.0
firm0 active; p1=1.1	1.1	1.1 (inactive)	0.495	0.495	firm1 active; p0=1.1
firm1 active; p0=0.0	0.0 (inactive)	0.0	0.0	0.0	firm0 active; p1=0.0
firm1 active; p0=0.0	0.0 (inactive)	0.1	0.0	0.0	firm0 active; p1=0.1
firm1 active; p0=0.0	0.0 (inactive)	0.2	0.0	0.0	firm0 active; p1=0.2
firm1 active; p0=0.0	0.0 (inactive)	0.3	0.0	0.0	firm0 active; p1=0.3
firm1 active; p0=0.0	0.0 (inactive)	0.4	0.0	0.0	firm0 active; p1=0.4
firm1 active; p0=0.0	0.0 (inactive)	0.5	0.0	0.0	firm0 active; p1=0.5
		...			
firm1 active; p0=1.1	1.1 (inactive)	0.7	0.0	0.91	firm0 active; p1=0.7
firm1 active; p0=1.1	1.1 (inactive)	0.8	0.0	0.96	firm0 active; p1=0.8
firm1 active; p0=1.1	1.1 (inactive)	0.9	0.0	0.99	firm0 active; p1=0.9
firm1 active; p0=1.1	1.1 (inactive)	1.0	0.0	1.0	firm0 active; p1=1.0
firm1 active; p0=1.1	1.1 (inactive)	1.1	0.495	0.495	firm0 active; p1=1.1

Table 4.3: Excerpt from the tabular representation of the game *Sequential Price Competition.*

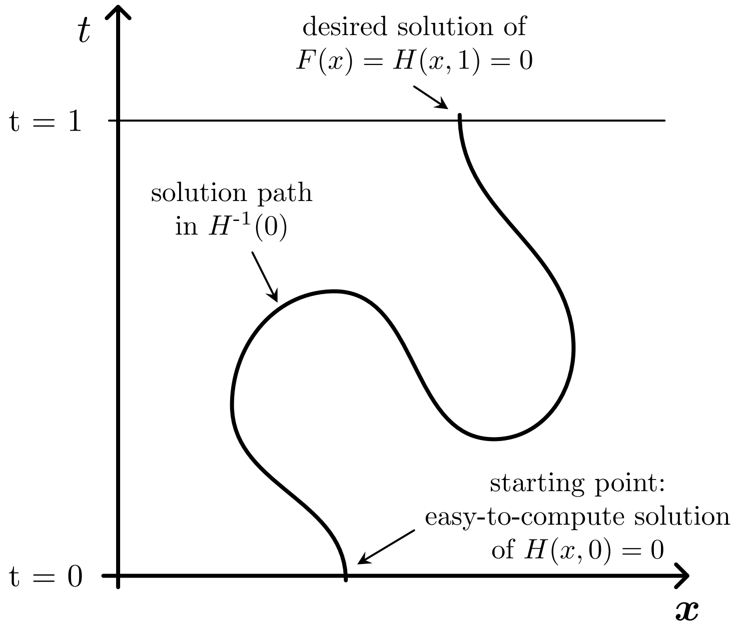


Figure 4.1: The homotopy principle: Starting at an easily computed solution of $H(x, 0) = 0$, a path contained in $H^{-1}(0)$ is followed to the desired solution of $H(x, 1) = 0$.

$t = 1$ solves the original problem. Second, a solution to $H(x, 0) = 0$ is known or easy to compute. Finally, the solution set of $H(x, t) = 0$ contains a path that can be followed from the easy solution at $t = 0$ to the desired one at $t = 1$. For this purpose, the path can be either continuously differentiable or piecewise-linear; we will focus on the differential case, which is what sgamesolver uses exclusively. Figure 4.1 illustrates the homotopy principle.

It should be noted that a homotopy method consists of two distinct components: First, a function H that defines the homotopy path; and second, an algorithm that implements the actual numerical tracking of the path. Thus, the path is a mathematical object whose existence and properties can be established with according certainty. On the other hand, the tracking will usually be done in finite precision; therefore, even if one can guarantee existence of a path, this does not exclude potential numerical difficulty in tracking it. In this section, we will cover the homotopy functions implemented in sgamesolver; the tracking algorithm will then be discussed in Section 4.4.

The standard way to ensure existence of the solution path is via three properties of the homotopy function H: Regularity, uniqueness of the solution at $t = 0$, and boundedness of $H^{-1}(0)$. First, regularity makes it sensible to speak of paths in the first place. It means that the Jacobian of H has full rank everywhere on

$H^{-1}(0)$ and thus allows application of the implicit function theorem, which ensures that $H^{-1}(0)$ is indeed a 1-dimensional manifold and consists of paths and loops only. Regularity thereby rules out potential pathologies such as higher-dimensional subsets, splitting of paths, spirals or sudden endpoints. Next, if the solution at $t = 0$ is unique, it must be that a single path crosses there, that this path is not a loop, and that it can not return to $t = 0$: this component of $H^{-1}(0)$ is the homotopy path. Other, disjoint components of $H^{-1}(0)$ may exist, but do not impede tracking the path. Finally, by virtue of boundedness, the path cannot go off to infinity in any of the dimensions of $\boldsymbol{x}$. The path is boxed in, must reach $t = 1$ eventually, and thus lead to the desired solution of the original problem.

4.3.1 Homotopy Functions for Stochastic Games

Currently, sgamesolver implements two different homotopy functions that allow the computation of stationary equilibria in stochastic games: Logit QRE and logarithmic tracing. Both receive a detailed exposition in the respective Chapters 2 and 3. Their treatment here will therefore be rather cursory, and focus on a few features relevant for their usage with sgamesolver. Others have introduced further homotopy functions for stochastic games, including the linear tracing procedure by Herings and Peeters (2003, 2004), on which the logarithmic version is directly based, the interior point method by Dang et al. (2022), and the global Newton method by Govindan and Wilson (2009). While these are not currently implemented in sgamesolver, the package is structured to make such extensions straightforward and easy. Borkovsky et al. (2010) and Herings and Peeters (2010) offer more general discussions of homotopy methods for (stochastic) games.

Implementation of the homotopy functions in sgamesolver primarily consists in routines to evaluate $H(\boldsymbol{\sigma}, \boldsymbol{V}, t)$ and its Jacobian $J(\boldsymbol{\sigma}, \boldsymbol{V}, t)$, where $\boldsymbol{\sigma}$ are stationary strategy profiles, $\boldsymbol{V}$ state-player-values, and t the homotopy parameter. Because the speed of these evaluations is critical, they are implemented in cython, a language that is easy to interface from python, but compiles to C, resulting in good performance.[5] In addition, the implementations also provide functionality to compute the starting point, to set parameters, and similar tasks.

[5] As a fallback option, sgamesolver also has implementations based only on NumPy, which are however noticeably slower.

4.3.2 The Logarithmic Stochastic Tracing Procedure

The Logarithmic Stochastic Tracing Procedure (short: LogTracing, introduced in Chapter 3) is an extension of Herings and Peeters (2004), who in turn adapt the linear tracing procedure of Harsanyi and Selten (1988) to stochastic games. The core idea of the tracing methods is to introduce a prior $\boldsymbol{\rho}$, representing a first belief by the players on others' likely strategies. A family of auxiliary games $\mathcal{G}^t$, with $t \in [0,1]$, is then constructed as follows: For $t = 0$, players play only against their priors. For $t \in (0,1)$, each player maximizes against a belief that is a convex mixture of their priors and the responses of all other players. At $t = 1$ finally, players play only against others' responses, so that the original game is restored. Formally, the auxiliary games simply use convex combinations for u and ϕ:

$$\bar{u}^t_{si}(\boldsymbol{\sigma}_s) := t u_{si}(\boldsymbol{\sigma}_{si}, \boldsymbol{\sigma}_{s,\text{-}i}) + (1-t) u_{si}(\boldsymbol{\sigma}_{si}, \boldsymbol{\rho}_{s,\text{-}i})$$

$$\bar{\phi}^t_{s \to s'}(\boldsymbol{\sigma}_s) := t \phi_{s \to s'}(\boldsymbol{\sigma}_{si}, \boldsymbol{\sigma}_{s,\text{-}i}) + (1-t) \phi_{s \to s'}(\boldsymbol{\sigma}_{si}, \boldsymbol{\rho}_{s,\text{-}i})$$

LogTracing further adds a logarithmic penalty term to instantaneous utilities:

$$\hat{u}^t_{si}(\boldsymbol{\sigma}_s) := t u_{si}(\boldsymbol{\sigma}_{si}, \boldsymbol{\sigma}_{s,\text{-}i}) + (1-t) u_{si}(\boldsymbol{\sigma}_{si}, \boldsymbol{\rho}_{s,\text{-}i}) + (1-t)\eta \sum_{a \in A_{si}} \nu_{sia} \log(\sigma_{sia})$$

Essentially, the logarithmic terms have a regularizing function: They ensure that the path is well-defined for all games, and make it smooth and interior, thereby improving computational performance. Note that the penalties are weighted by the parameter vector $\boldsymbol{\nu} = (\nu_{sia})$, whose main role is to ensure specific formal properties. For computational purposes, leaving the weights at the default of 1 generally works well in our experience.

Because no strategic interaction is present at $t = 0$, $\mathcal{G}^0$ corresponds to a set of independent Markov decision processes, which are easily solved. Starting from their solution, a path of equilibria of the auxiliary games is followed until reaching an equilibrium of the original game at $t = 1$. The set of these equilibria can be characterized as the zero set of the homotopy function H,

$$\begin{aligned} H_{sia}(\boldsymbol{\sigma}, \boldsymbol{V}, t) \;:=\; & \overbrace{\sigma_{sia}\Big(\bar{U}^t_{si}(a, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}_i) - V_{si}\Big)}^{\text{replicator dynamics}} \\ & + (1-t)\eta \underbrace{\Big(\nu_{sia} + \sigma_{sia} \sum_{a' \in A_{si}} \nu_{sia'} \Big[\log(\sigma_{sia'}) - 1\Big]\Big)}_{\text{logarithmic perturbation}} \end{aligned} \tag{4.1}$$

for each (s, i) and $a \in A_{si}$. In addition, there is a standard sum-to-one condition for the mixed strategies σ_{sia} not shown here. As indicated in equation (4.1), H can be understood to combine a form of replicator dynamics with a logarithmic perturbation, which is then faded out as $t \to 1$. The zero set of H, which includes the homotopy path, consists of the stationary points of this dynamic.

4.3.2.1 Dependence of the Path on the Prior

One of the inputs for the logarithmic tracing procedure is the prior $\boldsymbol{\rho}$, a strategy profile that can be chosen freely. Because the path varies with this choice, different priors may lead to different equilibria. Harsanyi and Selten (1988) base their theory of equilibrium selection for normal form games on this property of the tracing procedure. They interpret the traversal along the homotopy path as a form of Bayesian strategic reasoning, in which all players gradually transform priors into equilibrium beliefs. Much of their theory then concerns the choice of priors for that interpretation. An analogous, axiomatic theory of selection for stochastic games is likely out of reach.[6] However, one can of course use the connection to a particularly salient prior (either specific to the game, or e.g. the centroid strategy) or the size the basins of attraction as part of a selection criterion.

From a more practical perspective, that the computed equilibrium depends on the prior allows to uncover potentially multiple equilibria by searching the prior space, either systematically or using random draws.

Example 2 (cont.): Searching the prior space. Earlier, we introduced the game *Dynamic Price Competition*; with its high number of equilibria, it is an excellent example to illustrate the idea of searching the prior space and discuss

[6] The reason for that is simple: Starting with size 3 × 3, even normal-form games defy attempts at useful categorization of games and equilibria. For example, the concept of risk-dominance prominent in Harsanyi and Selten's work does not generalize beyond 2 × 2. Clearly, adding states and transitions only potentiates that problem.

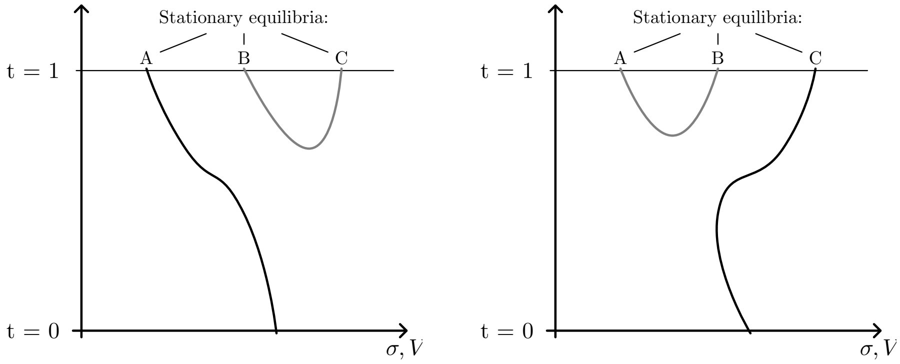


Figure 4.2: Variation of the homotopy path with different priors.

the selective properties of LogTracing. Because this is somewhat tangential, it is relegated to Appendix 4.A.2, where we exemplify in code how to perform the search over priors, comment on the equilibria we find, and show that the equilibrium with the largest basin of attraction in this game does seem to have properties resembling risk-dominance, in line with the formal results of Harsanyi and Selten (1988) for one-shot games.

It is natural to ask whether all equilibria of a given game can be found this way, provided the search is sufficiently dense. The answer depends on the specific game; we will first consider the generic case, and then briefly discuss exceptions. Generically, stochastic games have an odd number of isolated equilibria (Herings and Peeters, 2004). This situation is illustrated in Figure 4.2. For any chosen prior, one equilibrium will be connected to the starting point by the homotopy path, while the others will be connected pairwise by additional paths contained in $H^{-1}(0)$. Generically, half of all equilibria, rounded up are reachable via the homotopy (in Figure 4.2: A and C). For any prior, those equilibria not connected to the starting point are always connected in pairs of one reachable and one unreachable equilibrium. This could be shown formally using degree theory and the fact that 0 is a regular value of H (Herings and Peeters, 2004; Zangwill and Garcia, 1981, ch. 3; Chapter 3). For reasons of brevity, we will not spell this out here. As discussed before, it is possible to interpret the homotopy function H as a game dynamic (equation 4.1) – in which case it resembles replicator dynamics with an additional logarithmic perturbation, and the solutions to $H = 0$ are its stationary points. Generically, the equilibria in the reachable set (A and C in the example) will be stable under this dynamic, and the others (B) unstable, so that the method will tend to select from the subset of equilibria that is more plausible to begin with.

With a small trick, one can still use the homotopy method to find equilibria that can not be reached directly. Suppose that one has found equilibrium A using prior $\boldsymbol{\rho}_A$, and equilibrium C with prior $\boldsymbol{\rho}_C$. Then set the prior to $\boldsymbol{\rho}_A$ (as in the left panel of Figure 4.2), but now follow the additional path starting at equilibrium C, rather than the usual starting point at $t = 0$. This will lead to equilibrium B. The following example demonstrates how to do this in sgamesolver.

Example 3: Finding additional equilibria. To illustrate all the preceding discussion, consider well-known one-shot game *Stag Hunt*, with the following payoff matrix:

	stag	hare
stag	$10, 10$	$1, 8$
hare	$8, 1$	$5, 5$

The game has three equilibria: $(stag, stag)$, $(hare, hare)$, and a mixed equilibrium where both play $({}^2/_3, {}^1/_3)$. Only the former two have a basin of attraction with positive measure and can be reached by varying the prior. The mixed equilibrium is unstable, and one could arrive at it only if priors $\boldsymbol{\rho}$ and weights $\boldsymbol{\nu}$ are chosen such that the starting point already exactly coincides with it; if the choice is randomized, this is a probability zero event. To still compute it using the homotopy method, one can use a prior known to lead to $(hare, hare)$, but start to follow the path at $(stag, stag)$ rather than the starting point, or vice versa. Appendix 4.B contains code that demonstrates how to do this in sgamesolver.

Returning to the discussion on the effect of a varying prior, we will briefly mention what can happen in non-generic edge cases. First, the equilibrium set of a game may not be finite and discrete, but contain higher-dimensional subsets (trivial examples can be constructed by duplicating equilibrium strategies). In that case, finite sampling can of course never reach all equilibria; but it is of course possible to compute some points in these subsets and then proceed from there. Second, non-generic games may have a finite, potentially even number of equilibria, some of which are not reachable at all. The most simple example of this is the game

	L	R
T	$0,0$	$0,0$
B	$0,0$	$1,1$

which has only two equilibria: (T, L) and (B, R). Here, the homotopy path will always lead to the stable equilibrium (B, R). The degenerate equilibrium (T, L) is always isolated in $H^{-1}(0)$, so that it can not be computed via the homotopy, not even starting at another equilibrium as discussed above. Because such equilibria are always highly unstable and thus rather implausible outcomes of the game, this is arguably not a serious limitation.

To finish discussion of the role of the prior, we should mention that varying the penalty weights $\boldsymbol{\nu}$ similarly affects the path and thus the selected equilibrium. It seems to us that there is no good reason to vary both at the same time, which also is much harder to interpret. We therefore suggest to generally use the default $\boldsymbol{\nu} = \mathbf{1}$ and vary the prior $\boldsymbol{\rho}$ as desired. Alternatively, set $\boldsymbol{\rho}$ to zero and vary $\boldsymbol{\nu}$ – this removes all utility components stemming from the prior, so that the auxiliary games are convex combinations of the original game and the log-penalties only. Harsanyi (1973) suggested a similar transformation for normal form games.

4.3.3 Logit Markov QRE

This homotopy is an extension of quantal response equilibrium (QRE) to stochastic games, a prominent solution concept in behavioral game theory (McKelvey and Palfrey, 1995; Goeree et al., 2016). A detailed exposition is found in Chapter 2. The core idea of QRE is that players do not play perfect best responses, but rather noisy version: They make mistakes, but the more costly a mistake, the less likely it is. Thus, each action is chosen with a probability that is increasing in its utility. Many mappings from utilities to probabilities are possible; sgamesolver uses logit response, where choice probabilities are proportional to exponentiated utilities:

$$\sigma_{sia} = \frac{\exp\left(\lambda U(a, \boldsymbol{\sigma}_{\text{-}i}, \boldsymbol{V})\right)}{\sum_{a' \in A_{si}} \exp\left(\lambda U(a', \boldsymbol{\sigma}_{\text{-}i}, \boldsymbol{V})\right)}$$

Stationary logit QRE are then the fixed points of this function. Note that here, λ plays the role of the homotopy parameter t and, somewhat atypically, takes values in $[0, \infty)$. It measures the precision with which agents respond to utility differences. At $\lambda = 0$, responses are completely driven by noise: The fraction simplifies to $\frac{1}{|A_{si}|}$, so that the starting point is trivial to determine. As λ increases, responses get closer and closer to best replies. As shown in Chapter 2, for $\lambda \rightarrow \infty$, the limiting points of the stationary QRE correspondence are always actual stationary equilibria. Thus, while the method uses an approximate solution concept to define the homotopy path, it does compute an exact equilibrium (save for the numerical limitations of any finite precision method).

Example 4: Computing a sequence of QREs. Nevertheless, sgamesolver can of course be used to compute QRE for finite values of λ. For example, one might want to fit QRE to some empirical data or graph the QRE correspondence. sgamesolver can aide this by computing QRE strategies for a fine grid of λ. The following code snippet returns to the example game *Rock, Paper, Laser Scissors* and computes stationary QRE for $\lambda = 0.01, 0.02, ..., 10$.

```
import sgamesolver, numpy as np
rps_game = sgamesolver.SGame.from_table("RPS_table.xlsx")
homotopy = sgamesolver.homotopy.QRE(rps_game)
homotopy.solver_setup()

strategies = np.zeros((1000, 3, 2, 3))
for n in range(1000):
    homotopy.solver.set_parameters(t_target=(n+1)/100)
    homotopy.solve()
    strategies[n] = homotopy.equilibrium.strategies
```

4.4 The Path Tracking Algorithm

In this section, we discuss the algorithm that is responsible for numerically tracking the path defined by $H = 0$ to the desired solution at $t = 1$. sgamesolver includes a solver module that, ideally, does this without further user interaction, so that detailed knowledge of its workings is not necessary. However, a basic understanding is helpful if one wants to tune the parameters used by the solver, or if the solver encounters unexpected problems. For example, rarely but depending on the specific game, the solver may encounter numerically unstable regions; these can generally be navigated through by slowing the solver down with parameter adjustments. In the following, we therefore sketch how predictor-corrector methods work in general, the concrete implementation in sgamesolver, and the parameters that govern its behavior.

In homotopy functions for stochastic games, the vector of unknowns generally consists of a strategy profile $\boldsymbol{\sigma} = (\sigma_{sia})$ and state-player-values $\boldsymbol{V} = (V_{si})$.[7] We will sometimes use the shorthands $\boldsymbol{y} := (\boldsymbol{x}, t) := (\boldsymbol{\sigma}, \boldsymbol{V}, t)$. sgamesolver uses differentiable homotopies, where the path in H is smooth (the alternative are piece-wise linear homotopies, which are not covered here). A naïve approach to follow the path would be to start at $t_0 = 0$, then repeatedly increase t_i slightly and compute a solution to $H(\boldsymbol{x}, t_i) = 0$, until the desired $t = 1$ is reached. However, this is usually not feasible, because the path is not always monotonic in t: As Figure 4.1 illustrates, one may have to decrease t while following some sections of the path. The commonly used tool are therefore predictor-corrector procedures, which alternate between predictor and corrector steps until the target is reached.[8] Figure 4.3 illustrates this principle. For the predictor step, the direction of the curve is extrapolated from the current point $\boldsymbol{y}_k$, yielding a point $\boldsymbol{y}_k^0$ further down the curve, but only approximately on it. sgamesolver uses a simple Euler step as predictor, i.e. just follows the tangent direction at the current point of the curve. Starting from the predictor point $\boldsymbol{y}_k^0$, a sequence of corrector iterations y_k^i is computed using Newton's method, which brings the current point back onto the curve; this step of course makes use of the fact that the curve is defined by $H = 0$. Once that has been accomplished with the desired precision, the next predictor point $\boldsymbol{y}_{k+1}^0$ is computed, and so on.

Importantly, the presence of the corrector steps means that there is no accumulation of numerical error along the path, in contrast to e.g. the numerical integration of ODEs. Because of this, the precision during tracking only needs to be sufficient to not lose the path completely; once one has reached a solution at $t = 1$, one can then refine it to the desired accuracy.[9]

An essential aspect if the procedure is choosing the step length ds, which determines how far along the tangent the next predictor point is set (see Figure 4.3). The choice involves a clear trade-off: The smaller ds, the higher the number of steps needed to traverse the path. On the other hand, the larger ds relative to the path's curvature, the larger the prediction error and thus the number of corrector

[7] To be precise, our implementations operate on $\log(\sigma_{sia})$ in place of σ_{sia} for numerical reasons, as suggested by Turocy (2005, 2010). However, this should be irrelevant to users.

[8] Allgower and Georg (1990) offer a great introduction to the computational aspects of homotopy methods.

[9] In principle, the homotopy path can be characterized as the trajectory of a system of ODEs, which would allow to use the more complex, higher-order methods common in solving those (e.g. Runge-Kutta). However, since the corrector steps prevent the accumulation of error, this is not necessary – the simple, but computationally inexpensive Euler variant is faster and ultimately no less accurate in this context.

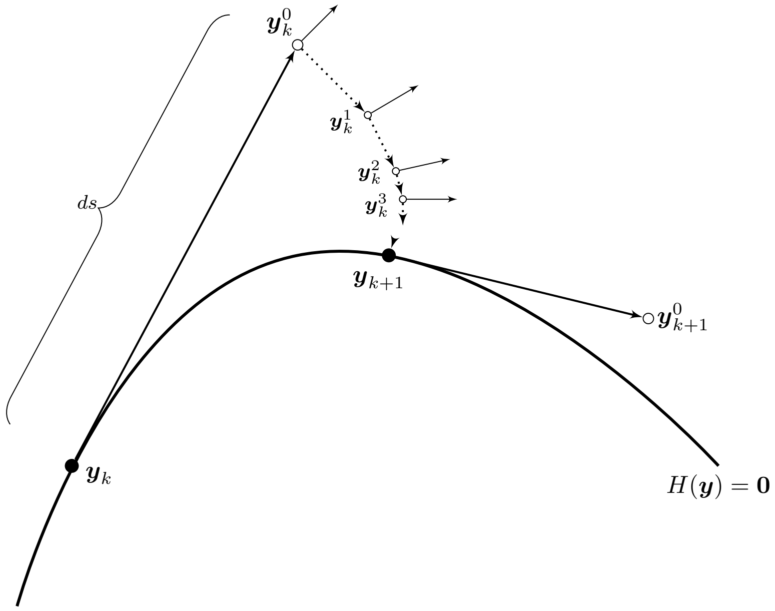


Figure 4.3: The predictor-corrector principle. Starting at $\boldsymbol{y}_k$, a predictor point $\boldsymbol{y}_k^0$ is computed; a sequence of corrector steps $\boldsymbol{y}_0^1, \boldsymbol{y}_0^2, ...$ then (most of) the prediction error, so that $\boldsymbol{y}_{k+1}$ again lies on the curve. This is repeated until reaching the target of $t = 1$.

iterations needed for each step. If the error is too large, the predictor point may even land outside the region of convergence of the corrector, which will then fail completely – in which case one has to repeat the predictor step with a smaller step size. Therefore, ds should ideally adapt to the local curvature of the path and the local speed and stability of Newton's method. To this end, sgamesolver uses an adaptive procedure to control ds: It is increased following sequences of successful steps with sufficiently fast convergence, and decreased whenever a corrector failure occurs. Section 4.4.1 discusses this in detail, and describes the parameters that govern this adaptation.

A potential problem to be aware of is *path jumping*, illustrated in Figure 4.4. It occurs when the corrector converges, but to a point that does not lie ahead on the curve as desired, but either on a completely different component of $H^{-1}(0)$ or even behind the previous point on the same curve. In the former case, path tracing will generally resume normally on the new component and will still eventually reach an equilibrium, albeit not the one connected to the starting point. In the latter case, it is possible that the solver passes the critical region safely when reaching it again. But path tracking may also get stuck in a loop if path-jumping re-occurs when passing the problematic point again. Similarly, repeated occurrences of jumping back and forth between components in a badly conditioned

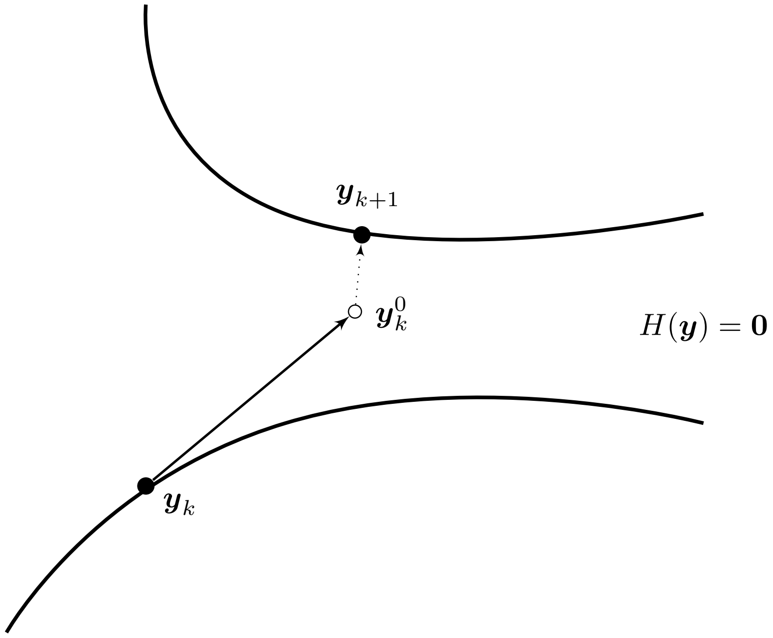


Figure 4.4: The issue of path jumping, which occurs if the predictor point lands in the region of convergence of another path segment.

region could also lead to a loop. Unfortunately, both path jumping in general and looping in particular are hard to detect heuristically, especially without incurring considerable computational overhead; luckily, it seems to be a rather rare occurrence. Based on our experience, we therefore recommend to oversee the path tracking process, and if problems are encountered, to limit ds and place tighter restrictions on the acceptance of corrector steps until the critical regions have been passed. The next section should help in this regard, by giving details on the exact implementation of the predictor-corrector procedure in sgamesolver. Further instructions for trouble-shooting are found in the online documentation.

The tangent used for the predictors is by itself of course ambiguous, as it determines direction only up to the sign. This is handled by using the orientation of the path (Zangwill and Garcia, 1981, ch. 2.2). Because the orientation may change when crossing a bifurcation point, it is necessary to detect such events and change the sign of the tangent accordingly. This is illustrated in Figure 4.5. Provided that the sign is changed as needed (which sgamesolver does automatically), such bifurcations do not impede path tracking, which can continue right across them. In principle, it is even possible to trace all paths emanating from such a bifurcation point to uncover additional equilibria (Allgower and Georg, 1990, ch. 8).

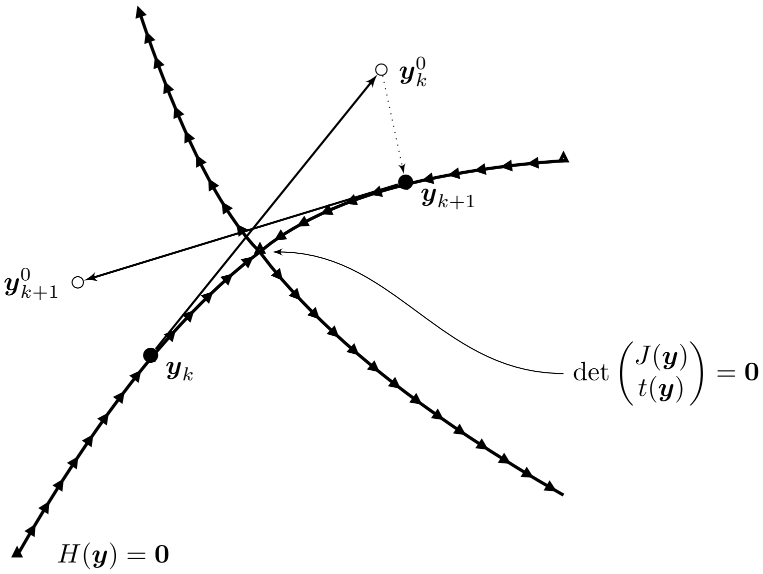


Figure 4.5: A pitchfork bifurcation with a reversal of orientation.

4.4.1 Implementation in sgamesolver

The implementation of the predictor-corrector-method in sgamesolver is presented in Algorithm 1 in pseudo-code. Of course, this bird's eye view omits many aspects, e.g. the actual computation of the tangent (using a QR-decomposition), caching of function evaluations and other performance-related details, some additional options described in the text below, or functionality allowing to store the path and return to previous points manually. All variables listed in `monospace` are parameters of the solver. sgamesolver comes with a set of defaults, adapted to the specific homotopy functions.[10] Each of these parameters can be changed by the user, either before starting the solver, or by pausing and then resuming computation at any time. The following shows a minimal example; please refer to the online manual for details.

```
homotopy = sgamesolver.homotopy.QRE(game)
homotopy.solver_setup()
homotopy.solver.set_parameters(ds_inflation_factor=1.5,
    ds_min=1e-8,
    max_steps=100)
homotopy.solve()
homotopy.solver.set_parameters(corrector_tol=0.0001,
```

[10] For our tests and tuning of parameters, we mainly used games where u ranges from 0 to 1, and with δ around 0.95. Because the solver also operates on $\boldsymbol{V}$-space, which scales in u and δ, parameter adjustments might be helpful if a game differs greatly. An alternative is to re-scale u to obtain similar ranges for $\boldsymbol{V}$.

```
    max_steps=200)
homotopy.solve()
```

As Algorithm 1 reveals, the solver mainly operates through two nested loops: An outer predictor loop [lines 5–26], where each iteration performs an Euler update [6] and then enters the inner corrector loop [8–17], which runs corrector updates until a zero of H is reached with sufficiently small error.

The actual corrector updates [9] use the Moore-Penrose pseudo-inverse of the Jacobian, J^+. By default, sgamesolver uses Quasi-Newton updates, meaning that J^+ is computed once at the predictor point, and then reused for each corrector iteration, rather than re-computing it each time. While this decreases the rate of convergence and thus necessitates a higher number of corrector iterations, avoiding additional evaluations and inversions of J usually more than makes up for it (Allgower and Georg, 1990). However, it is possible to switch to full Newton updates by setting parameter `quasi_newton` to false; line 9 then becomes

$$y_{corr} \leftarrow y_{corr} - J^+(y_{corr})H(y_{corr}).$$

There are three failure criteria for the corrector loop [11]. The first limits the combined length of all corrector updates; failure will occur if y_{pred} is too far off the path. The second places an upper bound on the ratio the current and previous corrector update distances, thereby enforcing a specific rate of convergence. Finally, the total number of corrector iterations is limited. If the corrector loop fails, ds is decreased and the predictor step repeated [12–14]. Setting strict thresholds for the criteria will increase stability, mainly by preventing path jumping, but also reduce the speed, because more steps will be discarded and ds will be lower on average.

If the corrector loops exits successfully, step size ds is increased if the following additional criteria are met [18–20]. First, the number of corrector steps must not exceed a set threshold, which may be lower than the one for corrector success (for example, one might allow up to 10 corrector iterations, but increase ds only if convergence occurred in 5 or less). Second, one can allow increases of ds only after a certain number of consecutive successful steps (as recommended by Bates et al., 2008). Both conditions help to slow down in regions where the path is winded or the Jacobian badly conditioned.

Before proceeding to the next predictor iteration, the algorithm checks heuristically whether a bifurcation was crossed and a sign swap is necessary [21–23], as

Algorithm 1 The predictor-corrector-procedure in sgamesolver

```
 1: $ds \leftarrow$ ds_initial                                          ▷ Initial step length
 2: $y \leftarrow y_0$                                                   ▷ Starting point
 3: $sign \leftarrow$ SETSIGN()                   ▷ Set initial direction towards t_target
 4: $consecutive \leftarrow 0$                  ▷ Counts consecutive successful steps
 5: repeat for $step = 1, 2, ...$                                        ▷ Predictor loop
 6:     $y_{pred} \leftarrow y + sign \cdot ds \cdot$ TANGENT($y$)         ▷ Euler predictor
 7:     $y_{corr} \leftarrow y_{pred}$
 8:     repeat for $iteration = 1, 2, ...$                               ▷ Corrector loop
 9:         $y_{corr} \leftarrow y_{corr} - J^{+}(y_{pred})H(y_{corr})$  ▷ Quasi-Newton update
10:         $dist \leftarrow ||J^{+}(y_{pred})H(y_{corr})||_2$         ▷ Distance of update
11:         if $||y_{corr} - y_{pred}||_2 \geq$ c_dist_max or  ▷ Failure criteria: total distance,
                $\frac{dist}{dist_{old}} \geq$ c_ratio_max or  ▷ ratio (skipped on $1^{st}$ iteration),
                $iteration >$ c_max_iter then                      ▷ number of iterations
12:             $ds \leftarrow ds \cdot$ ds_deflation_factor                ▷ Reduce $ds$ ...
13:             $consecutive \leftarrow 0$
14:             go to line 6                                  ▷ ... and repeat predictor
15:         end if
16:         $dist_{old} \leftarrow dist$
17:     until $||H(y_{corr})||_{max} <$ corrector_tol
18:     if $iteration \leq$ ds_inflation_max_corrector_steps and
            $consecutive \geq$ ds_inflation_min_consecutive_successes then
19:         $ds \leftarrow$ MAX($ds \cdot$ ds_inflation_factor, ds_max)        ▷ Increase $ds$
20:     end if
21:     if ANGLE(TANGENT($y$), TANGENT($y_{corr}$)) $>$ bifurc_angle_min then
22:         $sign \leftarrow -sign$                   ▷ Bifurcation detected: swap sign
23:     end if
24:     $y \leftarrow y_{corr}$
25:     $consecutive \leftarrow consecutive + 1$
26: until $|t -$ t_target$| <$ convergence_tol or                             ▷ Success
        $ds <$ ds_min or $step \geq$ max_steps                              ▷ Failure
```

illustrated in Figure 4.5. If the angle between the tangents at the previous and at the new point is too close to 180°, orientation is reversed before the next step.

Choi et al. (1996) suggest an additional heuristic to detect path jumping, not included in Algorithm 1. It monitors the change in the determinant of the augmented Jacobian (J with the tangent added as additional row) between the previous and the new point. If the ratio exceeds a threshold, this is interpreted as a sign of potential path jumping, the step discarded, and repeated with a reduced step size. In our experience, the criterion leads to a very high number of false positives when used with the homotopies implemented in sgamesolver, thereby increasing computation times by an order of magnitude or more. Thus, we suggest to consider this an experimental feature, and if at all use it in regions where one has already observed that path jumping seems to be a problem. The test can be activated by setting parameter `test_segment_jumping` to true and `det_ratio` to the desired ratio threshold.

The algorithm terminates successfully if t reaches `t_target` (e.g. $t = 1$) with the desired precision [26]. The refined end-game step size control, which prevents overshooting, is not featured in Algorithm 1. It is possible to define other convergence criteria. For example, the actual stationary equilibria in the QRE homotopy are the limit points of $\boldsymbol{\sigma}, \boldsymbol{V}$ as $t \to \infty$. To account for this, the algorithm by default runs until $\boldsymbol{\sigma}$ has converged with the desired precision. Of course, one can also just set a finite value for `t_target` to compute the QRE that corresponds to this t.

Failure occurs when at some point, the predictor-corrector-step keeps failing, although step size ds is already at its defined minimum `ds_min`. This should rarely occur (in our experience, if the algorithm fails, it manifests more often in the form of looping, see the discussion above) and is often a sign of a problem in the game's definition, e.g. ill-defined transition probabilities. Another failure criterion is a maximum number of predictor-corrector-steps; however, this never means hard failure, because it is always possible to simply increase `max_steps` and continue where the solver halted. The main use case for this criterion is to prevent endless running if the solver is stuck in a loop, which is hard to detect programmatically. It can also be used to pause and restart computations in a pre-defined frequency, e.g. to save progress regular intervals.

Note that the solver also has functionality to store an array of all visited points for diagnostic reasons, for plotting, and also to return to previous points if problems are encountered. Similarly, it is possible to save the current state of the solver (mainly $\boldsymbol{y}$, and some additional state variables like *ds* and *consecutive*) to disk and load it later, for example to continue after a reboot or to be able to return to previous points. The online documentation contains details.

4.5 Symmetries in Stochastic Games

In this section, we discuss the notion of symmetry in stochastic games, which is an important selection criterion, but also a property that can be used to speed up computations. While other treatments exist (e.g. Zinkevich, 2006), we found them hard to operationalize for our computational purposes. We begin with a basic definition.

Definition: Symmetry structure. A symmetry structure of a stochastic game $\mathcal{G}$ is an equivalence relation $\sim$ that partitions the set of agents, $S \times I$, into classes $T_0, T_1, ...$, called types. Informally speaking, two agents of the same type face the same situation provided all agents of each other type act the same; and because all agents of a type face the same situation, acting the same is then actually justified. The following formalizes this notion.

1. The type compositions (with multiplicity) of any two states either coincide or are disjoint.

2. Agents of the same type have the same number of actions: If $(s, i) \sim (s', i')$, then $|A_{si}| = |A_{s'i'}|$. Actions are assumed to be ordered such that the first action of one corresponds to the first action of the other, and so on.[11] "Acting the same" then simply means $\boldsymbol{\sigma}_{si} = \boldsymbol{\sigma}_{s'i'}$.

3. We call a vector of state-player-values $\boldsymbol{V} = (V_{si})$ or a strategy profile $\boldsymbol{\sigma}$ symmetric under $\sim$ if $(s, i) \sim (s', i')$ implies $V_{si} = V_{s'i'}$ or $\boldsymbol{\sigma}_{si} = \boldsymbol{\sigma}_{s'i'}$, respectively.

4. Note that equivalence can hold between two agents of the same player in different states, between agents of different players in different states, and between different players in the same state.

[11] Rather than presupposing this ordering, one could make the existence of according permutations part of the definition of a symmetry structure. We choose the former route, as there is no technical difference, and it eases notation.

5. The former two cases, $(s,i) \sim (s',i')$ with $s \neq s'$, are formalized as follows. Let $\boldsymbol{V} = (V_{si})$ be symmetric under $\sim$ and $\boldsymbol{\sigma}$ symmetric under $\sim$, then $(s,i) \sim (s',i')$ requires

$$U_{si}(\boldsymbol{\sigma}_{si}, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}) = U_{s'i'}(\boldsymbol{\sigma}_{s'i'}, \boldsymbol{\sigma}_{s',\text{-}i'}, \boldsymbol{V})$$

   for all $\boldsymbol{\sigma}_{si} = \boldsymbol{\sigma}_{s'i'}$.

6. The remaining case from 4. is $(s,i) \sim (s,i')$. We will first treat the case of exactly two such agents in a state, and then generalize to more. Let $\boldsymbol{V}$ be symmetric under $\sim$, and let $\boldsymbol{\sigma}_{s,\text{-}i}$ be an arbitrary strategy profile for all other agents in s that is symmetric under $\sim$. Then symmetry requires

$$U_{si}(\boldsymbol{\sigma}_{si}, \boldsymbol{\sigma}_{s,i'}, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}) = U_{si'}(\boldsymbol{\sigma}_{si'}, \boldsymbol{\sigma}_{s,i}, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}),$$

   where the first argument of U indicates the strategy of i, the second that of i'. Now suppose more than two agents from the same state s are symmetric, say $i_0, i_1, ..., i_N$. Denote by $\pi(\cdot)$ a permutation of $(i_0, i_1, ..., i_N)$. Symmetry then requires

$$U_{si_0}(\boldsymbol{\sigma}_{si_0}, \boldsymbol{\sigma}_{s,i_1}, ..., \boldsymbol{\sigma}_{s,i_N}, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V}) = \\ U_{s\pi(i_0)}(\boldsymbol{\sigma}_{s\pi(i_0)}, \boldsymbol{\sigma}_{s,\pi(i_1)}, ..., \boldsymbol{\sigma}_{s,\pi(i_N)}, \boldsymbol{\sigma}_{s,\text{-}i}, \boldsymbol{V})$$

   for all permutations and all pairs of agents. This notion of symmetry within states corresponds to the standard notion of symmetry in normal form games, after factoring in $\boldsymbol{V}$.

To illustrate this concept, reconsider the example game *Rock, Paper, Laser Scissors*. Here, a symmetry structure is given by $(Neutral, P_0) \sim (Neutral, P_1)$, $(P_0Loaded, P_0) \sim (P_1Loaded, P_1)$, and $(P_0Loaded, P_1) \sim (P_1Loaded, P_0)$. Note that this example nicely illustrates that to identify symmetries, one has to consider the complete partition, rather than just individual pairs of agents. This is because whether two agents are symmetrical often hinges on whether specific *other* pairs of agents are also symmetrical. Consider e.g. the two agents with loaded scissors. They are symmetrical, but only because their “un-loaded” direct opponents are also symmetrical – if the latter were not to act the same, the incentives for the loaded agents would not be identical either. Likewise, symmetry between the $Neutral$ agents is required for the loaded agents to be symmetric, because the expected utilities from the neutral state enter as continuation values. The other example game *Dynamic Price Competition* also has an obvious symmetry structure. Specifically, Firm 0 (Firm 1) after Firm 1 has just set $p_1 = p$ is symmetric to Firm 1 (Firm 0) after Firm 0 has just set $p_0 = p$.

Note that the symmetry structure of a game is generally not unique. As a simple example, consider a game with a single state and three players and suppose full symmetry between them. Then, all partitions into a pair and a singleton must also be symmetry structures (because if 6. holds for all permutations of the three agents, it must also hold when exchanging just two). Moreover, it is easily seen that the trivial partition of agents into singleton sets always fulfills the conditions for any game.[12]

The preceding example illustrates that between symmetry structures that are refinements of each other, it is clearly the coarsest one that is most restrictive and thus most interesting. This raises the question whether a game always has a coarsest symmetry structure. This is at least not immediately obvious, because the ordering imposed by refinement is only partial. However, in the following we will show that the answer is affirmative.

Proposition: Maximal symmetry structure. *Every game $\mathcal{G}$ has a maximally coarse symmetry structure (so that every other symmetry structure is a refinement of it).*

To prove this, we will first introduce a bit of notation. Given two equivalence relations $\sim_a$ and $\sim_b$, denote by $\sim_{a\cup b}$ the equivalence relation defined as follows:

$$x \sim_{a\cup b} y \;\Leftrightarrow\; x \sim_a y \,\vee\, x \sim_b y \,\vee\, \exists z : (x \sim_a z \wedge z \sim_b y \,\vee\, x \sim_b z \wedge z \sim_a y)$$

In words, two elements are equivalent under $\sim_{a\cup b}$ if they are either directly equivalent under either $\sim_a$ or $\sim_b$, or indirectly linked by a third element that is equivalent to both. In effect, $\sim_{a\cup b}$ is the smallest possible equivalence relation that retains equivalence under either $\sim_a$ or $\sim_b$. Next, we will show that this preserves the properties that define a symmetry structure.

Lemma: Combining symmetry structures. *If $\sim_a$ and $\sim_b$ are symmetry structures of a game, then $\sim_{a\cup b}$ is also a symmetry structure.*

[12] While the example may suggest that refinements of symmetry structures are again symmetry structures, this is not generally true. An example is again Rock, Paper, Laser Scissors, where the only symmetry structure besides the one stated above is the trivial one of singletons.

To show this, we just need to establish that for all pairs of agents with $x \sim_{a \cup b} y$, properties 5. and 6. also hold with respect to $\sim_{a \cup b}$. First consider the cases where $x \sim_a y$ or $x \sim_b y$, so that 5. and 6. hold under at least one of these. Next, observe that the sets of $\boldsymbol{V}$ and $\boldsymbol{\sigma}$ that are symmetric under $x \sim_{a \cup b} y$ are a subset of those symmetric under $x \sim_a y$ and a subset of those under $x \sim_b y$. If 5. and 6. hold when quantifying over either superset, they must also hold when quantifying over the smaller set, thus hold with respect to $x \sim_{a \cup b} y$. This covers agents for which either $x \sim_a y$ or $x \sim_b y$. We now turn to those only indirectly linked. Without loss of generality, suppose that $x \sim_a z$ and $z \sim_b y$. This implies that properties 5. and 6. hold between x and z and between z and y with respect to $\boldsymbol{\sigma}$ and $\boldsymbol{V}$ symmetric under $\sim_{a \cup b}$, as just shown. Since properties 5. and 6. are clearly transitive, they must then also hold between x and y.

Returning to the proposition, observe that if $\sim_a$ is a refinement of $\sim_b$, then $\sim_{a \cup b} = \sim_b$ and vice versa. If neither is a refinement of the other, then both are refinements of $\sim_{a \cup b}$. Every game has at least one symmetry structure, because the trivial partition is always one. Finally, we can now rule out that a game can have multiple maximal elements in terms of coarseness. Because if $\sim_a \neq \sim_b$ were both maximal, then $\sim_{a \cup b}$ would also be a symmetry structure and coarser than both – a contradiction.

Proposition: Existence of a symmetric equilibrium. *For every symmetry structure of a game, there exists a stationary equilibrium that is symmetric under that structure.*

As mentioned earlier, a strategy profile that is symmetric under the maximally coarse symmetry structure is also symmetric under all other symmetry structures, which are refinements of the former. Thus, it will suffice to show existence of an equilibrium that is symmetric under the maximal symmetry structure. Note that this equilibrium is Markov perfect, as defined in Section 4.2.

While other proofs of existence have been put forward (Maskin and Tirole, 2001), we want to take a slightly different route here and demonstrate the ability to use homotopy functions for constructive proofs. The idea is to establish the existence of a path that is itself symmetric, and leads to an equilibrium – which must then also be symmetric. We will use the QRE homotopy (Section 4.3.3 and Chapter 2) to that end.

First, note that symmetry of a game implies symmetry of u; to see this, simply set $\boldsymbol{V} = 0$ in properties 5. and 6. The strategy $\boldsymbol{\sigma}^0$ at the starting point of the QRE homotopy is given by $\sigma^0_{sia} = \frac{1}{|A_{si}|}$ for all s, i and $a \in A_{si}$. Since all symmetric agents must have the same number of actions (property 1.), $\boldsymbol{\sigma}^0$ clearly is symmetric. If u and $\boldsymbol{\sigma}$ are symmetric, values $\boldsymbol{V}^0$ at the starting point must also be symmetric. Thus, the path begins at the symmetric point $(\boldsymbol{\sigma}^0, \boldsymbol{V}^0, 0)$. Since u is symmetric, so are its derivatives (the derivative of $u(\boldsymbol{\sigma}_{si})$ with respect to σ_{sia} is simply $u(a, \sigma_{s,\text{-}i})$). The derivatives determine the direction of the path. Thus, as long as the current point is symmetric, the direction of the path is symmetric, and because the direction is symmetric, the points on the curve remain symmetric. In consequence, $\boldsymbol{\sigma}$ along the principal branch of the QRE homotopy is symmetric. In Chapter 2, we show that the limiting point of any branch of the QRE homotopy is a stationary equilibrium. Since the complete branch is symmetric, so must be its limit point. The existence of a symmetric equilibrium follows as claimed.

Note that the preceding argument also establishes that sgamesolver will always compute an equilibrium that is symmetric under the maximal symmetry structure of the given game when using the QRE homotopy. The same holds for LogTracing only if one ensures that the prior $\boldsymbol{\rho}$ and the vector of weights $\boldsymbol{\nu}$ are symmetric. One way to do so is to use the centroid strategy as prior, and a vector of ones as weights – which is actually the default in sgamesolver. sgamesolver also implements functionality to test symmetry structures, and make vectors like $\boldsymbol{\rho}$ and $\boldsymbol{\nu}$ symmetric. In general, symmetry may not only serve as a selection criterion, but can also be used to speed up computations. For each type of agents, it is possible to drop the rows and columns from H and its Jacobian J for all agents but the first from each symmetry class. For example, in the *Dynamic Price Competition* game, each agent has a symmetric agent of the other player in a corresponding state; thus, using symmetries would allow to half the time required for each evaluation of H. In addition, the algorithm involves the repeated computation of QR-decomposition and pseudo-inverse of J. Both operations scale cubically in the size of the system, so that halving the number of components of H should speed up these parts of the algorithm by a factor of 8. Some features regarding symmetries are currently in development and subject to change; thus, please refer to the online manual for up-to-date instructions.

4.6 Conclusion

This paper has introduced the package sgamesolver, a toolbox to compute stationary equilibria of finite discounted stochastic games via the homotopy method. The goal behind the package is to provide applied researchers with a tool that allows to solve models efficiently and without large investment into knowledge related to computational methods. In that regard, an important feature is the ability to solve games independently of their specific structure.

This paper has covered the basic usage, in particular how games can be defined and passed to the software. Some more advanced use cases were also discussed, e.g. the possibility to approximate the graph of a QRE correspondence. We also discussed the dependence of the LogTracing homotopy path on the prior in more depth than was possible in the original paper (Chapter 3), showed practically how to use prior search to uncover additional equilibria, and discussed the selective properties of this homotopy in an example. We presented the predictor-corrector-principle underlying the solver module of the package, and pointed out some potential pitfalls that can occur when using it. Finally, we discussed symmetries in stochastic games, again with a focus on their significance for users of sgamesolver.

We hope to extend the package in the future, by incorporating additional homotopies, but also wider functionality. One additional feature that is under development is to incorporate game dynamics. These are potentially interesting both as an object of study themselves, but also as a potential alternative tool to solve games. Inclusion in the current package will hopefully prove synergistic, as in terms of computation, there is considerable overlap between the evaluation of homotopy functions and dynamics. The close relation is particularly apparent in a dynamic derived from QRE, covered in Chapter 2. Another area for potential development concerns the steps after an equilibrium has been found, for example its graphical representation or its use in further computations. We always welcome feedback, suggestions, and contributions, and are always happy to learn how how the software is being used.

Appendix

4.A Game *Dynamic Price Competition*

4.A.1 Creating a Tabular Representation

The following code listing exemplifies how the *Dynamic Price Competition* game can be defined and then solved in sgamesolver. We use python here, and are creating and passing the table as a Pandas dataframe; however, other programs or languages can be used, and the table passed as a file. The structure of the code itself is rather simple. Three loops generate all the combinations of active firm, other firm's old price, and the active firm's new price. For each combination, a line is appended with state- and action labels as well as calculated profits.

```python
import sgamesolver, pandas as pd

# set up price grid and a profit function
price_grid = [round(0.1*n, 1) for n in range(12)]
def profit(own_price, other_price):
  if own_price > other_price:
    return 0
  elif own_price == other_price:
    return (2 - own_price) * own_price/2
  elif own_price < other_price:
    return (2 - own_price) * own_price

# prepare a table for the game, and append a single row for the deltas
table = pd.DataFrame(
    columns=["state", "a_firm0", "a_firm1", "u_firm0", "u_firm1", "to_state"])
table.loc[len(table)] = {"state": "delta", "u_firm0": 0.95, "u_firm1": 0.95}

# loop over active firm, other firm's price, own price
for active_firm in [0, 1]:
  inactive_firm = 1 - active_firm
  for inactive_firm_price in price_grid:
    for active_firm_price in price_grid:
      # append a row with: state, action labels, payoffs, transitions
      # python f-strings are used to piece together the labels
      table.loc[len(table)] = {
        f"state": f"firm{active_firm} active; p{inactive_firm}={inactive_firm_price}",
        f"a_firm{active_firm}": active_firm_price,
        f"a_firm{inactive_firm}": f"{inactive_firm_price} (inactive)",
        f"u_firm{active_firm}": profit(active_firm_price, inactive_firm_price),
        f"u_firm{inactive_firm}": profit(inactive_firm_price, active_firm_price),
        "to_state": f"firm{inactive_firm} active; p{active_firm}={active_firm_price}"
      }

game = sgamesolver.SGame.from_table(table)
```

4.A.2 Searching the Prior Space and Equilibrium Selection

We now return to the example game *Dynamic Price Competition.* Maskin and Tirole (1988) already showed that this game, independently of the exact parametrization, has two types of (symmetric) stationary equilibria: A *cycling* type, in which the firms generally repeatedly undercut each other; once the price reaches marginal costs, both firms mix between staying there and returning to a much higher price, from which the undercutting then repeats. The resulting saw-tooth price pattern is known as an Edgeworth cycle. The other type of equilibrium consists in both firms setting some fixed price $p_i = \bar{p}$, and retaliate only if the other undercuts. Such equilibria generally exist for many values of $\bar{p}$.

The following code snippet demonstrates how to perform a search of the prior space using the LogTracing homotopy with just a few lines of code. In the example, 10 000 priors are drawn at random, and the associated equilibrium is computed. A list stores all distinct equilibria and counts how often they occurred. It is assumed that the code from Appendix 4.A.1 has been run prior to create the game itself. As mentioned in Section 4.3.2.1, an alternative approach would be to set the prior to $\mathbf{0}$ and vary $\boldsymbol{\nu}$ randomly from trial to trial.

```
game = sgamesolver.SGame.from_table(table)
equilibria = []
n_runs = 10000
for run in range(n_runs):
    print(f"Run: {run}")
    # generate a random prior, using a seed for reproducibility
    rho = game.random_strategy(seed=run, zeros=True)
    # set up the LogTracing-homotopy, prepare and start the solver
    logtracing = sgamesolver.homotopy.LogTracing(game, rho=rho)
    logtracing.solver_setup()
    logtracing.solver.set_parameters(bifurcation_angle_min=177.5,
                                     max_steps=1000, verbose=0)
    logtracing.solve()

    if logtracing.equilibrium:
        for old_eq in equilibria:
            if np.allclose(logtracing.equilibrium.strategies,
                           old_eq.strategies, atol=0.001,
                           equal_nan=True):
                old_eq.count += 1
                break
        else:
            logtracing.equilibrium.count = 1
            equilibria.append(logtracing.equilibrium)
```

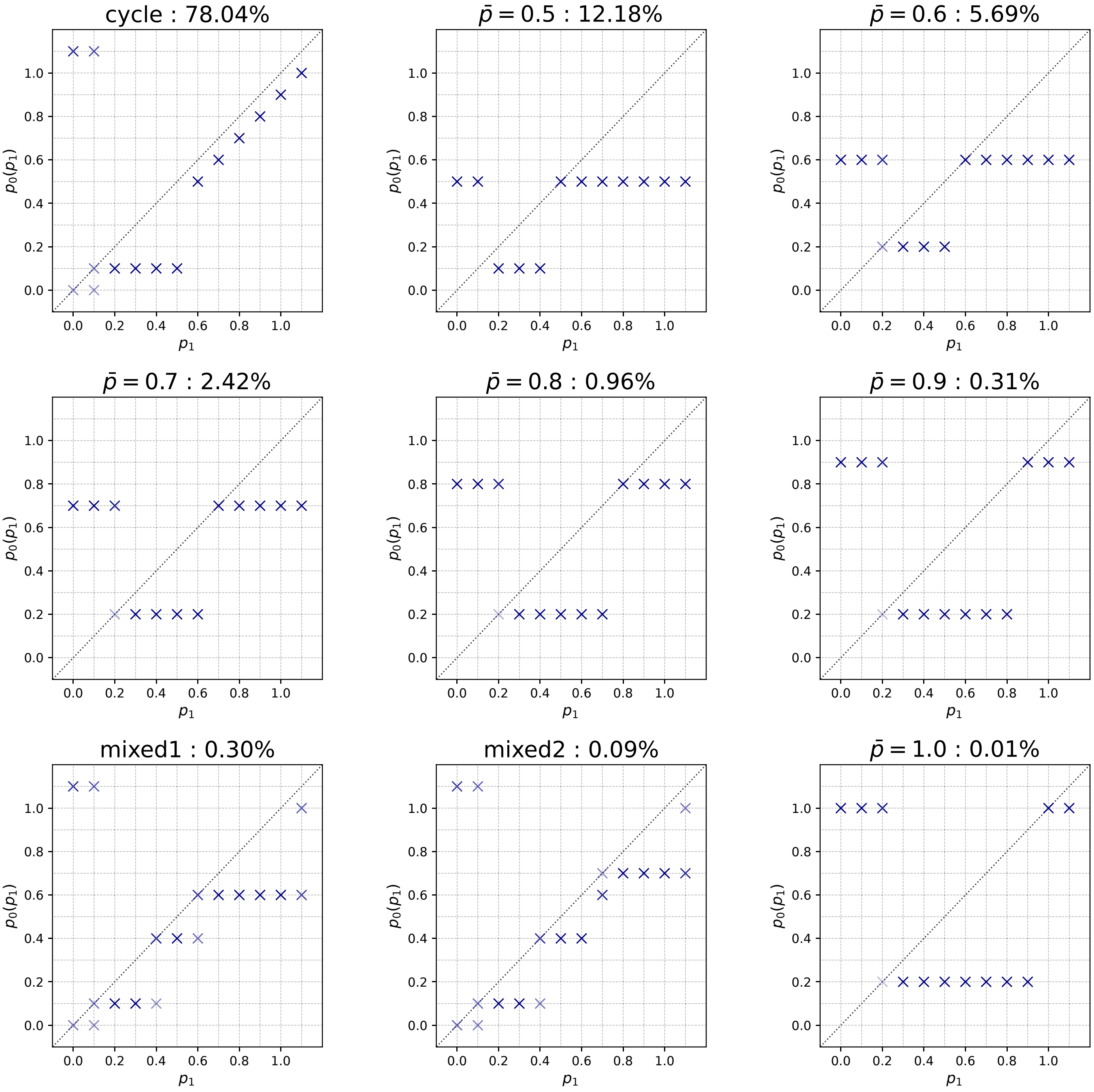


Figure 4.6: The nine symmetric equilibria of the game *Dynamic Price Competition* found by searching the (symmetric) prior space and their relative frequency.

When searching the prior space with 10 000 random, symmetric priors, we found 9 symmetric equilibria, which are shown in Figure 4.6. Each panel shows the equilibrium strategy of Firm 0; Firm 1's strategy is symmetric, but not shown. On the x-axis is the price set by Firm 1 previously; the y-axis then shows Firm 0's response; if a response is mixed, this is indicated by multiple markers, with darker shades indicating higher probability mass for the respective price. The title of each panel lists the type of the equilibrium and the relative frequency with which it was reached. The cycling equilibrium is by far the most common, arising from 78% of priors. Fixed price equilibria with $\bar{p}$ ranging from 0.5 to 1 occur, with a frequency decreasing in $\bar{p}$. Fixed price equilibria with prices lower than 0.5 are absent; presumably, these are in the set of equilibria that could only be reached via secondary branches. Finally, there are also the two somewhat mixed types *mixed*1 and *mixed*2, which feature both undercutting, but also ranges of fixed price responses.

We now briefly touch on the question of what seems to determine the relative sizes of the basins of attraction, in particular the prevalence of the cycling equilibrium. Harsanyi and Selten (1988) have shown that for 2 × 2 games, the risk-dominant equilibrium always has the largest basin of attraction under the tracing procedure, and that this equilibrium is always selected when starting from the centroid prior. Risk dominance is is tightly connected to the idea of strategic uncertainty; if the latter is high, the risk dominant equilibrium is arguably a plausible outcome, even if Pareto-dominated by another. A well-known example is the game *stag hunt* (Section 4.3.2.1), where the *hare*-equilibrium is risk-dominant, but payoff dominated. Unfortunately, the concept of risk dominance does not readily generalize. Even for larger one-shot-games there is no guarantee that a transitive relation of this sort between different equilibria; the situation in stochastic games clearly is no better.

Still, in the game *Dynamic Price Competition*, the selection by LogTracing does seem to resemble something similar to risk dominance. To asses how exposed to to strategic uncertainty of the different equilibrium strategies are, we simulated mis-coordinated equilibrium play in a tournament-like manner. For each pair of equilibrium strategies, we simulated the game 1000 times with 200 periods each. Table 4.4 lists the resulting average total discounted utility for the row-strategy when facing the column-strategy. As the diagonal shows, the $\bar{p}$-equilibria clearly Pareto-dominate the others, with value increasing in $\bar{p}$. However, if one considers what happens in case of mis-coordination, i.e. off the diagonal, it is apparent

	cycle	$\bar{p}=0.5$	$\bar{p}=0.6$	$\bar{p}=0.7$	$\bar{p}=0.8$	$\bar{p}=0.9$	$mix1$	$mix2$	$\bar{p}=1.0$
	78.0%	12.2%	5.7%	2.4%	1.0 %	0.3%	0.3%	0.1%	0.01%
cycle	6.59	3.70	5.64	6.09	6.43	6.64	5.32	5.28	6.76
$\bar{p}=0.5$	0.12	7.45	5.46	5.46	5.46	5.46	3.10	2.90	5.45
$\bar{p}=0.6$	1.79	1.80	8.35	2.36	2.42	2.48	4.98	1.80	2.43
$\bar{p}=0.7$	1.78	1.78	5.56	9.03	1.99	2.02	1.78	2.70	2.03
$\bar{p}=0.8$	1.77	1.77	5.53	5.93	9.52	1.78	1.76	1.77	1.72
$\bar{p}=0.9$	1.75	1.75	5.56	5.96	6.22	9.79	1.75	1.75	1.56
$mix1$	4.98	3.69	6.65	5.91	5.91	5.91	5.73	4.78	5.93
$mix2$	5.09	3.70	5.03	6.47	6.25	6.24	5.58	5.34	6.28
$\bar{p}=1.0$	1.74	1.74	5.56	5.94	6.27	6.44	1.74	1.74	9.89

Table 4.4: Total discounted utility for the row-strategy when facing column.

that *cycle* generally still does well, while the $\bar{p}$-equilibria often fare pretty badly, in particular against *cycle*. Performance of the *mix*-types against the $\bar{p}$-types is mostly similar to that of *cycle*; but *cycle* Pareto-dominates these, and also does better when paired against either.

Altogether, equilibrium *cycle*, by far the most prevalent, has properties that resemble risk-dominance, so that at least in this case and only in an informal sense, selection is in line with the results of Harsanyi and Selten (1988). At the same time, one should ask to what extent risk dominance constitutes a sensible criterion for selection among stationary equilibria. In situations in which mis-coordination is an important concern, the restriction to stationary strategies is perhaps harder to defend than usually, and players would be better off choosing a non-stationary strategy that allows to adapt after observing what the other players are doing. While we think that this is a topic worthy of discussion and study, it transcends the scope of this chapter.

To conclude discussion of the current example, we briefly mention what type of equilibria arise if one repeats the above process with asymmetric priors. Our results from solving the game 1 000 times in this manner are depicted in Figure 4.7. Markers in form of a cross signify Firm 0's equilibrium strategy, while circles show Firm 1's strategy with accordingly inverted axes. We found 12 equilibria, but include only 6, as the others are minor variations of those shown. We expect that still more will be found with a higher number of runs. Some of the

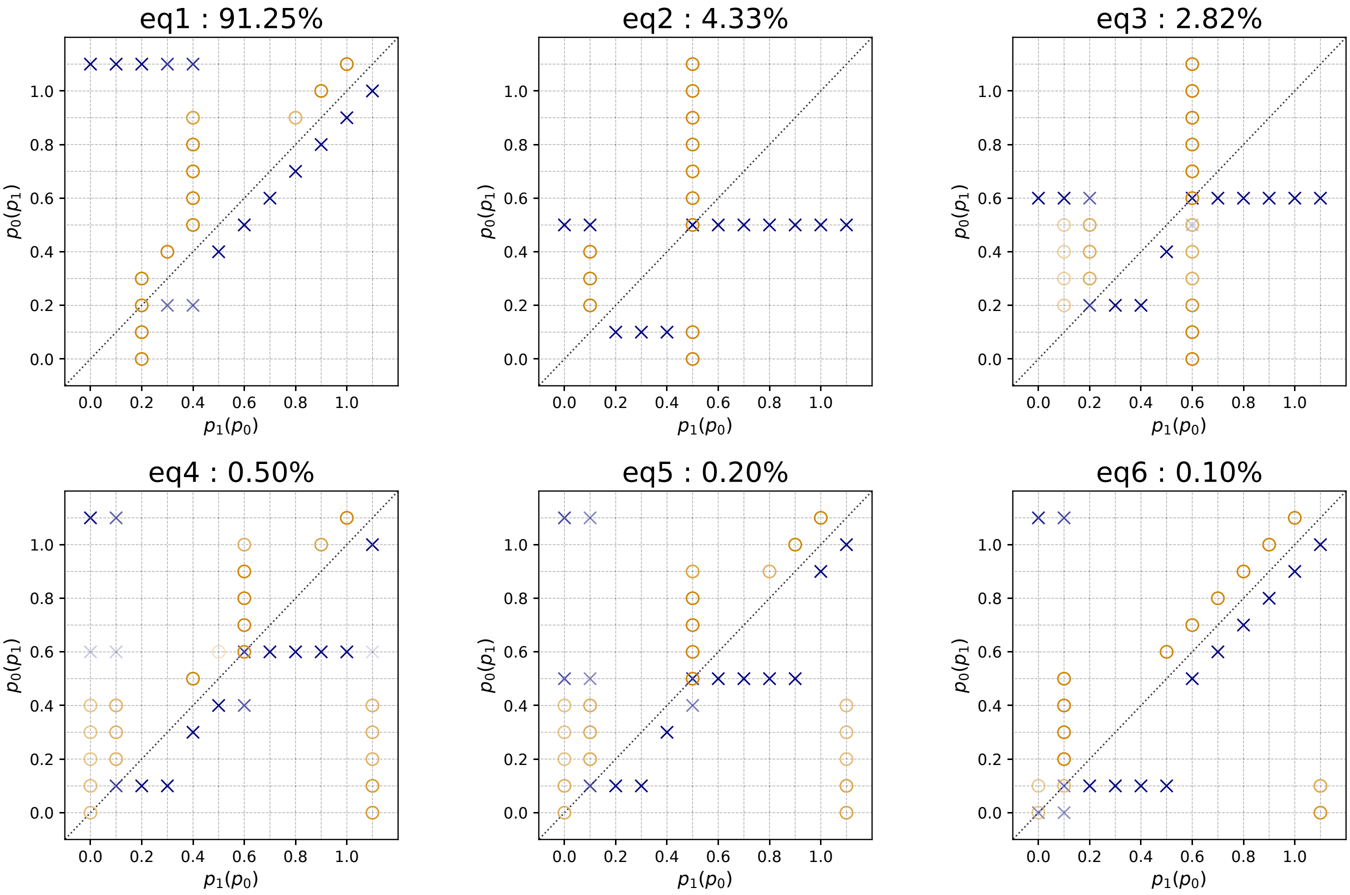


Figure 4.7: 6 of the 12 equilibria of the game *Dynamic Price Competition* we found by searching the asymmetric prior space and their relative frequency.

equilibria are symmetric, namely 2 and 6, but most are not. *eq1*, by far the most prevalent, essentially results in a cycle as well, even if both firms' strategies look quite different from the symmetric *cycle* equilibrium. Interestingly, the strategies also differ significantly from one another; in particular, it is always Firm 0 that initiates the resets. *eq6* on the other hand seems to resemble *cycle* perfectly – but is very rare in the asymmetric setting. The other equilibria are all variants of the $\bar{p}$-type equilibria; in particular, they all involve a fixed point on the main diagonal where $p_0(p_1) = p_1(p_0)$; once this price has been reached, neither firm will deviate.

4.B Finding Additional Equilibria in *Stag Hunt*

In this example, we demonstrate how to use the homotopy method to compute equilibria that are not directly reachable by varying the prior (Section 4.3.2.1). We will use the simple and well-known game *Stag Hunt*:

	stag	hare
stag	$10, 10$	$1, 8$
hare	$8, 1$	$5, 5$

The plan is to first identify two priors that lead to different equilibria; then use one prior, but start at the other equilibrium. In the given game, a prior of $(stag, stag)$ (or close to it) will lead to the equilibrium $(stag, stag)$. The same holds for *hare*. In other, more complex games, it might be necessary to experiment to identify the directly reachable equilibria first.

Turning to the code, we first define the game and the priors. Because the game is one-shot, we can use the method `.one_shot_game()` to create it from a single array containing the payoffs. We then define two homotopies, with the respective priors, and let them solve for the two pure equilibria. Then we create another homotopy, with the *stag* prior; but instead of the usual starting point, we start at the final point of the *hare* homotopy path. Here, we need some trickery: If we just started the solver, it would rightfully note that it already is at $t = 1$, i.e. at a solution – and do nothing. Thus, we tell it that we want a solution at a smaller value, e.g. $t = .99$, and then start it. Once we are at $t = .99$, we can set the target value back to 1 and let the solver walk the rest of the path. It then finds the mixed equilibrium.

```python
import sgamesolver
import numpy as np

payoff_matrix = np.array([[[10, 1],
                           [8, 5]],
                          [[10, 8],
                           [1, 5]]])
game = sgamesolver.SGame.one_shot_game(payoff_matrix)
game.action_labels = ['stag', 'hare']

stag_prior = np.array([[[1, 0],
                        [1, 0]]])
hare_prior = np.array([[[0, 1],
                        [0, 1]]])

# Find the first pure equilibrium, using 'stag' as prior:
homotopy_stag = sgamesolver.homotopy.LogTracing(game, rho=stag_prior)
homotopy_stag.solver_setup()
homotopy_stag.solve()
print(homotopy_stag.equilibrium)

# Find the second pure equilibrium, using 'hare' as prior:
homotopy_hare = sgamesolver.homotopy.LogTracing(game, rho=hare_prior)
homotopy_hare.solver_setup()
homotopy_hare.solve()
print(homotopy_hare.equilibrium)

# Create a homotopy to find the mixed equilibrium, using the stag prior:
homotopy_mixed = sgamesolver.homotopy.LogTracing(game, rho=stag_prior)
homotopy_mixed.solver_setup()
# But, set the final point (y) of the hare homotopy as current point:
homotopy_mixed.solver.y = homotopy_hare.solver.y.copy()
# If we just started now, the solver would (rightfully)
# think it is already at a solution.
# We therefore tell it to walk away from t=1 a bit:
homotopy_mixed.solver.t_target = 0.99
# Make sure the orientation points towards t_target, i.e. decreasing t:
homotopy_mixed.solver.set_greedy_sign()
# Now, we can start
homotopy_mixed.solve()
# The solver now reports it has found a point with t=.99
# We can set the target to t=1 again and keep going:
homotopy_mixed.solver.t_target = 1
homotopy_mixed.solve()
print(homotopy_mixed.equilibrium)
# This results in the final, mixed equilibrium
```

When run, the code outputs all three equilibria, as expected:

```
An equilibrium was found via homotopy continuation.
                      stag  hare
player0 : v=10.00, σ=[1.000 0.000]
player1 : v=10.00, σ=[1.000 0.000]
[...]
```

```
An equilibrium was found via homotopy continuation.
                     stag  hare
player0 : v=5.00, σ=[0.000 1.000]
player1 : v=5.00, σ=[0.000 1.000]
[...]
An equilibrium was found via homotopy continuation.
                     stag  hare
player0 : v=7.00, σ=[0.667 0.333]
player1 : v=7.00, σ=[0.667 0.333]
```

References

ABBRING, J. H., J. R. CAMPBELL, J. TILLY, AND N. YANG (2018): "Very Simple Markov-Perfect Industry Dynamics: Theory," *Econometrica*, 86, 721–735.

ALLGOWER, E. L. AND K. GEORG (1990): *Numerical Continuation Methods: An Introduction*, New York: Springer.

ALTMAN, E. (1996): "Non zero-sum stochastic games in admission, service and routing control in queueing systems," *Queueing Systems*, 23, 259–279.

BATES, D. J., J. D. HAUENSTEIN, A. J. SOMMESE, AND C. W. WAMPLER (2008): "Adaptive multiprecision path tracking," *SIAM Journal on Numerical Analysis*, 46, 722–746.

BESANKO, D., U. DORASZELSKI, Y. KRYUKOV, AND M. SATTERTHWAITE (2010): "Learning-by-Doing, Organizational Forgetting, and Industry Dynamics," *Econometrica*, 78, 453–508.

BORKOVSKY, R. N., U. DORASZELSKI, AND Y. KRYUKOV (2010): "A User's Guide to Solving Dynamic Stochastic Games Using the Homotopy Method," *Operations Research*, 58, 1116–1132.

BREITMOSER, Y., J. H. TAN, AND D. J. ZIZZO (2010): "Understanding Perpetual R&D Races," *Economic Theory*, 44, 445–467.

CHOI, S., D. A. HARNEY, AND N. BOOK (1996): "A Robust Path Tracking Algorithm for Homotopy Continuation," *Computers & Chemical Engineering*, 20, 647–655.

DANG, C., P. J.-J. HERINGS, AND P. LI (2022): "An Interior-Point Differentiable Path-Following Method to Compute Stationary Equilibria in Stochastic Games," *INFORMS Journal on Computing*, 34, 1403–1418.

DORASZELSKI, U. AND M. SATTERTHWAITE (2010): "Computable Markov-Perfect Industry Dynamics," *The RAND Journal of Economics*, 41, 215–243.

ERICSON, R. AND A. PAKES (1995): "Markov-Perfect Industry Dynamics: A Framework for Empirical Work," *Review of Economic Studies*, 62, 53–82.

FINK, A. M. (1964): "Equilibrium in a Stochastic n-Person Game," *Journal of Science of the Hiroshima University, Series A-I (Mathematics)*, 28, 89–93.

Goeree, J. K., C. A. Holt, and T. R. Palfrey (2016): *Quantal Response Equilibrium: A Stochastic Theory of Games*, Princeton, New Jersey: Princeton University Press.

Goettler, R. L., C. A. Parlour, and U. Rajan (2005): “Equilibrium in a Dynamic Limit Order Market,” *The Journal of Finance*, 60, 2149–2192.

Govindan, S. and R. Wilson (2009): “A Global Newton Method for Stochastic Games,” *Journal of Economic Theory*, 144, 414–421.

Harsanyi, J. C. (1973): “Oddness of the number of equilibrium points: A new proof,” *International Journal of Game Theory*, 2, 235–250.

Harsanyi, J. C. and R. Selten (1988): *A General Theory of Equilibrium Selection in Games*, Cambridge, Massachusetts: MIT Press.

Herings, P. J.-J. and R. Peeters (2010): “Homotopy methods to compute equilibria in game theory,” *Economic Theory*, 42, 119–156.

Herings, P. J.-J. and R. J. Peeters (2003): “Equilibrium Selection in Stochastic Games,” *International Game Theory Review*, 5, 307–326.

——— (2004): “Stationary Equilibria in Stochastic Games: Structure, Selection, and Computation,” *Journal of Economic Theory*, 118, 32–60.

Maskin, E. and J. Tirole (1988): “A Theory of Dynamic Oligopoly II: Price Competition, Kinked Demand Curves, and Edgeworth Cycles,” *Econometrica*, 56, 571–599.

——— (2001): “Markov perfect equilibrium,” *Journal of Economic Theory*, 100, 191–219.

McKelvey, R. D. and T. R. Palfrey (1995): “Quantal Response Equilibria for Normal Form Games,” *Games and Economic Behavior*, 10, 6–38.

Mertens, J.-F., S. Sorin, and S. Zamir (2015): *Repeated Games*, Cambridge: Cambridge University Press.

Miranda, M. J. and P. L. Fackler (2004): *Applied Computational Economics and Finance*, Cambridge, Massachusetts: MIT Press.

Pakes, A. and P. McGuire (1994): “Computing Markov Perfect Nash Equilibria: Numerical Implications of a Dynamic Differentiated Product Model,” *RAND Journal of Economics*, 25, 555–589.

Shapley, L. S. (1953): "Stochastic Games," *Proceedings of the National Academy of Sciences*, 39, 1095–1100.

Solan, E. and N. Vieille (2015): "Stochastic games," *Proceedings of the National Academy of Sciences*, 112, 13743–13746.

Takahashi, M. (1964): "Equilibrium Points of Stochastic, Noncooperative n-Person Games," *Journal of Science of the Hiroshima University, Series A-I (Mathematics)*, 28, 95–99.

Turocy, T. L. (2005): "A Dynamic Homotopy Interpretation of the Logistic Quantal Response Equilibrium," *Games and Economic Behavior*, 51, 243–263.

——— (2010): "Computing Sequential Equilibria Using Agent Quantal Response Equilibria," *Economic Theory*, 42, 255–269.

Zangwill, W. I. and C. B. Garcia (1981): *Pathways to Solutions, Fixed Points, and Equilibria*, Upper Saddle River, New Jersey: Prentice-Hall.

Zinkevich, M. A. (2006): "Generalized Symmetry in Stochastic Games," *Working paper.*