Modeling rational agents with limited capability

by

Kai Jia

B.S., Tsinghua University, 2015
M.S., Massachusetts Institute of Technology, 2021

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2024

Authored by:	Kai Jia Department of Electrical Engineering and Computer Science July 19, 2024
Certified by:	Martin C. Rinard Professor of Electrical Engineering and Computer Science Thesis Supervisor
Accepted by:	Leslie A. Kolodziejski Professor of Electrical Engineering and Computer Science Chair, Department Committee on Graduate Students

Modeling rational agents with limited capability

by

Kai Jia

Submitted to the Department of Electrical Engineering and Computer Science
on July 19, 2024 in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

ABSTRACT

In many scenarios, players exhibit inherent limitations in various aspects of their capability to generate maximally rational play in strategic games. Modeling such capability limitations and elucidating their implications will advance our understanding of the strategic interactions among players. In this thesis, I study two novel settings where players have limited capabilities. I formalize a hierarchy of capabilities and study related equilibrium concepts, computational complexity, solution algorithms, and the impact of varying capabilities on game outcomes.

The first limited-capability setting is limited-perception games. I focus on a class of one-shot limited-perception games. Such games extend simultaneous-move normal-form games by presenting each player with an individualized perception of the true game. Players' payoffs are determined by the true game hidden from players. The accuracy of a player's perception is determined by the player's capability level, with a higher level corresponding to a more accurate perception. I study both capability-oblivious and capability-aware players. A capability-oblivious player does not know they have limited perception and therefore plays the optimal strategy of their perceived game. I present payoff bounds and other predictable behavior of capability-oblivious players in a special class of limited-perception games. A capability-aware player reasons with the set of possible true payoff functions and other players' perceptions and incentives to maximize their own objective (e.g., the worst-case payoff) based on their limited perception. I present novel formalizations of simultaneous-move equilibria and show the hardness of equilibrium solving. I further present positive results that (i) an approximate equilibrium has a compact, tractable representation; and (ii) a few classes of zero-sum games can be efficiently solved.

The aforementioned efficiently solvable zero-sum games are reduced to solving nonsmooth convex programs. To this end, I present the Trust Region Adversarial Functional Subdifferential (TRAFS) algorithm for constrained optimization of unstructured nonsmooth convex Lipschitz functions. Unlike previous methods that assume a subgradient oracle model, I propose the functional subdifferential, defined as a set of subgradients that simultaneously captures sufficient local information for effective minimization while being easy to compute for a wide range of functions. Intriguingly, the TRAFS design also incorporates game-theoretical thinking. In each iteration, TRAFS solves a zero-sum game between the optimizer and a local approximation of the objective function to guarantee progress. The optimizer has access to step vectors in a local ℓ_2-bounded trust region; the local approximation uses the functional subdifferential. TRAFS finds an approximate solution with an absolute error

up to ϵ in $\mathcal{O}(1/\epsilon)$ or $\mathcal{O}\left(\sqrt{1/\epsilon}\right)$ iterations depending on whether the objective function is strongly convex, improving the previously best-known bounds of $\mathcal{O}((1/\epsilon)^2)$ and $\mathcal{O}(1/\epsilon)$ in these settings. TRAFS makes faster progress if the functional subdifferential satisfies a locally quadratic property; as a corollary, TRAFS achieves linear convergence (i.e., $\mathcal{O}(\log 1/\epsilon)$) for strongly convex smooth functions. In the numerical experiments, TRAFS solves twice as many problems compared to the second-best method and is on average 39.1x faster on problems solved by both methods.

The second limited-capability setting is limited-strategy games where a player's capability limits the strategies available to them. I work with a formalization where a player's strategy space is defined as programs in a Domain-Specific Language (DSL). A player's capability limits the size of programs available to that player. I focus on characterizing the impact of player capability on game outcomes. I study a new game model called McDncDa derived from network congestion games. I show that it is computationally hard to determine whether an McDncDa instance is capability-positive (i.e., whether increasing a player's capability level leads to a better payoff). I then study a parameterized special class of McDncDa called MGMG. I show that MGMG is always capability-positive, and it is socially capability-positive (i.e., the sum of all players' payoffs always gets better if every player's capability level is increased by one) if some resources in the game have increasing returns to scale despite the existence of multiple equilibria.

Thesis supervisor: Martin C. Rinard
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I own my deepest gratitude to my advisor, Martin Rinard. This thesis research would be impossible without his guidance and support. Over the years, Martin has been guiding his students with his unique research taste and his unique sense of humor. He does not only give us valuable insights on research and great suggestions on how to survive the academia world, but also provides us with his profound wisdom to help us become a better person. His unlimited energy, unceasing patience, unconditional support, unconstrained enthusiasm, and unbounded imagination have been a great inspiration to me. I feel extremely fortunate to have Martin as my advisor.

I would like to thank my thesis committee members, Gabriele Farina and Pablo Parrilo, for their insightful comments and suggestions. I am grateful to their support to accommodate my schedule and make everything happen in a short time frame despite their busy schedules.

I want to thank my labmates, especially Yichen Yang, Charles Jin, Farid Arthaud, Idan Orzech, José Cambronero, Rem Yang, Nikos Vasilakis, Shivam Handa, Sara Achour, Thurston Dang, Jürgen Cito, Jeff Perkins, Deokhwan Kim, and Malavika Samak, for all the fun and insightful chats we had. I have enjoyed my collaboration with Yichen on limited-strategy games that led to this thesis. Yichen is an innovative thinker and a great collaborator, with a deep understanding of technical details. Yichen pioneered the limited-capability game theory research in our group. I am also grateful to Fan Long for introducing Martin and me to each other, which is a necessary condition for me to be here. Furthermore, I'd like to thank Lingxiao Li and Zhengyang Jiang; we have known each other since high school, and magically, they crossed paths with me again during my research career. Both Lingxiao and Zhengyang have provided me with valuable support on my research.

Another important research project that I have worked on, although not included in this thesis, is predicting risks of severe yet hard-to-diagnose diseases such as pancreatic cancer through routinely collected electronic health records (EHR). I am grateful to my collaborators, including Limor Appelbaum, Pasapol Saowakon, Jeff Warnick, José Cambronero, and many others, for their support, insights, and efforts to make this project happen. This project is still ongoing. I hope to see it deployed in the real world soon to help save lives.

Besides research, I also had a great time with my friends at MIT. I am fortunate to have worked with many talented people, including Husayn Karimi, Ajay Brahmakshatriya, Daniel Shen, Nishad Gothoskar, Belinda Li, Chenyang Shao, and many more, to help build the MIT Graduate Student Union. Our efforts finally led to better working and living conditions for graduate students at MIT. I have enjoyed the beautiful nights out stargazing and astrophotography with my friends, including Sihan Chen, Sheng Huang, Di Tong, Irene Ko, and others. The stargazing experience is an invaluable memory that I will cherish forever.

Finally, I want to thank my family for their unconditional love and support. My wife, Qi, has been my biggest supporter and my best friend. We went through our individual PhD journeys together, sharing our excitement and frustration.

Qi and I are both grateful to the love and emotional support we received from our cats, Miki and Suki. Cats are intelligent, amazingly incomprehensible creatures. Remarkably, Suki won the fight against Feline Infectious Peritonitis (FIP). Although the treatment of FIP is still not a standard vet practice in the US, I am grateful to the research done by Dr. Niels Pedersen and his team at UC Davis [Ped20] and more recent work by Zwicklbauer et al. [Zwi+23], which allowed us to treat Suki effectively without too much pain. We are also grateful to the joy brought by other short-term foster cats we had, including Wes, Winnie, Lila, Eenie, Cora, Caden, and Binky. Before Suki came, we painfully lost Binky to FIP. I hope people can have better awareness of FIP and better access to the treatment in the future so less cats will suffer from this deadly disease.

There are many more people I want to thank. I apologize for not being able to list everyone here. I am grateful to all the people who have helped me in my graduate school journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context, motivation, and result overview

Modeling human behavior is a central topic in many fields, including sociology, economics, psychology, and computer science. One highly influential approach is to model human behavior as strategic interactions between self-interested rational agents, which leads to the field of game theory [VM67]. A typical assumption in game theory is that each agent chooses a strategy to maximize their utility represented as a real number; each agent's decision-making process also considers the strategies and responses of other agents. Prominent topics in game theory research include the form and representation of games (e.g., normal form for simultaneous-move games and extensive form for sequential games), solution concepts (e.g., Nash equilibria and correlated equilibria), and computational aspects such as computational hardness and algorithms for solving games (e.g., PPAD-completeness, optimization, and learning) [MZS13; Rou16; Nis+07]. In game theory, a rational agent is often referred to as a *player*.

Real-world agents often have different capabilities. In a game-theoretic context, a player's *capability* can be broadly defined as a certain aspect of their ability that impacts their decision-making process. A higher capability can often enable a player to better exploit some aspects of the game, such as using more complex strategies, perceiving the environment or the game more accurately, reasoning more deeply, or communicating more efficiently. One can anticipate that different player capabilities can influence game play and game outcomes.

This thesis studies novel settings of game theory in which players have different capabilities. I consider two aspects of these different capabilities: *limited perception capabilities* (when a player perceives approximations of all players' true payoff functions) and *limited strategy capabilities* (when a player has a limited set of usable actions). Figure 1.1 illustrates a hierarchy of limited-capability game models explored in this thesis. This thesis considers the following research questions. Table 1.1 summarizes how those questions are addressed in different parts of this thesis.

Question 1: How to model player capabilities?

I study *limited-capability games*, where each player is assigned a capability from a set of possible capabilities. The capability of a player impacts some aspect of their game play; the

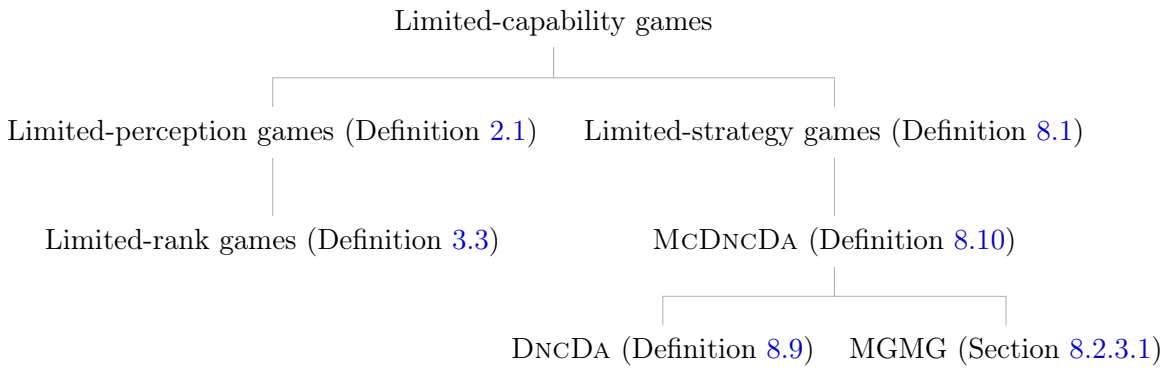


Figure 1.1: A hierarchy of limited-capability game models explored in this thesis.

exact impact depends on the specific setting. I represent a player's capability level in a game instance as an extended natural number (i.e., either a natural number or ∞). In this way, all capabilities are totally ordered. To study the impact of increasing or decreasing player capabilities, we can define the next or previous capability level of a player (when they have a finite capability level). Of course, the model can be extended to capabilities represented by uncountable sets, lattices, or even only partially-ordered sets; whether there are interesting results in those extensions is left as a future research direction.

I consider two instantiations of limited-capability games: (i) *limited-perception games* in which a player's capability limits their perception of the environment; and (ii) *limited-strategy games* in which a player's capability limits the complexity of their strategy. I present formal definitions that capture a large set of limited-capability games in both settings. A few games under more concrete settings are analyzed in detail.

Question 2: What is the rational behavior of players in limited-capability games?

For limited-perception games, I work with variations of finite one-shot normal-form non-cooperative games where players use randomized strategies to maximize their expected payoff. An equilibrium is achieved when no player has an incentive to unilaterally deviate from the strategy prescribed by the equilibrium. Even in this seemingly straightforward setting, player capabilities introduce new challenges. We are faced with the choice of different epistemic models for the players regarding their capabilities, i.e., what knowledge a player has about their own and others' capabilities, and what knowledge a player has about others' knowledge of players' capabilities, and so on.

I analyze two player types for limited-perception games: *capability-oblivious* players who do not know that their perception is limited and play the optimal strategy of their perceived game, and *capability-aware* players who know their own and others' capabilities and use this knowledge in their reasoning. I consider two types of games given the two player types: the first with all players being capability-oblivious, and the second, called *fully-capability-aware games*, with all players knowing as common knowledge that all players are capability-aware.

For limited-strategy games, since a player at least knows their own capability so that they can choose a valid strategy, I focus on analyzing fully-capability-aware games. I also briefly discuss players with incomplete information about others' capabilities in limited-capability games. I propose novel variations of network congestion games to study the behavior of

Table 1.1: Summary of sections corresponding to research questions.

Game model	**Setting(s)**	**Topic(s)**	**Section(s) / Chapter**
Limited-perception games	Fully-capability-aware general-sum	Equilibrium formulation, existence, and hardness	4.2
		Capability transfer properties	4.4
	Two-player fully-capability-aware zero-sum	Equilibrium formulation	4.3.1
		Equilibrium hardness	4.3.2
		Efficiently solvable games	4.3.3 and 7
Limited-rank games	Capability-oblivious	Miscellaneous properties	3.2 and 3.3
	Capability-oblivious and fully-capability-aware	Capability transfer properties	3.4 and 5
McDncDa	Fully-capability-aware	Hardness of equilibrium and capability transfer properties	8.2.2
MGMG	Two-player fully-capability-aware	Equilibrium values and capability transfer properties	8.2.3

limited-strategy players in those games.

Question 3: How hard is it to compute an equilibrium in limited-capability games?

I analyze the hardness of solving limited-capability games. I present hardness results of solving a few classes of games, including general limited-perception games, two-player zero-sum limited-perception games, and a class of limited-strategy games derived from network congestion games. These hardness results imply that computing a solution of those games, even an approximate solution, likely requires at least exponential time in the worst case.

For fully-capability-aware limited-perception games, since a lower-capability player needs to consider a potentially infinite set of possible perceptions of the higher-capability player, even representing an equilibrium can require infinite space if done naively. Despite the hardness, I present two positive results. First, I show that although the aforementioned naive representation can require infinite space, there are compact representations of approximate equilibria. Second, by imposing additional constraints on the perception process, I identify three classes of efficiently solvable limited-perception zero-sum games. Numerical solutions to those efficiently-solvable games can be obtained via nonsmooth convex optimization. Chapter 7 is dedicated to a new algorithm for nonsmooth convex optimization with better iteration complexity than previous methods. Intriguingly, the design of the new algorithm incorporates game-theoretic adversarial thinking to achieve this faster convergence rate.

Question 4: What are the capability transfer properties?

Given a limited-capability game, a *capability transfer property* describes how a player's capability level (or some players' capability levels) impacts the game outcome. There are

many properties that one can study, including:

- If one player increases their capability, how will their payoff change? This question can be asked under many different contexts, including whether the players are capability-oblivious or capability-aware, whether the game is limited-perception or limited-strategy, is the player interested in their final true payoff, the worst case of their payoff, or the expected payoff over games sampled from certain distributions, etc.
- If multiple players increase their capabilities simultaneously, how will the social welfare (i.e., sum of players' payoffs) change? Again, this question can be asked under many different contexts.

I study a few capability transfer properties for both limited-perception and limited-strategy games. One result is that determining whether a player obtains a better payoff with higher capability in certain limited-perception and limited-strategy games is as computationally hard as solving the games themselves. Consequently, a higher-capability player does not necessarily obtain a better payoff; from a player's perspective, if they have the opportunity to increase their capability (potentially at some cost), it is computationally challenging to decide if they should do so. Nevertheless, I demonstrate via numerical experiments that in a class of limited-perception games with randomly sampled payoffs, a player benefits from a higher capability in the totally competitive (i.e., zero-sum) setting, but in the general-sum setting, a player should match their opponent's capability to maximize their own payoff. I also analyze a class of limited-strategy games and show that a higher-capability player obtains a better payoff despite the existence of multiple equilibria.

Publication status At the time of writing, most of the results in Chapter 4 are available on arXiv [JR24a]. The results in Chapter 7 are available on arXiv [JR24b] and under review by a journal. Chapter 8 extends the published work Yang, Jia, and Rinard [YJR22], where Yichen and I shared equal contributions.

1.2 Limited-perception games

1.2.1 Motivating limited-rank games

Almost all beings in the physical world, including humans, can perceive reality only approximately. They need to make the best decisions based on the limited information they have. To formalize this idea in a game-theoretic setting, I consider an agent with "blurry vision." I use the *low-rank approximation* to model blurry vision. Let $\boldsymbol{M}$ be a matrix representing reality. A rank-r approximation of $\boldsymbol{M}$ is a matrix $\boldsymbol{M}'$ that minimizes $\|\boldsymbol{M} - \boldsymbol{M}'\|_F$ subject to $\operatorname{rank}(\boldsymbol{M}') \leq r$. A rank-$r$ matrix with $m \times n$ entries can be represented by $r(m+n)$ numbers, which is much smaller than mn when $r \ll \min\{m, n\}$. As shown in Fig. 1.2, a low-rank approximation can produce visually appealing results with a much smaller amount of data. Low-rank approximation techniques and their variants have found applications in many fields, including signal processing for noise reduction [SJ03], bioinformatics for genomic

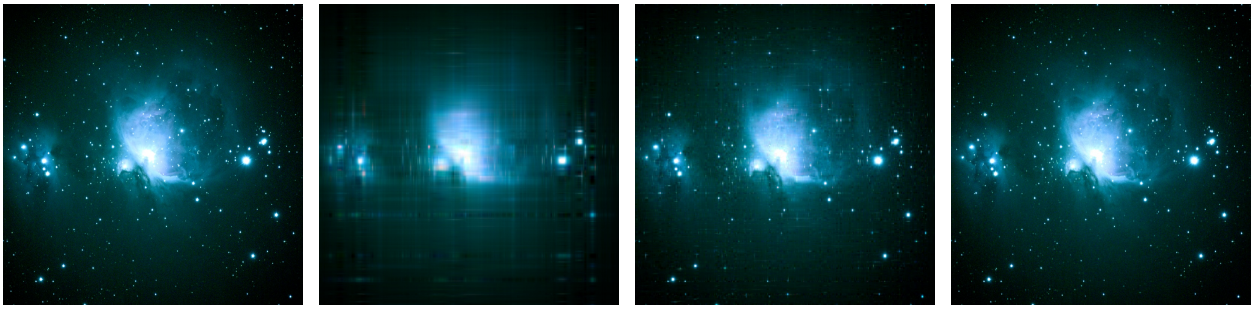


Figure 1.2: An example of low-rank approximations of an image (the Orion Nebula M42, photographed by the author).
From left to right: the original image and the low-rank approximations with 1%, 5%, and 20% of the original rank. The low-rank approximation is obtained via truncated Singular Value Decomposition (SVD) on the three color channels (red, green, blue) separately. A lower rank approximation only preserves the most significant structures, while a higher rank approximation retains more details.

data processing [ABB00], recommendation systems for user personalization [KBV09], and finance for portfolio allocation [FLM13].

Given the above motivation, it is reasonable to define *limited-rank games* where a player with a capability level c perceives rank-c approximations of both players' true payoff matrices in a normal-form game (see Section 1.6.2 for a brief formal introduction to normal-form games). The true payoff matrices are unknown to the players; players therefore make decisions based on their perceived games. For a matrix $\boldsymbol{M}$, let $\widetilde{\boldsymbol{M}}^{(r)}$ denote its rank-r approximation. For a bimatrix game $(\boldsymbol{A}, \boldsymbol{B})$, when the players have capability levels c_1 and c_2, respectively, they perceive the game as $\left(\widetilde{\boldsymbol{A}}^{(c_1)}, \widetilde{\boldsymbol{B}}^{(c_1)}\right)$ and $\left(\widetilde{\boldsymbol{A}}^{(c_2)}, \widetilde{\boldsymbol{B}}^{(c_2)}\right)$, respectively. If the players are capability-oblivious, they solve the Nash equilibria $(\boldsymbol{x}_1, \boldsymbol{y}_1)$ and $(\boldsymbol{x}_2, \boldsymbol{y}_2)$ of their respective perceived games. Their final combined strategy profile is $(\boldsymbol{x}_1, \boldsymbol{y}_2)$. If $c_1 \neq c_2$, then $(\boldsymbol{x}_1, \boldsymbol{y}_2)$ is not necessarily an equilibrium of any game that satisfies the conventional definition of equilibrium. Nevertheless, there are interesting, predictable outcomes in this setting, as shown in Chapter 3.

I further argue that limited-rank players may naturally emerge in a game-theoretic setting. Since solving a general-sum game is computationally hard, it is unclear (and I believe unlikely) whether real-world rational players can achieve a Nash equilibrium in an intricate game. In contrast, if both payoff matrices are of rank-k, then an exact Nash equilibrium of support size $k+1$ exists, which enables computationally feasible solutions for small k by enumerating all $(k+1)$-sized supports [LMM03]. Moreover, some natural games, such as poker, have an inherently low-rank structure [FS22]. Hence, in scenarios involving computationally bounded players interacting, certain individuals may evolve to perceive low-rank approximations of the true game, akin to how humans tend to simplify their reasoning in complex real-world situations by concentrating on salient aspects.

1.2.2 Introducing limited-perception games

Limited-perception games generalize the idea behind limited-rank games. The low-rank approximation is a specific perception model. In a limited-perception game, all players share a *payoff perception function* that maps a pair of *true payoff function* and an integer capability level to a *perceived payoff function*. The perceived payoff function is also called an *abstraction* of the true payoff function. The payoff perception function is characterized by three axioms (Definition 2.2):

1. *Capability path independence*: an abstraction obtained by applying the payoff perception function multiple times with different capability levels in an arbitrary order is the same as applying the perception payoff function once with the minimum capability level.

2. *Perfect perception with infinite capability*: any payoff function can be perfectly perceived by a player with an infinite capability level.

3. *Bounded concretization*: given any abstraction, the payoff values of compatible true games are bounded.

The first axiom ensures that a single capability level (rather than a path of capability levels) determines a perception, which induces a hierarchical structure over the perception accuracy (e.g., a higher-capability player can predict a lower-capability player's perception). The other two axioms induce desirable properties of true payoff bounds (Theorem 2.8). Besides limited-rank games, this model captures a range of games with limited payoff perceptions, such as perceiving payoff values quantized to a given decimal place and perceiving a payoff function whose entries with small absolute values are masked out (see Examples 2.1 to 2.3).

Chapters 2 to 6 study limited-perception games. Chapter 2 presents the formal model of limited-perception games and analyzes their general properties. Chapter 6 reviews previous work related to limited-perception games. Other chapters study specific classes or properties of limited-perception games.

1.2.3 Capability-oblivious players in limited-rank games

As mentioned earlier, Chapter 3 studies capability-oblivious players in limited-rank games. Section 3.1 formalizes limited-rank games. Although the best low-rank approximations of a given matrix can be nonunique, I show that a payoff perception function can be well defined by imposing some selection rule on the set of low-rank approximations. Therefore, a limited-rank game is a limited-perception game.

To analyze the outcomes of capability-oblivious players in limited-rank games, one crucial quantity is the bounds of possible true payoffs given a low-rank perception and a mixed strategy profile. Theorem 3.3 presents analytical forms of the true payoffs' lower and upper bounds. The bounds can be expressed as second-order cone programs, which enables efficiently solving certain equilibria as will be shown in Chapter 4.

Although the combined strategy profile of capability-oblivious players is not necessarily an equilibrium of any game, there are still predictable outcomes as studied in Section 3.3. Section 3.3.1 analyzes to what extent a low-rank approximation can encode the best-response

behavior of a player; in particular, Theorem 3.8 presents a sufficient condition so that the set of pure Nash equilibria of a perceived game is the same as the set of pure Nash equilibria of the true game. Theorem 3.10 shows that in a zero-sum game, the optimal strategy of the perceived game is an ϵ-optimal strategy of the true game.

Section 3.3.3 analyzes an alternative setting called *uncoupled dynamics* for capability-oblivious players in general-sum limited-rank games. Uncoupled dynamics considers repeated play of the same game. Each player only accesses their own payoff function and the game history. Each player maintains a model of other players' behaviors instead of analyzing other players' incentives using others' payoff functions. Capability-oblivious players is a natural fit for uncoupled dynamics since each player only needs to access their own perceived payoff function. Theorem 3.11 shows that fictitious play [Bro51], one of the simplest uncoupled dynamics, ensures convergence for capability-oblivious players in limited-rank games if any player has a capability level of 1; Example 3.2 presents a non-convergence game when two rank-two players use fictitious play.

Section 3.4 studies capability transfer properties; in particular, I try to answer the question of whether a capability-oblivious player can benefit from increasing their capability level in a limited-rank game. Theorem 3.12 presents a negative result that it is PPAD-hard to decide whether a player gets a strictly better payoff with a full-rank capability compared to a rank-one capability. This hardness motivates the study of a 2×2 parameterized game in Section 3.4.2. I study a two-player 2×2 *defense game* parameterized by (C, k_1, k_2, β). There is a target of value C. The defender can choose to defend the target with a cost of k_1 or not defend with zero cost. If the defender does not defend, players' final payoffs are either $(C, 0)$ or $(0, C)$ depending on whether the attacker attacks. If the defender defends, the attacker needs to pay a cost of k_2 to attack to obtain a proportion $0 < 1 - \beta < 1$ of the target value; the final payoffs are either $(C - k_1, 0)$ or $(C\beta - k_1, C(1 - \beta) - k_2)$ depending on whether the attacker attacks. I fully characterize the game outcomes when one player has a rank-one capability and the other player changes their capability level from 1 to 2 in Theorem 3.16, which shows that the change of payoffs and social welfare can be either positive or negative depending on the game parameters.

1.2.4 Capability-aware players in limited-perception games

Chapter 4 studies *fully-capability-aware games*. Although a capability-aware player may choose to optimize their best payoff, worst payoff, or uncertainty (the gap between the best and worst payoffs), Section 4.1 argues that maximizing the worst-case payoff is the most reasonable choice among the three since it dominates the uncertainty-minimization objective by giving not worse upper and lower payoff bounds, and maximizing the best-case payoff may result in no equilibrium (Example 4.1). Therefore, I define players in a fully-capability-aware game as players with the common knowledge that every player knows (i) the payoff perception function; (ii) the capability levels of all players; and (iii) that each player maximizes their worst-case expected payoff given their limited perception of the true payoff function. Of note, in a fully-capability-aware game, each player knows their own perceptions of all players' true payoff functions privately.

Equilibrium concepts The players use their (limited) perception of the true payoffs to reason about the incentives and resulting actions of all players in a fully-capability-aware game. In a conventional normal-form game, the Nash equilibrium is defined as a pair of mixed strategies $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ so that no player can improve their expected payoff by unilateral deviation [Nas50]. However, using a mixed strategy pair is inadequate in the limited-capability setting. In the two-player case with the first player having a lower capability, because the first player has only limited knowledge of the second player's perception, the first player cannot solve $\boldsymbol{y}^*$ using their private information; instead, they must consider all possible perceptions of the second player. I propose a new equilibrium formalization $(\boldsymbol{x}^*, \boldsymbol{Y})$ (Definition 4.5), where $\boldsymbol{x}^*$ is a mixed strategy of the first (i.e., lower-capability) player, and $\boldsymbol{Y}(\cdot)$ is a higher-order response function that maps each possible perception of the second player (from the first player's perspective) to a mixed-strategy best response of the second player for that perception. Then $\boldsymbol{x}^*$ maximizes the first player's worst-case payoff against all possible responses of the second player.

It may not be immediately clear that the equilibria defined above exist for all games. The payoff bounds derived by reasoning about the possible true payoffs given an abstraction produced via a payoff perception function are continuous and concave in each player's strategy (Theorem 2.8). The correspondence (a correspondence is a set-valued function) from other players' response functions to the set of a player's best response functions is therefore upper hemicontinuous and convex-valued. By considering an infinite dimensional space containing the Cartesian product of all players' response function spaces, Theorem 4.5 proves the existence of the equilibrium by applying the generalized Kakutani fixed-point theorem [Fan52].

Because the domain of $\boldsymbol{Y}(\cdot)$ can be exponentially large or even infinite, finding an equilibrium can be computationally intractable or even undecidable; the hardness still holds even if best response strategies are efficiently computable (Theorem 4.6). Despite this hardness, I present a positive result that for any $\epsilon > 0$, there exists an ϵ-equilibrium with a compact, tractable representation whose size is independent of $|\mathrm{dom}\, \boldsymbol{Y}|$ (Theorem 4.7). One consequence is that even if $\boldsymbol{Y}(\cdot)$ is not computable by any PSPACE algorithm, a trusted oracle can still communicate an efficient representation of an ϵ-equilibrium $(\boldsymbol{x}', \boldsymbol{Y}')$ to Turing-complete players such that $\boldsymbol{Y}'(\cdot)$ is efficiently computable.

Extended settings Section 4.2.4 briefly discusses a few extended settings that can be of interest for future research. These settings include:

- **Privately solvable games** When generalized to more than two players, the equilibrium defined above can not be solved from private information of lower-capability players (Remark 4.5), although all players can be convinced of an equilibrium if each player publicly announces their commitment to the equilibrium in the order of higher to lower capability level. Section 4.2.4.1 discusses an alternative equilibrium formulation that can be solved from private information of all players.

- **Stackelberg equilibrium** Section 4.2.4.2 discusses Stackelberg equilibria in two-player fully-capability-aware games. At a Stackelberg equilibrium, one player (called the leader) commits to a strategy first, and the other player (called the follower) observes the commitment and then chooses their strategy. Compared to conventional

normal-form games where a Stackelberg equilibrium exists if the follower breaks ties in favor of the leader [CS06], in a general-sum fully-capability-aware game, the leader's Stackelberg objective function may be noncontinuous (or only lower semicontinuous); the existence of an exact Stackelberg equilibrium is not guaranteed. Characterizing when an exact equilibrium exists is left as an open question. However, an approximate Stackelberg equilibrium, which is arguably sufficient for practical purposes, always exists. Moreover, exact Stackelberg equilibria exist in zero-sum fully-capability-aware games if the follower breaks ties in favor of the leader (Theorem 4.9).

- **Partially-capability-aware games** Section 4.2.4.3 presents an equilibrium formulation when there are players who only know their own capability levels but not the capability levels of other players. It demonstrates one way of generalizing the ideas in this thesis to work with more game models.

- **Games with probabilistic prior** Bayesian games are an influential model of incomplete information games where players share a common prior over the distribution of payoff functions [Har67]. By comparison, this thesis focuses on games without any probabilistic knowledge. Section 4.2.4.4 discusses how to extend the limited-perception model to incorporate games with a shared common prior on the distribution of true payoff functions. Future work may build on this extension to combine results from Bayesian games and limited-perception games.

Zero-sum games Section 4.3 studies two-player zero-sum fully-capability-aware limited-perception games. In sharp contrast to zero-sum normal-form games, which are efficiently solvable due to the minimax theorem [Nis+07, Section 1.4.2], finding an equilibrium in a zero-sum limited-perception game is as computationally hard as in the general-sum case since the latter can be reduced to the former (Theorem 4.11). Nevertheless, I generalize the minimax condition to identify efficiently solvable classes of two-player zero-sum games (Section 4.3.3), where finding an equilibrium becomes a nonsmooth convex optimization problem (Theorem 4.15) that is tractable in many cases [Bub+15]. The efficiently solvable classes are:

- **Constant-gap games** (Section 4.3.3.3), where the gap between the lower and upper bounds of true payoffs only depends on the capability level.

- **Strategy-separable linear component games** (Section 4.3.3.4), where the payoff perception function is defined by keeping the most significant linear components of the true payoff function given a predefined set of basis, and the number of pure strategies coupled with each pure strategy for unused basis in a perception is bounded.

- **Narrowly-reversible games** (Section 4.3.3.5), where the lower bound of the true payoff can be approached by upper bounds of true payoffs of a sequence of games with a higher capability level.

Such efficiently solvable games include the examples of limited-perception games mentioned earlier: limited-rank games, a subclass of the masked-payoff game, and quantized-payoff games.

Capability transfer properties Section 4.4 studies capability transfer properties of fully-capability-aware games. A *pessimistically capability-positive* game guarantees that, if a player increases their capability level, they will obtain a better payoff lower bound. A *programmatically capability-positive* game guarantees that, if a player increases their capability level, they will obtain a better true payoff. Theorem 4.25 shows that, with Stackelberg equilibrium, all two-player fully-capability-aware games are pessimistically capability-positive for both players. Theorem 4.27 presents a sufficient condition for a game to be both pessimistically and programmatically capability-positive with both Nash and Stackelberg equilibria for a given player. One interesting phenomenon is that a player may obtain a worst payoff by increasing their capability, even for the higher-capability player in a zero-sum game (Example 4.6).

1.2.5 Additional capability transfer properties

Sections 3.4 and 4.4 study a few capability transfer properties in limited-perception games via theoretical analysis. Since there is no known closed-form solution for the low-rank approximation of a general matrix, and even deciding whether a limited-rank game is capability-positive for a capability-oblivious player is computationally hard (Theorem 3.12), this situation motivates numerical experiments to investigate the capability transfer properties in large limited-rank games.

Chapter 5 presents numerical experiment results on capability transfer properties in limited-rank games, under both the capability-oblivious and fully-capability-aware settings with both zero-sum or general-sum games. I consider two classes of games: (i) random games with payoff values drawn from a normal distribution; and (ii) Kuhn poker [Kuh50], which is a zero-sum game with a predefined payoff matrix. For the random game, one interesting observation is that for capability-oblivious players, having a higher capability gives a better average payoff in a totally competitive setting (i.e., zero-sum games). By contrast, in general-sum games with independent payoff matrices for both players, a player should match their opponent's capability level to get the best average payoff; both players need to increase and match their capability levels to get better average payoffs. For Kuhn poker, increasing a player's capability level sometimes leads to a better payoff and sometimes leads to a worse payoff. Section 5.3 presents a more detailed summary of the numerical experiment observations.

1.3 Nonsmooth convex optimization

As introduced in Section 1.2, solving the maximin strategy for a capability-aware player in a zero-sum game reduces to solving a nonsmooth convex optimization problem. Nonsmooth convex optimization is a fundamental problem with applications in various fields besides game theory, including machine learning [Tib96], signal processing [Cha04], control theory [Cla+97], and operations research [Ber98].

I work with the general formulation $\min_{\boldsymbol{x}\in C} f(\boldsymbol{x})$ where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a convex function, and the constraint set $C \subset \mathbb{R}^n$ is compact and convex. The function $f(\cdot)$ is Lipschitz continuous over C but may be nondifferentiable on certain parts of C. There is a long history of research on nonsmooth convex optimization problems [Lem78; HL96b; BKM14;

Bag+20]. Most existing methods assume an oracle that delivers an arbitrary subgradient of $f(\cdot)$ at any given point. Such methods iteratively update the solution based on the current and previous subgradients. However, an arbitrary subgradient does not contain sufficient information to characterize the local behavior of the objective function. Since the function is nonsmooth, the subgradient can change abruptly in a neighborhood; following the (opposite of) subgradient direction may even increase the objective value. Consequently, any methods that query a subgradient oracle once per iteration need at least $\Omega(\epsilon^{-2})$ iterations to find an ϵ-approximate solution in the worst case [NY83; IN14].

Chapter 7 proposes and analyzes a new algorithm, the *Trust Region Adversarial Functional Subdifferential* (TRAFS), for unstructured nonsmooth convex optimization with $\mathcal{O}(\epsilon^{-1})$ iteration complexity, which improves on the previously best-known bound of $\mathcal{O}(\epsilon^{-2})$ in the same setting. The research on TRAFS is motivated by the optimization problems arising from solving certain limited-perception games. Intriguingly, the TRAFS algorithm embodies an approach that echoes the adversarial thinking in game theory.

TRAFS builds on two key ideas to achieve faster convergence. First, I abandon the subgradient oracle model and instead work with the *functional subdifferential* (Definition 7.4), a set of subgradients that capture sufficient local information to enable fast convergence while being easy to compute for a wide range of functions (Remark 7.3). Second, I adopt the adversarial thinking from game theory to find a descent direction in each iteration by solving a two-player zero-sum game. One player represents the optimizer with access to an ℓ_2-bounded trust region around the current iterate. The other player represents the objective function with access to its functional subdifferential at the current iterate. By using the equilibrium of this game as the iterate update, TRAFS finds an ϵ-approximate solution in $\mathcal{O}(\epsilon^{-1})$ or $\mathcal{O}(\epsilon^{-0.5})$ iterations depending on whether the objective function is strongly convex (Theorems 7.19 and 7.22), compared to the previously best-known bounds of $\mathcal{O}(\epsilon^{-2})$ and $\mathcal{O}(\epsilon^{-1})$ in these settings. Because the functional subdifferential contains multiple subgradients, we escape the corresponding lower bounds of $\Omega(\epsilon^{-2})$ and $\Omega(\epsilon^{-1})$ for optimization algorithms that use a single subgradient per iteration [NY83; IN14].

When the functional subdifferential satisfies the locally quadratic property (Definition 7.6), which intuitively means that it guarantees bounded function changes in a quadratically larger (compared to the standard functional subdifferential) neighborhood, TRAFS makes faster iterate progress (Theorem 7.25). As a corollary, TRAFS achieves linear convergence (i.e., $\mathcal{O}(\log \epsilon^{-1})$) for strongly convex smooth functions (Corollary 7.26), which recovers the lower bound of first-order methods for this class of functions [Bub+15].

The above complexity results of TRAFS assume the ability to solve a minimax problem involving the functional subdifferential in each iteration. Although the functional subdifferential can be defined mathematically for all Lipschitz convex functions (Propositions 7.6 and 7.7), the minimax problem with universally-defined functional subdifferentials is typically intractable. Fortunately, a function usually has more than one functional subdifferential, with some easier to work with than others. Section 7.1.2 presents compositional rules to compute the functional subdifferential for functions that use common operators (such as sum, max reduction, linear transform, etc.). Such rules enable efficient implementation of TRAFS for a wide range of functions, including nonsmooth convex benchmark problems used in previous work [HMM04; BKM14; NS15; Kar20] and various hard-case functions constructed in the proofs of the aforementioned lower bounds of iteration complexities [Bub+15; IN14;

Aga+09]. In other cases, a computation-efficient functional subdifferential may be difficult to obtain for certain functions. To this end, one can use the *almost-functional subdifferential* with weaker properties (Definition 7.7) in the TRAFS algorithm, which still guarantees convergence but results in worse upper bounds on the iteration complexities. With the almost-functional subdifferential, TRAFS is as fast as previous general-purpose nonsmooth convex optimization algorithms, with iteration complexities of $\mathcal{O}(\epsilon^{-2})$ and $\mathcal{O}(\epsilon^{-1})$ in the Lipschitz and strongly convex Lipschitz settings, respectively (Theorem 7.27).

TRAFS' complexity results also assume the knowledge of Lipschitz and convexity parameters. These parameters are often unknown and difficult to estimate for a given objective function. Section 7.2.5 presents an adaptive TRAFS implementation using heuristics inspired by the convergence analysis, which requires no knowledge of the convexity or smoothness parameters. Section 7.3 evaluates the performance of the adaptive TRAFS implementation numerically. The benchmark set includes problems used in previous research on nonsmooth convex optimization [HMM04; BKM14; NS15; Kar20], ℓ_1-regularized linear models in machine learning, and a new set of problems inspired by the maximin strategy objective in limited-rank games. TRAFS solves twice as many problems compared to the second-best method and is on average 39.1 times faster on the cases solved by both methods (Section 7.3.4).

1.4 Limited-strategy games

Chapter 8 studies *limited-strategy games* as a second setting of limited-capability games where a player's capability limits the complexity of their strategy. In a limited-strategy game, a higher-capability player has access to a larger strategy space. Section 8.1 presents the definition of general limited-strategy games and defines capability-aware players and players with limited capability information. Unlike limited-perception games, there is no truly capability-oblivious player in a limited-strategy game since a player must know their own capability level to choose a valid strategy. Players who are uncertain about others' capability levels are modeled under the Bayesian setting in Definition 8.5.

There can be a variety of mechanisms to formalize a hierarchy of strategy spaces in a limited-strategy game. Different formalizations may need different analysis techniques and may lead to different results. I propose representing a player's strategy space using programs in a Domain-Specific Language (DSL) and quantifying player capabilities based on size-bounded programs in the DSL. A DSL is a formal language that is designed for a specific application domain. Like general-purpose programming languages, DSLs have syntax and semantics, but they are often more tailored to the specific domain without necessarily being Turing-complete. Many properties related to programs in Turing-complete languages are undecidable, including whether a program takes finite time to finish, whether two programs are equivalent, whether a line of code is reachable, generating the shortest program that produces a given output, etc. [Sip12; Aho+13]. By contrast, the synthesis and analysis of a DSL program can be more tractable [WB23; RSH19; Han+21].

I study *distance-bounded network congestion games with default action* (DncDa) as a prominent model of limited-strategy games. DncDa was first introduced in the author's previous work with Yichen Yang [YJR22]. In a DncDa game, each player chooses a path

from a source to a destination on a network. The players experience delays on their paths depending on how many other players use the overlapping edges, which are described by *delay functions*. Each vertex except the source has a default outgoing edge. A player's strategy is represented by a program in a DSL that specifies their non-default choices and falls back to the default action for the rest of the path. The player's capability level bounds the maximum size of programs they can use, which corresponds to the maximum number of non-default edges they can choose. The game is symmetric-capability in that all players have the same capability level. The game is thus also symmetric.

One result is that solving DncDa games is PLS-complete [YJR22], in contrast to ordinary symmetric network congestion games whose solutions are in P [FPT04]. This finding is consistent with the argument of Papadimitriou [Pap92] that limiting the implementation complexity of strategies tends to increase the design complexity of solving a game, although their model uses limited-size automata in the repeated prisoner's dilemma game. Another result is if we want to ensure a socially capability-monotonic game (i.e., games where social welfare increases/decreases with capability level) for all possible network structures by choosing a shared delay function for all edges, then there are no non-trivial delay functions other than constant functions [YJR22].

Section 8.2.2 defines the McDncDa model, which extends the DncDa model by allowing players to have different capability levels. I define *capability-positive* McDncDa games as games where a player never gets a longer delay if they increase their capability level by one starting with a given capability profile. I show that characterizing capability-related properties in McDncDa games is computationally hard (Theorems 8.3 to 8.7), which implies that there are games that are capability-positive and games that are not.

Finally, I study a special class of McDncDa called *Mixed Gold and Mines Games* (MGMG) where the network has a parameterized structure in Section 8.2.3. In an MGMG, two players compete to collect gold and avoid mines. Gold and mine sites are placed on two parallel horizontal lines. Players can only move rightward or switch lines. Switching the line once costs a player one unit of capability. The return on a gold site depends on how many players cover the same site. The penalty on a mine site is constant. I present closed-form expressions for possible player payoffs at pure Nash equilibria. Although the game can have multiple Nash equilibria, I prove that MGMG is universally capability-positive (i.e., capability-positive for all capability profiles; see Corollary 8.9). Another interesting result is that if the gold sites have increasing returns to scale (i.e., the sum of returns of two players is more than the return of a single player), then the game is socially capability-positive. Of note, even in the two-player case, not all McDncDa games are universally capability-positive; Example 8.2 presents a counterexample.

1.5 Summarization of concepts and results

This thesis introduces a range of new concepts to game theory. This section summarizes the key new concepts and results.

- *Limited-perception games* (Definitions 2.1 and 2.3) are games in which each player perceives approximations of all players' true payoff functions depending on this player's capability level. A player's perception is modeled by a *payoff perception function*.

  - Payoff perception functions satisfy three axioms (Definition 2.2). They can be equivalently defined by hierarchical partitions of the payoff functions (Theorems 2.3 and 2.4).
  - The bounds of expected true payoffs given a perception and a mixed strategy profile are monotonic with respect to the capability level (Theorem 2.6) and continuous and concave/convex with respect to a player's mixed strategy (Theorem 2.8).

- *Capability-oblivious players* play the optimal strategy of their perceived games (Definition 3.1). Properties related to such players depend on the specific game setting. I summarize results in limited-rank games later in this list.

- All players in a *fully-capability-aware game* know the capability levels of all players as common knowledge; each player maximizes their worst-case expected payoff (Definition 4.3).

  - Section 4.2.2 presents the equilibrium formulation for simultaneous-move fully-capability-aware games, which differs from conventional Nash equilibria by using a best-response function (which can be understood as a higher-order function that maps a perception into a mixed strategy) instead of a mixed strategy for the higher-capability player. Theorem 4.5 proves the existence of the new equilibrium concept.
  - Fully-capability-aware games are hard to solve in general (Theorem 4.6), but approximate equilibria can be compactly represented (Theorem 4.7).
  - The two-player zero-sum case is not easier to solve than the general-sum case (Theorem 4.11), but there are efficiently solvable zero-sum game classes that reduce to solving nonsmooth convex optimization problems (Section 4.3.3).

- *Limited-rank games* are two-player games where each player perceives low-rank approximations of the true payoff matrices. They are a special class of limited-perception games (Section 3.1).

  - Theorem 3.3 presents analytical expressions for lower and upper true payoff bounds. Solving zero-sum fully-capability-aware limited-rank games reduces to solving second-order cone programs (Corollary 3.4).
  - Capability-oblivious players obtain a guaranteed optimality gap in the zero-sum case (Theorem 3.10), recover the pure Nash equilibria of the true game in certain general-sum cases (Theorem 3.8), and converge to an approximate equilibrium with fictitious play when one of them is rank-1 (Theorem 3.11).

- *Limited-strategy games* are games in which a player's strategy space is restricted according to their capability level (Definition 8.1).

- *Mixed-capability distance-bounded network congestion games with default action (McDNCDA)* are a class of limited-strategy games. They generalize network congestion games by defining default outgoing edges for each vertex. The number of a player's non-default edge choices in a McDNCDA is bounded by their capability level (Definition 8.10).

  - Computing an equilibrium in McDncDa is PLS-complete in general, but in P if the number of players is treated as a constant (Theorem 8.2).
  - It is NP-hard to find the best or worst equilibrium for a given single player in McDncDa (Corollary 8.6).

- *Mixed gold and mine games (MGMG)* are a special class of two-player McDncDa where the network layout and delay functions are parameterized. Players compete to collect gold and avoid mines on the network (Section 8.2.3). Theorem 8.8 analytically characterizes players' payoffs at pure Nash equilibria in an MGMG.

- *Capability transfer properties* describe how the outcome of a game changes with varying player capability levels.
  - For capability-oblivious players in limited-rank games, it is PPAD-hard to determine if a player gets strictly better payoffs when they are full-rank compared to when they are rank-1 (Theorem 3.12). Section 3.4.2 analyzes a parameterized 2×2 defense game to reveal the possible changes of both players' payoffs and social welfare when one player is rank-1 and the other changes from rank-1 to rank-2; it shows that player payoffs or social welfare can either increase or decrease as the rank-1 player's capability level increases.
  - For fully-capability-aware limited-perception games, Section 4.4 defines and analyzes *pessimistically* and *pragmatically* capability-positive games, where a player gets a non-decreasing payoff lower bound or a non-decreasing true payoff when they increase their capability level, respectively. Two-player games are pessimistically capability-positive with Stackelberg equilibria (Theorem 4.25); if a game is maximin-attainable, then it is also pessimistically capability-positive with Nash equilibria (Corollary 4.26). Theorem 4.27 presents a sufficient condition for a game to be both pessimistically and pragmatically capability-positive.
  - For limited-rank games, Chapter 5 presents numerical experiment results on capability transfer properties under both the capability-oblivious and fully-capability-aware settings with both zero-sum or general-sum games. Section 5.3 summarizes the empirical discoveries. One interesting finding is that for capability-oblivious players in random games, if the game is general-sum, then a player should match their opponent's capability level to maximize their average payoff; if the game is zero-sum, then a player obtains better payoffs with a higher capability level.
  - For McDncDa, it is coNP-complete to determine if a player gets no worse payoff when they increase their capability level by one (Theorem 8.7).
  - For MGMG, a player gets no worse payoff when they increase their capability level by one; if the gold sites have increasing returns to scale, then the social welfare is non-decreasing when both players increase their capability levels by one (Corollary 8.9).

- *Functional subdifferential* is a set of subgradients of a convex continuous function that satisfy certain properties (Definition 7.4). With functional subdifferential, the TRAFS

algorithm achieves faster convergence rates than previous methods for solving nonsmooth convex optimization problems (Section 7.2). Of note, nonsmooth convex optimization problems arise in solving certain two-player fully-capability-aware zero-sum limited-perception games (Section 4.3.3).

1.6 Technical background

This section provides an overview of the background required to understand the technical details of this thesis. I will describe the notation used in this thesis and briefly introduce key concept in game theory.

1.6.1 Notation

The following notation is used throughout this thesis:

- $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{N}$ denote the set of real numbers, integers, and natural numbers, respectively. Derived sets include $\mathbb{R}_+$ for positive real numbers, $\mathbb{R}_{\geq 0}$ for non-negative real numbers, $\mathbb{Z}_+$ for positive integers, and $\overline{\mathbb{Z}}_+ \coloneqq \mathbb{Z}_+ \cup \{\infty\}$ for extended positive integers.
- P, NP, coNP, PPAD, and PSPACE denote the complexity classes polynomial time, nondeterministic polynomial time, complement of NP, Polynomial Parity Arguments on Directed graphs, and Polynomial Space, respectively.
- Given $a, b \in \mathbb{R} \cup \{\pm\infty\}$, $[\![a, b]\!] \coloneqq \{x \in \mathbb{Z} \cup \{\pm\infty\} \mid a \leq x \leq b\}$ is the set of extended integers between a and b.
- Bold symbols denote vectors (lower case) and matrices (upper case). For example, $\boldsymbol{a}$ is a vector, $\boldsymbol{a}_i$ is a vector indexed by i in a family of vectors, a_i is the i-th element of the vector $\boldsymbol{a}$, $\boldsymbol{A}$ is a matrix, $\mathbf{1}$ is the all-one vector whose dimension is inferred from context, and $\mathbf{1}_n$ is the all-one vector of dimension n.
- For a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{A}_i \in \mathbb{R}^{1 \times n}$ is the i-th row of $\boldsymbol{A}$ (indexed from 1), $\boldsymbol{A}_{:,j} \in \mathbb{R}^{m \times 1}$ is the j-th column of $\boldsymbol{A}$, and $\boldsymbol{A}_{i:j,\,k:l}$ is the submatrix of $\boldsymbol{A}$ consisting of rows i to j and columns k to l (all endpoints inclusive).
- $\boldsymbol{e}_i$ denotes the i-th standard basis vector in $\mathbb{R}^n$ (i.e., the vector whose i-th element is 1 with all other elements being 0), where $i \in [\![1, n]\!]$ and n is inferred from context.
- Δ_n denotes the n-dimensional simplex, defined as $\Delta_n \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n \mid \mathbf{1}^\intercal \boldsymbol{x} = 1, \boldsymbol{x} \geq \mathbf{0}\}$. $k\Delta_n$ denotes the n-dimensional simplex scaled by $k \in \mathbb{R}_+$. When s is a finite set, Δ_s denotes the simplex over the elements of s, which can be interpreted as a probability distribution over s.
- For $\boldsymbol{x} \in \Delta_n$, $\mathrm{supp}^{\mathbf{v}}(\boldsymbol{x})$ is the vector representation of the support of $\boldsymbol{x}$ such that $\mathrm{supp}^{\mathbf{v}}(\boldsymbol{x}) = \boldsymbol{y}$ where $\boldsymbol{y} \in \{0, 1\}^n$ and $y_i = 1$ if and only if $x_i > 0$; $\mathrm{supp}(\boldsymbol{x}) \coloneqq \{i \in [\![1, n]\!] \mid x_i > 0\}$ denotes the set of indices of the support of $\boldsymbol{x}$.

- For a function $f : X \mapsto \mathbb{R}$, $\operatorname{argmin}_{x \in X} f(x) \coloneqq \{x \in X \mid \forall y \in X : \ f(x) \leq f(y)\}$ denotes the set of minimizers of f over X; $\operatorname{argmin}^{*}_{x \in X} f(x)$ denotes an arbitrary element in $\operatorname{argmin}_{x \in X} f(x)$. Similarly, argmax and $\operatorname{argmax}^{*}$ are defined for the maximizers. Of note, argmin and argmax can be empty sets.

- For a set A, $\operatorname{cl} A$ denotes the closure of A (i.e., A plus all limit points of A), and $\operatorname{conv} A$ denotes the convex hull of A (i.e., the smallest convex set containing A). Of note, a set A is convex if and only if for x, $y \in A$ and $\alpha \in [0, 1]$, $\alpha x + (1 - \alpha) y \in A$.

- For $\boldsymbol{x} \in \mathbb{R}^n$ and $r \in \mathbb{R}_{\geq 0}$, $B_r[\boldsymbol{x}] \coloneqq \{\boldsymbol{y} \in \mathbb{R}^n \mid \|\boldsymbol{x} - \boldsymbol{y}\|_2 \leq r\}$ is the closed ball centered at $\boldsymbol{x}$ with radius r.

- Definitions and examples are typeset in normal font and end with a left triangle ($\triangleleft$). Proofs are typeset in normal font and end with a square ($\square$). Theorems, lemmas, corollaries, and propositions are typeset in *italic* font without a terminating symbol.

- Given two sets X and Y, $f : X \rightrightarrows Y$ denotes a *correspondence* from X to Y, i.e., a set-valued function $f : X \mapsto 2^Y$.

1.6.2 Games and rational behavior

This thesis mainly works with finite one-shot normal-form non-cooperative games.

Definition 1.1 (Normal-form game)
An n-player normal-form game is a tuple $G = \left(\mathcal{N}, (\mathcal{S}_i)_{i \in \mathcal{N}}, (u_i)_{i \in \mathcal{N}}\right)$, where

- $\mathcal{N} \coloneqq [\![1, n]\!]$ is the set of players
- $\mathcal{S}_i$ is the finite set of strategies available to player i
- $u_i : \mathcal{S} \mapsto \mathbb{R}$ is the utility function of player i, where $\mathcal{S} \coloneqq \prod_{i \in \mathcal{N}} \mathcal{S}_i$ is the set of strategy profiles.

When there are two players, the game is also called a *bimatrix game* since the payoff functions u_1 and u_2 can be represented as matrices. $\triangleleft$

In a one-shot game, players simultaneously choose strategies; each player chooses a strategy from their own strategy set. Let s_i denote the strategy chosen by player i. The vector $\boldsymbol{s} \coloneqq (s_1, \ldots, s_n)$ is the *strategy profile*. The payoff of player i given strategy profile $\boldsymbol{s}$ is $u_i(\boldsymbol{s})$. Each player aims to maximize their own payoff.

Nash [Nas51] introduced *Nash equilibrium* as a solution concept for non-cooperative games. A strategy profile $\boldsymbol{s}^*$ is a Nash equilibrium if no player has an incentive to unilaterally deviate from $\boldsymbol{s}^*$. For a strategy profile $\boldsymbol{s}$, let $\boldsymbol{s}_{-i}$ denote the strategies of all players except player i. A strategy profile $\boldsymbol{s}^*$ is a Nash equilibrium if and only if for all $i \in \mathcal{N}$, $s_i^* \in \operatorname{BR}_i\left(\boldsymbol{s}^*_{-i}\right)$ where $\operatorname{BR}_i\left(\boldsymbol{s}^*_{-i}\right) \coloneqq \operatorname{argmax}_{s_i' \in \mathcal{S}_i} u_i\left(s_i', \boldsymbol{s}^*_{-i}\right)$ is called the *best response* of player i to $\boldsymbol{s}^*_{-i}$. If a rational player knows that other players commit to a Nash equilibrium, then their best strategy is to also commit to the Nash equilibrium.

The Nash equilibrium defined above is typically referred to as a *pure* Nash equilibrium. Many games do not have any pure Nash equilibria. One way to address this issue is to consider *mixed* strategies, where players choose strategies according to probability distributions. A mixed strategy for player i is represented by $\sigma_i \in \Delta_{\mathcal{S}_i}$. The expected payoff of player i given a mixed strategy profile $\boldsymbol{\sigma}$ is $\mathbb{E}_{\boldsymbol{s}\sim\boldsymbol{\sigma}}[u_i(\boldsymbol{s})] = \sum_{\boldsymbol{s}\in\mathcal{S}} \prod_{j\in\mathcal{N}} \sigma_j(s_j) u_i(\boldsymbol{s})$. A mixed strategy profile $\boldsymbol{\sigma}^*$ is a Nash equilibrium if for all $i \in \mathcal{N}$, $\sigma_i^* \in \operatorname{argmax}_{\sigma_i'\in\Delta_{\mathcal{S}_i}} \mathbb{E}_{s_i'\sim\sigma_i', \boldsymbol{s}_{-i}\sim\boldsymbol{\sigma}^*_{-i}}[u_i(s_i', \boldsymbol{s}_{-i})]$. All finite games have at least one mixed Nash equilibrium [Nas51]. In this thesis, unless otherwise specified, a Nash equilibrium refers to a mixed Nash equilibrium.

It is computationally challenging to find a Nash equilibrium. Chen, Deng, and Teng [CDT09] proves that solving a Nash equilibrium in a bimatrix game is PPAD-complete, even for the win-lose bimatrix game where all payoffs are either 0 or 1. One may also consider the ϵ-*approximate Nash equilibrium* (ϵ-Nash for short) where no player can improve their payoff by more than ϵ by unilaterally deviating from the equilibrium. Chen, Deng, and Teng [CDT09] shows that there is no polynomial-time algorithm to compute an ϵ-approximate Nash equilibrium in a bimatrix game unless $\mathsf{PPAD} \subseteq \mathsf{P}$. The complexity class PPAD stands for *Polynomial Parity Arguments on Directed graphs*, which was first introduced by Papadimitriou [Pap94]. An example of a PPAD-complete problem is finding a Brouwer fixed point [Pap94]. Given a function $f : [0, 1]^n \mapsto [0, 1]^n$, Brouwer's fixed-point theorem implies that there is $\boldsymbol{x}^* \in [0, 1]^n$ such that $f(\boldsymbol{x}^*) = \boldsymbol{x}^*$ [Sub18, Chapter 10]. Finding such $\boldsymbol{x}^*$ for an approximate Turing-machine implementation of $f(\cdot)$ is a PPAD-complete problem. Hirsch, Papadimitriou, and Vavasis [HPV89] proves an exponential lower bound on the number of function evaluations required to find a Brouwer fixed point assuming black-box access to $f(\cdot)$ even in two dimensions.

For two-player zero-sum games, defined as games where the sum of payoffs of all players is zero (or any fixed constant) for every strategy profile, the *minimax theorem* [Sim95] implies that $(\boldsymbol{x}_1^*, \boldsymbol{x}_2^*)$ is a Nash equilibrium where $\boldsymbol{x}_i^*$ is the mixed strategy of player i that maximizes their worst-case payoff when the other player can use any mixed strategy. Consequently, a Nash equilibrium can be found by solving a pair of linear programs [MZS13, Section 5.2.5]. Of note, a linear program with rational coefficients can be solved exactly in polynomial time [Kar84].

Interested readers can refer to Maschler, Zamir, and Solan [MZS13], Roughgarden [Rou16], Nisan et al. [Nis+07] and references therein for more comprehensive treatments of game theory and its algorithmic aspects.

Chapter 2

The limited-perception game model

This chapter formalizes the limited-perception game model to provide the basis for the subsequent chapters. In a limited-perception game, each player is presented with an *abstraction* of the true payoff functions. The true payoff functions are unknown to the players but determine the final outcome of the game. A player's capability level affects the accuracy of their perception of the payoff functions, with a higher capability level corresponding to a more accurate perception. The perception process is governed by a *payoff perception function* that maps a payoff function and an integer capability level to an abstraction of the payoff function. Sections 2.1 and 2.2 define the payoff perception function and *one-shot limited-perception games*; these two sections also present properties that hold in general limited-perception games. Section 2.3 provides an alternative understanding of the payoff perception function by presenting an equivalent definition based on a hierarchy of partitions over the set of payoff functions. Finally, Section 2.4 defines the bounds of possible true payoffs given a perceived payoff function and a mixed strategy profile and proves some properties of the bounds. Of note, although most of this thesis assumes no probabilistic prior on the distribution of the true payoff functions, limited-perception games can be extended to incorporate such priors (Section 4.2.4.4).

2.1 The payoff perception function

Definition 2.1 (One-shot limited-perception game)
A finite *one-shot limited-perception game* (*OLP-Game* for short) with n players is a tuple $G = (\mathcal{N}, (\mathcal{S}_i)_{i\in\mathcal{N}}, (u_i)_{i\in\mathcal{N}}, \alpha)$ where:

- $\mathcal{N} = [\![1, n]\!]$ is the set of players ($[\![a, b]\!]$ denotes integers from a to b inclusive).
- $\mathcal{S}_i$ is a finite set of pure strategies available to i.

- $u_i : \mathcal{S}_1 \times \cdots \times \mathcal{S}_n \to \mathbb{R}$ is the *true payoff function* of player i.
- α is the *payoff perception function* of the game as defined in Definition 2.2.

For a given game G, there can be multiple *instances*. In each instance, each player has a capability level that determines their perception of the payoff functions. Definition 2.3 specifies the game play. ◁

Definition 2.2 (Payoff perception function)
In an OLP-Game defined in Definition 2.1, the *payoff perception function* is a function $\alpha : \left(U \times \overline{\mathbb{Z}}_+\right) \mapsto U$ where $U \coloneqq \mathbb{R}^{\mathcal{S}_1 \times \cdots \times \mathcal{S}_n}$ is the set of payoff functions, and $\overline{\mathbb{Z}}_+ \coloneqq \mathbb{Z}_+ \cup \{\infty\}$ is the extended set of positive integers. If $v = \alpha(u,\, c)$, then v is is called an *abstraction* of u at *capability level* c. The payoff perception function satisfies the following properties:

- *Capability path independence*: for any $u \in U$ and $c_1, c_2 \in \mathbb{Z}_+$, it holds that $\alpha(\alpha(u,\, c_1),\, c_2) = \alpha(u,\, \min\{c_1,\, c_2\})$.
- *Perfect perception with infinite capability*: for any $u \in U$, $\alpha(u,\, \infty) = u$.
- *Bounded concretization*: given any $v \in U$, there exists a constant $b \in \mathbb{R}_+$ such that for any $u \in U$ and $c \in \mathbb{Z}_+$ satisfying $\alpha(u,\, c) = v$, it holds that $\max_{s \in \mathcal{S}_1 \times \cdots \times \mathcal{S}_n} |u(s)| \le b$.

Let $\bar{c}(u) \coloneqq \min\left\{c \in \overline{\mathbb{Z}}_+ \;\middle|\; \alpha(u,\, c) = u\right\}$. The second property guarantees that $\bar{c}(u)$ is well-defined. The value $\bar{c}(u)$ is called the *intrinsic capability* of the payoff function $u(\cdot)$ given the payoff perception function $\alpha(\cdot,\, \cdot)$. ◁

Below are several examples of games with different payoff perception functions.

Example 2.1 (Masked payoff game)
Consider a function $\alpha_{\mathrm{s}} : \mathbb{R}^{m \times n} \times \overline{\mathbb{Z}}_+ \mapsto \mathbb{R}^{m \times n}$ where $\alpha_{\mathrm{s}}(\boldsymbol{M},\, c)$ keeps the c entries with the largest absolute values in $\boldsymbol{M}$ and sets the rest to zero. Tie-breaking can be arbitrary but must be consistent with capability path independence (e.g., by preferring to keep the smallest row/column indices). Then $\alpha_{\mathrm{s}}(\cdot)$ is a payoff perception function for bimatrix games. ◁

Example 2.2 (Quantized payoff game)
Consider a function that keeps only c decimal digits for each payoff value by rounding towards zero. Formally, let $\alpha_{\mathrm{q}}(u,\, c) = u'$, then for any strategy profile $\boldsymbol{s} \in \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$,

$$u'(\boldsymbol{s}) \coloneqq \begin{cases} \operatorname{sign}\left(u(\boldsymbol{s})\right) \cdot 10^{-c} \cdot \lfloor 10^c \cdot |u(\boldsymbol{s})| \rfloor & \text{if } c < \infty \\ u(\boldsymbol{s}) & \text{if } c = \infty \end{cases}$$

It is easy to verify that $\alpha_{\mathrm{q}}(\cdot,\, \cdot)$ is a payoff perception function. ◁

Example 2.3 (Limited-rank game)
A limited-rank game is a two-player OLP-Game. Let $(\boldsymbol{A},\, \boldsymbol{B})$ be the true payoff matrices. A player at capability level c perceives the game as $(\boldsymbol{A}',\, \boldsymbol{B}')$ where $\boldsymbol{A}'$ and $\boldsymbol{B}'$ are the best rank-c approximations (minimizing the Frobenius norm of the residual matrix) of $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively. ◁

Example 2.4 (Largest masked payoff game)
If we change Example 2.1 to keep the c entries with the smallest absolute values, the resulting function is not a payoff perception function since it violates the bounded concretization property. ◁

The properties in Definition 2.2 result in a structure compatible with an intuitive understanding of abstraction. The following theorem illustrates some interpretations of the structure induced by the payoff perception function.

Theorem 2.1 (Properties of the payoff perception function)
With the notation in Definition 2.2, let $u \in U$ be a payoff perception function. Then the following properties hold:

- Perfect perception beyond intrinsic capability*: For any $c \geq \bar{c}(u)$, we have $\alpha(u, c) = u$.*
- Perfect perception of an abstraction*:*
  For any $c \in \mathbb{Z}_+$, let $v = \alpha(u, c)$. Then $\alpha(v, c) = v$. Equivalently, $\bar{c}(v) \leq c$.
- First form of information loss*:*
  If $\alpha(u, c) \neq u$ for some $c \in \mathbb{Z}_+$, then $\alpha(u, c') \neq u$ for any $c' \in [\![1, c]\!]$.
- Second form of information loss*:*
  For any $c \in \mathbb{Z}_+$, let $v = \alpha(u, c)$. If $v \neq u$, then $\alpha(v, c') \neq u$ for any $c' \in \mathbb{Z}_+$.

Proof. To prove the first two properties, one can substitute u (resp. v) with $\alpha(u, \bar{c}(u))$ (resp. $\alpha(v, c)$) and apply the capability path independence property; i.e.,

$$\alpha(u, c) = \alpha(\alpha(u, \bar{c}(u)), c) = \alpha(u, \min\{\bar{c}(u), c\}) = \alpha(u, \bar{c}(u)) = u$$
$$\alpha(v, c) = \alpha(\alpha(u, c), c) = \alpha(u, \min\{c, c\}) = \alpha(u, c) = v$$

The first form of information loss is a direct consequence of the first property, since $\alpha(u, c') = u$ implies $\alpha(u, c) = u$ for $c \geq c'$.

To prove the second form of information loss, note that $\alpha(v, c') = \alpha(u, \min\{c, c'\})$. We have $\alpha(v, c') = \alpha(u, c) = v \neq u$ when $c' > c$, and $\alpha(v, c') = \alpha(u, c') \neq u$ when $c' \leq c$ (due to the first form of information loss). □

2.2 The game play

The following definition formally describes the game play of the OLP-Game.

Definition 2.3 (Game play of the OLP-Game)
In an instance of an OLP-Game as defined in Definition 2.1, each player i has a capability level $c_i \in [\![1, b_i]\!]$. The tuple $(c_i)_{i \in \mathcal{N}}$ is called the *capability profile* of this instance. Each player i perceives the payoff function of player j as $v_j^i := \alpha(u_j, c_i)$. Players also know $\mathcal{N}$ and $(\mathcal{S}_j)_{j \in \mathcal{N}}$ as common knowledge. Players do not know u_j. Depending on the setting, the players may or may not know $(c_j)_{j \in \mathcal{N}}$ and $\alpha(\cdot, \cdot)$. Player i chooses a strategy $s_i \in \mathcal{S}_i$ based on the perceived payoff functions $(v_j^i)_{j \in \mathcal{N}}$. After all players choose their strategies, the true game $(u_j)_{j \in \mathcal{N}}$ is revealed to all players. The payoff of player i is $u_i(s_1, \ldots, s_n)$.

In this game, the value of $v_j^i(s_1, \ldots, s_n)$ is called the *apparent payoff* of player j to player i and $u_i(s_1, \ldots, s_n)$ is the *true payoff* of player i. The game described by $(v_j^i)_{j\in\mathcal{N}}$ is the *perceived game* of player i. The game described by $(u_j)_{j\in\mathcal{N}}$ is the *true game*. ◁

Based on the perceived payoff function, a player can reason about the possible set of true payoff functions with the payoff concretization function defined below.

Definition 2.4 (Payoff concretization function)
With the notation in Definition 2.2, let $v \in U$ be a payoff function. The inverse of the payoff perception function is called a *payoff concretization function*, which is denoted as $\alpha^{-1} : \left(U \times \overline{\mathbb{Z}}_+\right) \mapsto 2^U$ and defined as

$$\alpha^{-1}(v, c_0) \coloneqq \{u \in U \mid \alpha(u, \max\{\bar{c}(v), c_0\}) = v\}$$ ◁

Note that in the above definition, the set of possible true payoff functions for a player with a known capability of c is $\alpha^{-1}(v, c)$; if their capability is unknown, then we can use a more conservative estimate by $\alpha^{-1}(v, 1)$.

Example 2.5 (Finite/infinite payoff concretization)
The value of a payoff concretization function can be a finite set or an infinite set. For the masked payoff game in Example 2.1, $\alpha_{\mathrm{s}}^{-1}(\boldsymbol{M}, c) = \{\boldsymbol{M}\}$ when $c \geq mn$. By comparison, for the quantized payoff game in Example 2.2, $\alpha_{\mathrm{r}}^{-1}(\boldsymbol{M}, c)$ is always an uncountable (thus infinite) set for $c < \infty$. ◁

The following theorem shows that the payoff perception function defines a hierarchy of abstractions of the true payoff function, thus rendering the OLP-Game an instance of the limited-perception game model informally defined at the beginning of this section.

Theorem 2.2 (Abstraction reduces perception accuracy)
With the notation in Definitions 2.2 and 2.4, for any $v \in U$ and $c \in \mathbb{Z}$, it holds that

$$\alpha^{-1}(v, c+1) \subseteq \alpha^{-1}(v, c) \tag{2.1}$$

Let $v' = \alpha(v, c)$, then for $c' \geq c$, it holds that

$$\alpha^{-1}(v, c') \subseteq \alpha^{-1}(v', c) \tag{2.2}$$

Proof. To prove (2.1), let $u \in \alpha^{-1}(v, c+1)$, which means $\alpha(u, \max\{\bar{c}(v), c+1\}) = v$. If $c \geq \bar{c}(v)$, then $\alpha(u, c+1) = v$. Applying the properties of capability path independence property and perfect perception beyond intrinsic capability yields $\alpha(u, c) = \alpha(\alpha(u, c+1), c) = \alpha(v, c) = v$. If $c < \bar{c}(v)$, then $\alpha(u, \max\{\bar{c}(v), c\}) = \alpha(u, \max\{\bar{c}(v), c+1\}) = v$. In both cases, we have $\alpha(u, \max\{\bar{c}(v), c\}) = v$, which means $u \in \alpha^{-1}(v, c)$.

To prove (2.2), first consider the case when $c \geq \bar{c}(v)$, which implies $v' = v$ due to the property of perfect perception beyond intrinsic capability. Then (2.2) follows from (2.1) by induction on c'. Now assume $c < \bar{c}(v)$. Let $u \in \alpha^{-1}(v, c')$, which means $\alpha(u, \max\{\bar{c}(v), c'\}) = v$. We also have $\bar{c}(v') \leq c$ due to the property of perfect perception of an abstraction. Then $v' = \alpha(v, c) = \alpha(\alpha(u, \max\{\bar{c}(v), c'\}), c) = \alpha(u, \min\{\max\{\bar{c}(v), c'\}, c\}) = \alpha(u, c)$. We thus have $\alpha(u, \max\{c, \bar{c}(v')\}) = \alpha(u, c) = v'$, which means $u \in \alpha^{-1}(v', c)$. □

An interpretation of (2.1) is that given the same perceived payoff function, if a player has a higher capability, then they can have a more precise set of possible true payoff functions. Eq. (2.2) can be understood as another form of information loss, as the set of possible true payoff functions becomes larger when an abstraction is applied to a perceived payoff function.

2.3 A hierarchical-partition perspective of the payoff perception function

An alternative view of the payoff perception function is to consider it as a hierarchy of partitions of the set of payoff functions. Given a capability level c, the payoff perception function defines equivalence classes that form a partition of U, where $u_1 \in U$ and $u_2 \in U$ belongs to the same class if and only if $\alpha(u_1, c) = \alpha(u_2, c)$. Each equivalence class becomes smaller as c gets larger. The following definition specifies the properties that such a hierarchy of partitions should satisfy.

Definition 2.5 (Hierarchical partitions of payoff functions)
Let $U \coloneqq \mathbb{R}^{\mathcal{S}}$ be the set of possible payoff functions over a finite strategy space $\mathcal{S} \coloneqq \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$. A *hierarchical partition* of U is a pair $\left((H_c)_{c \in \overline{\mathbb{Z}}_+}, r\right)$ where $H_c \subseteq 2^U$ defines a hierarchy of partitions of U indexed by the extended set of positive integers, and $r : 2^U \mapsto U$ is called the *representation function* that selects a representative element from an equivalence class. A hierarchical partition satisfies the following properties:

- *Partition at each level*: For any $c \in \overline{\mathbb{Z}}_+$, H_c is a partition of U.
- *Partition with refinement*: For any $c \in \mathbb{Z}_+$, for any $S \in H_c$, there exists $P \subseteq H_{c+1}$ such that P is a partition of S.
- *Singleton partition at infinite level*: $H_\infty = \{\{u\} \mid u \in U\}$.
- *Value-bounded partitions*: For any $c \in \mathbb{Z}_+$ and $S \in H_c$, there exists a constant $b \in \mathbb{R}_+$ such that for any $u \in S$, $\max_{x \in S} |u(x)| \leq b$.
- *Closed representation*: For any $c \in \overline{\mathbb{Z}}_+$ and $S \in H_c$, $r(S) \in S$.
- *Stable representation*: For $S \subseteq T$ where $S \in H_c$ and $T \in H_{c'}$ for some $c, c' \in \overline{\mathbb{Z}}_+$, if $r(T) \in S$, then $r(S) = r(T)$.

Of note, for any set S, $P \subseteq 2^S$ is a *partition* of S if and only if for any $t \in S$, there is exactly one element $q \in P$ such that $t \in q$. ◁

The following theorems show that the payoff perception function can be viewed as the hierarchical partition, and vice versa.

Theorem 2.3 (From payoff perception function to hierarchical partition)
Let $U \coloneqq \mathbb{R}^{\mathcal{S}}$ be the set of possible payoff functions over a finite strategy space $\mathcal{S} \coloneqq \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$. Let $\alpha : \left(U \times \overline{\mathbb{Z}}_+\right) \mapsto U$ be a payoff perception function. For $c \in \overline{\mathbb{Z}}_+$, define H_c as the set of equivalence classes of U under the relation $\sim_c$ where $u_1 \sim_c u_2$ if and only if $\alpha(u_1, c) = \alpha(u_2, c)$. Let $r : 2^U \mapsto U$ be any function such that if $S \in H_c$ for some $c \in \overline{\mathbb{Z}}_+$, then $r(S) = \alpha(u, c)$ for some $u \in S$. Then the function $r(\cdot)$ exists and (H_c, r) is a hierarchical partition of U.

Proof. The properties of partition at each level and singleton partition at infinite level are immediate from the definition of H_c. The value-bounded partitions property follows from the bounded concretization property of $\alpha(\cdot, \cdot)$.

To prove the partition with refinement property, let $c \in \mathbb{Z}_+$ and $S \in H_c$. Let T be the set of equivalence classes of S under the relation $\sim_{c+1}$, which forms a partition of S. Since $S \subseteq U$ and H_{c+1} is the set of equivalence classes of U under the relation $\sim_{c+1}$, for each $P \in T$, there exists $P' \in H_{c+1}$ such that $P \subseteq P'$. Take arbitrary elements $u \in P$ and $v \in P'$. Then there exists $w \in U$ such that $\alpha(u, c+1) = \alpha(v, c+1) = w$ since u and v both belong to P'. The capability path independence property implies $\alpha(u, c) = \alpha(v, c) = \alpha(w, c)$, which means $u \sim_c v$. Since $u \in P \subseteq S$, we have $v \in S$ by the definition of H_c, which implies $v \in P$. Therefore, we have $P = P'$, which implies $T \subseteq H_{c+1}$.

To prove the closed representation property, let $S \in H_c$ for some $c \in \mathbb{Z}_+$. The definition of H_c implies that there exists $w \in U$ such that for all $u \in S$, $\alpha(u, c) = w$. Theorem 2.1 implies that $\alpha(w, c) = w = \alpha(u, c)$. We thus have $w \sim_c u$ for all $u \in S$, which means $w \in S$ by our definition of H_c. The closed representation property is immediate from $r(S) = w \in S$. To show that the same S from different choices of c does not cause any conflict so that $r(\cdot)$ exists, assume there is $c' > c$ such that $S \in H_{c'}$. We have $u \sim_{c'} w$ for all $u \in S$, which implies $\alpha(u, c') = \alpha(w, c')$; with Theorem 2.1 we also have $\alpha(w, c') = \alpha(w, c) = w$. Therefore, $r(S) = w$ can be obtained from either H_c or $H_{c'}$ and is thus well-defined.

To prove the stable representation property, let $S \subseteq T$ where $S \in H_c$ and $T \in H_{c'}$ for some $c, c' \in \overline{\mathbb{Z}}_+$. Let $w = r(T) \in S$. Since $S \subseteq T$, we have $c \geq c'$ due to the partition with refinement property. Let $v = r(S) \in S$. The definition of $r(\cdot)$ implies that $\alpha(w, c) = \alpha(v, c) = v$ and $\alpha(w, c') = \alpha(v, c') = w$. The capability path independence property implies $w = \alpha(w, c') = \alpha(\alpha(w, c'), c) = \alpha(w, c) = v$. Therefore, $r(S) = r(T)$. □

Theorem 2.4 (From hierarchical partition to payoff perception function) *Let $U := \mathbb{R}^{\mathcal{S}}$ be the set of possible payoff functions over a finite strategy space $\mathcal{S} := \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$. Let $\left((H_c)_{c \in \overline{\mathbb{Z}}_+}, r\right)$ be a hierarchical partition of U. Define $\alpha : \left(U \times \overline{\mathbb{Z}}_+\right) \mapsto U$ as $\alpha(u, c) = r(S)$ for any given $u \in U$ and $c \in \overline{\mathbb{Z}}_+$ such that $S \in H_c$ and $u \in S$. Then $\alpha(\cdot, \cdot)$ is a payoff perception function.*

Proof. First note that the choice of S in the definition of $\alpha(\cdot, \cdot)$ is unique since H_c is a partition of U. To prove the capability path independence property, take any $u \in U$ and $c_1, c_2 \in \overline{\mathbb{Z}}_+$. Let $S_1 \in H_{c_1}$ such that $u \in S_1$. Let $w := \alpha(u, c_1) = r(S_1) \in S_1$. Consider two cases:

- $c_1 \leq c_2$: By recursively applying the partition with refinement property, there exists $P \subseteq H_{c_2}$ such that P is a partition of S_1. Since $w \in S_1$, there exists $S_2 \in P$ such that $r(S_1) = w \in S_2$. The stable representation property implies $r(S_2) = w$. Therefore, $\alpha(w, c_2) = r(S_2) = w$ and thus $\alpha(\alpha(u, c_1), c_2) = \alpha(r(S_1), c_2) = w = \alpha(u, c_1)$.

- $c_1 > c_2$: Let $S_2 \in H_{c_2}$ such that $w \in S_2$. Let $v := \alpha(w, c_2) = r(S_2)$. The partition with refinement property implies that there exists $P \subseteq H_{c_1}$ such that P is a partition of S_2. Since $w \in S_1$, $w \in S_2$, and $S_1 \in H_{c_1}$, we have $S_1 \in P$ and thus $S_1 \subseteq S_2$. Since $u \in S_1$, we have $u \in S_2$ and $\alpha(u, c_2) = r(S_2) = v = \alpha(w, c_2) = \alpha(\alpha(u, c_1), c_2)$.

Combining the two cases yields $\alpha(\alpha(u,\, c_1),\, c_2) = \alpha(u,\, \min\{c_1,\, c_2\})$.

The perfect perception with infinite capability follows immediately from the singleton partition at infinite level property. The bounded concretization property follows from the value-bounded partitions property. □

The above two theorems show that the payoff perception function and the hierarchical partition of payoff functions are equivalent, as summarized below.

Corollary 2.5
Let $U \coloneqq \mathbb{R}^{\mathcal{S}}$ be the set of possible payoff functions over a finite strategy space $\mathcal{S} \coloneqq \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$. Then there is a bijection between the set of payoff perception functions on U and the set of hierarchical partitions of U.

2.4 Payoff bounds and related properties

Since players are assumed to use mixed strategies to optimize their expected payoff, let's first define simplified notation for the expected payoff given a mixed strategy profile:

Definition 2.6 (Expected payoff notation)
Given finite strategy spaces $(\mathcal{S}_i)_{i\in\mathcal{N}}$ for the players, let $u \in \mathbb{R}^{\mathcal{S}}$ be a payoff function with $\mathcal{S} \coloneqq \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$. For an uncorrelated mixed strategy profile $(\boldsymbol{\sigma}_i)_{i\in\mathcal{N}}$ where $\boldsymbol{\sigma}_i \in \Delta_{\mathcal{S}_i}$, the expected payoff of this mixed strategy profile is denoted as $u(\boldsymbol{\sigma}_1,\, \ldots,\, \boldsymbol{\sigma}_n)$ and defined as

$$\begin{aligned} u(\boldsymbol{\sigma}_1,\, \ldots,\, \boldsymbol{\sigma}_n) &\coloneqq \mathbb{E}_{s_1\sim\boldsymbol{\sigma}_1,\ldots,s_n\sim\boldsymbol{\sigma}_n}[u(s_1,\, \ldots,\, s_n)] \\ &= \sum_{(s_1,\ldots,s_n)\in\mathcal{S}} \prod_{i\in\mathcal{N}} \boldsymbol{\sigma}_i(s_i) u(s_1,\, \ldots,\, s_n) \end{aligned} \tag{2.3}$$

◁

Definition 2.7 (Bounds of expected true payoff)
In an OLP-Game as defined in Definition 2.3, let $(\boldsymbol{\sigma}_j)_{j\in\mathcal{N}}$ be a mixed strategy profile with $\boldsymbol{\sigma}_j \in \Delta_{\mathcal{S}_j}$. If a player perceives their payoff function as v and has a capability of c, then the lower and upper bounds of the expected true payoffs are denoted as

$$\begin{aligned} \mathcal{P}^-\Big(v,\, c,\, (\boldsymbol{\sigma}_j)_{j\in\mathcal{N}}\Big) &\coloneqq \inf_{u\in\alpha^{-1}(v,\,c)} u(\boldsymbol{\sigma}_1,\, \ldots,\, \boldsymbol{\sigma}_n) \\ \mathcal{P}^+\Big(v,\, c,\, (\boldsymbol{\sigma}_j)_{j\in\mathcal{N}}\Big) &\coloneqq \sup_{u\in\alpha^{-1}(v,\,c)} u(\boldsymbol{\sigma}_1,\, \ldots,\, \boldsymbol{\sigma}_n) \end{aligned} \tag{2.4}$$

If the discussion is focused on the capability-aware player i with a perceived payoff function of $v_i^i(\cdot)$ and a capability of c_i, given other players' strategy profile $\boldsymbol{\sigma}_{-i}$, the lower and upper bounds of the expected true payoffs of the player i for their strategy $\boldsymbol{\sigma}_i$ are denoted as:

$$\begin{aligned} \mathcal{P}^-\left(v,\, c,\, \boldsymbol{\sigma}_i \mid \boldsymbol{\sigma}_{-i}\right) &\coloneqq \mathcal{P}^-\Big(v,\, c,\, (\boldsymbol{\sigma}_j)_{j\in\mathcal{N}}\Big) \\ \mathcal{P}^+\left(v,\, c,\, \boldsymbol{\sigma}_i \mid \boldsymbol{\sigma}_{-i}\right) &\coloneqq \mathcal{P}^+\Big(v,\, c,\, (\boldsymbol{\sigma}_j)_{j\in\mathcal{N}}\Big) \end{aligned} \tag{2.5}$$

◁

Remark 2.1 The bounds given in Definition 2.7 are finite due to the bounded concretization property of the payoff perception function.

Theorem 2.6 (Monotonicity of payoff bounds)
With the notation in Definition 2.7, for any $v \in U$, $c \in \mathbb{Z}_+$, and $\boldsymbol{\sigma}_j \in \Delta_{\mathcal{S}_j}$ for $j \in \mathcal{N}$, it holds that

$$\begin{aligned} \mathcal{P}^-\left(v,\, c,\, (\boldsymbol{\sigma}_j)_{j\in\mathcal{N}}\right) &\le \mathcal{P}^-\left(v,\, c+1,\, (\boldsymbol{\sigma}_j)_{j\in\mathcal{N}}\right) \\ &\le \mathcal{P}^+\left(v,\, c+1,\, (\boldsymbol{\sigma}_j)_{j\in\mathcal{N}}\right) \le \mathcal{P}^+\left(v,\, c,\, (\boldsymbol{\sigma}_j)_{j\in\mathcal{N}}\right) \end{aligned} \tag{2.6}$$

Let $v' = \alpha(v,\, c)$, then for $c' \ge c$, it holds that

$$\begin{aligned} \mathcal{P}^-\left(v',\, c,\, (\boldsymbol{\sigma}_j)_{j\in\mathcal{N}}\right) &\le \mathcal{P}^-\left(v,\, c',\, (\boldsymbol{\sigma}_j)_{j\in\mathcal{N}}\right) \\ &\le \mathcal{P}^+\left(v,\, c',\, (\boldsymbol{\sigma}_j)_{j\in\mathcal{N}}\right) \le \mathcal{P}^+\left(v',\, c,\, (\boldsymbol{\sigma}_j)_{j\in\mathcal{N}}\right) \end{aligned} \tag{2.7}$$

Proof. Eqs. (2.6) and (2.7) follow directly substituting Theorem 2.2 into (2.4). □

The following theorem provides an alternative representation of the payoff bounds, which is useful for certain analysis:

Theorem 2.7 (Bounds of expected true payoff via linear objectives over convex sets)
With the notation in Definition 2.7, for the player i, their payoff characteristic set *is*

$$\mathcal{C}\left(v_i^i,\, c_i \mid \boldsymbol{\sigma}_{-i}\right) \coloneqq \operatorname{conv}\operatorname{cl}\left\{\mathfrak{p}_i\left(u \mid \boldsymbol{\sigma}_{-i}\right) \;\middle|\; u \in \alpha^{-1}\left(v_i^i,\, c_i\right)\right\}, \tag{2.8}$$

where the function $\mathfrak{p}_i : U \mapsto \mathbb{R}^{|\mathcal{S}_i|}$ summarizes the expected payoff of each action of the player i. If $\boldsymbol{y} = \mathfrak{p}_i\left(u \mid \boldsymbol{\sigma}_{-i}\right)$, then $y_j = \mathbb{E}_{s_i=j,\, s_{-i}\sim\boldsymbol{\sigma}_{-i}}[u(s_1,\, \ldots,\, s_n)]$.

Then $\mathcal{C}\left(v_i^i,\, c_i \mid \boldsymbol{\sigma}_{-i}\right)$ is nonempty, compact, and convex.

Moreover, it holds that

$$\begin{aligned} \mathcal{P}^-\left(v,\, c,\, \boldsymbol{\sigma}_i \mid \boldsymbol{\sigma}_{-i}\right) &= \min_{\boldsymbol{y}\in\mathcal{C}\left(v_i^i,\, c_i \,\middle|\, \boldsymbol{\sigma}_{-i}\right)} \boldsymbol{y}^\intercal \boldsymbol{\sigma}_i \\ \mathcal{P}^+\left(v,\, c,\, \boldsymbol{\sigma}_i \mid \boldsymbol{\sigma}_{-i}\right) &= \max_{\boldsymbol{y}\in\mathcal{C}\left(v_i^i,\, c_i \,\middle|\, \boldsymbol{\sigma}_{-i}\right)} \boldsymbol{y}^\intercal \boldsymbol{\sigma}_i \end{aligned} \tag{2.9}$$

Proof. Assume v_i^i, c_i, and $\boldsymbol{\sigma}_{-i}$ are fixed. Let $C_0 \coloneqq \{\mathfrak{p}_i\left(u \mid \boldsymbol{\sigma}_{-i}\right) \mid u \in \alpha^{-1}(v_i^i,\, c_i)\}$. Then $\mathcal{C}\left(v_i^i,\, c_i \mid \boldsymbol{\sigma}_{-i}\right) = \operatorname{conv}\operatorname{cl} C_0$.

The first step is to show that $\mathcal{C}\left(v_i^i,\, c_i \mid \boldsymbol{\sigma}_{-i}\right)$ is nonempty, compact, and convex. The property of perfect perception with infinite capability implies that $v_i^i \in \alpha^{-1}(v_i^i,\, c_i)$, and thus $\mathcal{C}\left(v_i^i,\, c_i \mid \boldsymbol{\sigma}_{-i}\right)$ is nonempty. The property of bounded concretization implies that C_0 is bounded. Thus $\operatorname{cl} C_0$ is compact and $\operatorname{conv}\operatorname{cl} C_0$ is compact and convex [Lay07, Theorem 2.30].

Let's prove the equation of the lower bound in (2.9). The upper bound can be proved similarly. Given $u \in \alpha^{-1}(v_i^i,\, c_i)$, let $\boldsymbol{y} = \mathfrak{p}_i\left(u \mid \boldsymbol{\sigma}_{-i}\right)$. The law of total expectation implies

$$\mathbb{E}_{s_1\sim\boldsymbol{\sigma}_1,\, \ldots,\, s_n\sim\boldsymbol{\sigma}_n}[u(s_1,\, \ldots,\, s_n)] = \sum_{j\in\mathcal{S}_i} \mathbb{E}_{s_i=j,\, s_{-i}\sim\boldsymbol{\sigma}_{-i}}[u(s_1,\, \ldots,\, s_n)]\mathbb{P}[s_i = j] = \boldsymbol{y}^\intercal \boldsymbol{\sigma}_i,$$

which, when combined with (2.5), yields

$$\mathcal{P}^{-}\left(v,\, c,\, \boldsymbol{\sigma}_i \mid \boldsymbol{\sigma}_{-i}\right) = \inf_{\boldsymbol{y} \in C_0} \boldsymbol{y}^{\intercal} \boldsymbol{\sigma}_i$$

Let $C_1 \coloneqq \operatorname{cl} C_0$. It is easy to verify $\min_{\boldsymbol{y} \in C_1} \boldsymbol{y}^{\intercal} \boldsymbol{\sigma}_i = \inf_{\boldsymbol{y} \in C_0} \boldsymbol{y}^{\intercal} \boldsymbol{\sigma}_i$. Let $C_2 = \mathcal{C}\left(v_i^i,\, c_i \mid \boldsymbol{\sigma}_{-i}\right) = \operatorname{conv} C_1$. Since $C_1 \subseteq C_2$, we have $\min_{\boldsymbol{y} \in C_2} \boldsymbol{y}^{\intercal} \boldsymbol{\sigma}_i \leq \min_{\boldsymbol{y} \in C_1} \boldsymbol{y}^{\intercal} \boldsymbol{\sigma}_i$. Let $\boldsymbol{y}^* \coloneqq \operatorname{argmin}^*_{\boldsymbol{y} \in C_2} \boldsymbol{y}^{\intercal} \boldsymbol{\sigma}_i$. Carathéodory's theorem [ECK93] states that there are $k = |\mathcal{S}_i| + 1$ points $\boldsymbol{y}_1^*, \ldots, \boldsymbol{y}_k^* \in C_1$ and $\boldsymbol{w} \in \Delta_k$ such that $\boldsymbol{y}^* = \sum_{j=1}^{k} w_j \boldsymbol{y}_j^*$, which implies

$$\boldsymbol{y}^{*\intercal} \boldsymbol{\sigma}_i = \sum_{j=1}^{k} w_j \boldsymbol{y}_j^{*\intercal} \boldsymbol{\sigma}_i \geq \sum_{j=1}^{k} w_j \left(\min_{l \in [\![1, k]\!]} \boldsymbol{y}_l^{*\intercal} \boldsymbol{\sigma}_i \right) = \min_{l \in [\![1, k]\!]} \boldsymbol{y}_l^{*\intercal} \boldsymbol{\sigma}_i \geq \min_{\boldsymbol{y} \in C_1} \boldsymbol{y}^{\intercal} \boldsymbol{\sigma}_i$$

It follows that $\min_{\boldsymbol{y} \in C_2} \boldsymbol{y}^{\intercal} \boldsymbol{\sigma}_i = \min_{\boldsymbol{y} \in C_1} \boldsymbol{y}^{\intercal} \boldsymbol{\sigma}_i = \inf_{\boldsymbol{y} \in C_0}$, which completes the proof. □

Eq. (2.9) reframes the bounds of the expected true payoff as the extreme values of a linear objective. Since a linear function is both convex and concave, we obtain the convexity and concavity of the payoff bounds:

Theorem 2.8 (Convexity and concavity of the payoff bounds)
With the notation in Definition 2.7, $\mathcal{P}^{-}\left(v,\, c,\, \boldsymbol{\sigma}_i \mid \boldsymbol{\sigma}_{-i}\right)$ is continuous and concave in $\boldsymbol{\sigma}_i$, and $\mathcal{P}^{+}\left(v,\, c,\, \boldsymbol{\sigma}_i \mid \boldsymbol{\sigma}_{-i}\right)$ is continuous and convex in $\boldsymbol{\sigma}_i$. Both functions are bounded. Both are continuous in $\boldsymbol{\sigma}_{-i}$.

Proof. Given (2.4), the continuity follows from the fact that $u(\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n)$ is continuous in $\boldsymbol{\sigma}_i$ for $i \in \mathcal{N}$, and the point-wise infimum/supremum of a family of continuous functions is continuous [BD84, Theorem 7.4.9]. Given (2.9), the convexity/concavity follows from the fact that the point-wise supremum (resp. infimum) of a family of convex (resp. concave) functions is convex (resp. concave) [HL96a, Chapter IV, Proposition 2.1.2]. The boundedness follows directly from the bounded concretization property of the payoff perception function (Definition 2.2). □

Of note, the convexity/concavity proof in Theorem 2.8 only requires the continuity of $u(\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n)$ in $\boldsymbol{\sigma}_i$ for any $i \in \mathcal{N}$, which means it can be proved without (2.9). Eq. (2.9) is presented to provide a more intuitive interpretation of the payoff bounds; in particular, it shows that maximizing $\mathcal{P}^{-}\left(v,\, c,\, \boldsymbol{\sigma}_i \mid \boldsymbol{\sigma}_{-i}\right)$ over $\boldsymbol{\sigma}_i$ is a Bilinear Saddle Point Problem (BSPP), which may enjoy efficient algorithms [LJJ20; XHZ21].

Chapter 3

Capability-oblivious players in limited-rank games

This chapter discusses *capability-oblivious* players in limited-perception games. A capability-oblivious player does not know that their capability is limited, and thus reasons as if their perceived payoff functions are the true payoff functions by choosing the optimal strategy of the perceived game.

Definition 3.1 (Capability-oblivious player)
In an OLP-Game as defined in Definition 2.3, consider the player i. Let $\boldsymbol{\sigma}_{-i}$ be the mixed strategy profile of the other players. Player i is *capability-oblivious* if and only if their best response set is defined as:

$$\mathrm{BR}_i^{\mathrm{co}}\left(\boldsymbol{\sigma}_{-i}\right) \coloneqq \operatorname*{argmax}_{\boldsymbol{\sigma}_i' \in \Delta_{\mathcal{S}_i}} v_i^i(\boldsymbol{\sigma}_i', \boldsymbol{\sigma}_{-i}) \tag{3.1}$$

◁

If player i is capability-oblivious and perceives the payoff functions as $G_i = (v_1^i, \ldots, v_n^i)$, then they (wrongly) believe that all players are in the same true game G_i. Player i's rational behavior is to solve the Nash equilibrium (assuming some equilibrium selection rule) of the game G_i and play their strategy in the equilibrium.

One can anticipate that the outcome of limited-perception games with capability-oblivious players highly depends on the exact definition of the payoff perception function. This chapter focuses on two-player capability-oblivious limited-rank games. As introduced in Section 1.2,

players perceive low-rank approximations of the true payoff functions in a limited-rank game; the limited-rank game is arguably a natural model for players with limited perception. The payoff perception function of a limited-rank game generates a low-rank approximation of the payoff matrix, which can be computed using the truncated Singular Value Decomposition (SVD). In the multi-player case, one can use generalizations of the SVD to higher-order tensors (such as the higher-order SVD [DDV00]) to define limited-rank games; I do not consider the multi-player case in this chapter.

Section 3.1 formalizes the limited-rank games using the truncated SVD. Although the SVD is not unique in many cases, I show that a payoff perception function can be well defined by choosing a particular SVD decomposition. One central question is how the capability level impacts the final payoff of a capability-oblivious player. I answer this question from three perspectives:

- Section 3.2 presents theorems that give closed-form bounds on the true payoffs in a limited-rank game given a perceived payoff matrix, a capability level, and a mixed strategy profile. As one would expect, a higher capability level leads to a narrower range of possible payoffs.

- Section 3.3 analyzes the outcome from a player's perspective. I present theorems that characterize the final payoffs in a few settings assuming both players use their respective equilibrium strategies (which do not necessarily form an equilibrium when combined).

- Section 3.4 discusses the capability transfer properties. I show that it is computationally hard to determine whether increasing the capability level of a player will strictly increase their payoff. I then present comprehensive analysis of the payoff changes for different capability levels in a 2×2 parameterized limited-rank game.

3.1 The limited-rank game as a limited-perception game

This section formalizes the limited-rank game through the truncated Singular Value Decomposition (SVD). I discuss how to deal with the non-uniqueness of the SVD. A closed-form formula for the rank-1 approximation of $n \times 2$ matrices is also provided.

Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ be the true payoff matrix. A player with capability level c perceives a matrix $\alpha(\boldsymbol{A}, c) \coloneqq \widetilde{\boldsymbol{A}}^{(\min\{c,m,n\})}$. Given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $r \in [\![1, \min\{m, n\}]\!]$, $\widetilde{\boldsymbol{A}}^{(r)}$ denotes the best rank-r approximation in the sense that it minimizes $\left\| \boldsymbol{A} - \widetilde{\boldsymbol{A}}^{(r)} \right\|$ given some matrix norm operator $\|\cdot\|$. A norm is *unitarily invariant* if it satisfies $\|\boldsymbol{A}\| = \|\boldsymbol{U}\boldsymbol{A}\boldsymbol{V}\|$ for any appropriately sized orthogonal matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ in addition to the standard properties of matrix norms (i.e., triangle inequality $\|\boldsymbol{X} + \boldsymbol{Y}\| \leq \|\boldsymbol{X}\| + \|\boldsymbol{Y}\|$, absolute homogeneity $\|c\boldsymbol{X}\| = |c|\|X\|$, and positive definiteness $\|\boldsymbol{X}\| = 0 \implies \boldsymbol{X} = \boldsymbol{0}$). This section assumes a fixed unitarily invariant norm. Typical examples of unitarily invariant norms include the *Frobenius norm* ($\|\boldsymbol{A}\|_F \coloneqq \sqrt{\sum_{i,j} A_{i,j}^2}$) and the *spectral norm* ($\|\boldsymbol{A}\|_2 \coloneqq \sup_{\|\boldsymbol{x}\|_2 = 1} \|\boldsymbol{A}\boldsymbol{x}\|_2$).

For any unitarily invariant norm, the best limited-rank approximation is given by the truncated SVD of $\boldsymbol{A}$ [Mir60]. For a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, its SVD is given by $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ where $\boldsymbol{U} \in \mathbb{R}^{m \times m}$ and $\boldsymbol{V} \in \mathbb{R}^{n \times n}$ are orthogonal matrices and $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times n}$ is a diagonal

matrix with non-negative entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{m,n\}} \geq 0$. We can rewrite $\boldsymbol{A}$ as $\boldsymbol{A} = \sum_{i=1}^{\min\{m,n\}} \sigma_i \boldsymbol{U}_{:,i} \boldsymbol{V}_{:,i}^\intercal$. Its rank-$r$ approximation by truncated SVD is $\widetilde{\boldsymbol{A}}^{(r)} = \sum_{i=1}^{r} \sigma_i \boldsymbol{U}_{:,i} \boldsymbol{V}_{:,i}^\intercal$. The approximation error is $\left\|\boldsymbol{A} - \widetilde{\boldsymbol{A}}^{(r)}\right\| = \left\|\operatorname{diag}\left(\sigma_{r+1}, \ldots, \sigma_{\min\{m,n\}}\right)\right\|$ [Mir60].

The SVD of a real-valued matrix always exists and is unique up to unitary transformations of the subspace corresponding to repeated singular values [TB97, Theorem 4.1]. More precisely, let i and j be indices of a group of repeated singular values such that $\sigma_i = \sigma_{i+1} = \cdots = \sigma_j$. Let $\boldsymbol{U}' \coloneqq \boldsymbol{U}_{:,i:j}$ and $\boldsymbol{V}' \coloneqq \boldsymbol{V}_{:,i:j}$ be the corresponding singular vectors. Then any decomposition that replaces $\boldsymbol{U}'$ and $\boldsymbol{V}'$ with $\boldsymbol{U}'\boldsymbol{Q}$ and $\boldsymbol{V}'\boldsymbol{Q}$ for appropriately sized orthogonal matrix $\boldsymbol{Q}$ is also a valid SVD of $\boldsymbol{A}$, and all SVDs of $\boldsymbol{A}$ can be obtained in this way. Of note, the singular values are unique when ordered in non-increasing order.

Let $\operatorname{LR}(\boldsymbol{A}, r) \subseteq \mathbb{R}^{m \times n \times 3}$ denote the set of all possible truncated SVDs of $\boldsymbol{A}$ that keep the largest r singular values with corresponding singular vectors. Formally, $(\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}) \in \operatorname{LR}(\boldsymbol{A}, r)$ if and only if $\boldsymbol{U}^\intercal\boldsymbol{U} = \boldsymbol{I}_m$, $\boldsymbol{V}^\intercal\boldsymbol{V} = \boldsymbol{I}_n$, $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_r, 0, \ldots, 0)$ with $\sigma_i \geq \sigma_{i+1} \geq 0$ for $i \in [\![1, r-1]\!]$, and there exists non-negative numbers $\left\{\sigma_{r+1}, \ldots, \sigma_{\min\{m,n\}}\right\}$ such that $\sigma_k \geq \sigma_{k+1} \geq 0$ for $k \in [\![r, \min\{m, n\}]\!]$ and $\boldsymbol{A} = \boldsymbol{U} \operatorname{diag}\left(\sigma_1, \ldots, \sigma_{\min\{m,n\}}\right) \boldsymbol{V}^\intercal$.

Lemma 3.1
For any $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $r \in [\![1, \min\{m, n\}]\!]$, the set $\operatorname{LR}(\boldsymbol{A}, r)$ endowed with any norm topology is nonempty and compact.

Proof. The nonemptiness of $\operatorname{LR}(\boldsymbol{A}, r)$ follows from the existence of the SVD of $\boldsymbol{A}$. Take an arbitrary item $(\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}) \in \operatorname{LR}(\boldsymbol{A}, r)$. Then all other items in $\operatorname{LR}(\boldsymbol{A}, r)$ can be obtained by replacing $\boldsymbol{U}$ and $\boldsymbol{V}$ with $\boldsymbol{U}\boldsymbol{Q}$ and $\boldsymbol{V}\boldsymbol{Q}$ for a matrix $\boldsymbol{Q}$ that is block-orthogonal where the blocks correspond to the repeated singular values of $\boldsymbol{A}$. The mapping $\boldsymbol{Q} \mapsto (\boldsymbol{U}\boldsymbol{Q}, \boldsymbol{\Sigma}, \boldsymbol{V}\boldsymbol{Q})$ is continuous. Since the set of orthogonal matrices is the inverse image of the continuous function $\boldsymbol{M} \mapsto \boldsymbol{M}^\intercal\boldsymbol{M}$ at $\boldsymbol{I}$, it is closed. Since the column vectors of an orthogonal matrix are orthonormal, the set of orthogonal matrices is bounded. Thus $\operatorname{LR}(\boldsymbol{A}, r)$ is also compact. □

There are many ways to define a unique value of $\widetilde{\boldsymbol{A}}^{(r)}$ such that the derived payoff perception function is valid and odd (Definitions 2.2 and 4.11). To characterize the possible choices, I define the following *perception-compatible choice function* on $\operatorname{LR}(\boldsymbol{A}, r)$:

Definition 3.2 (Perception-compatible choice function)
A function $\mathfrak{f} : 2^{\mathbb{R}^{m \times n \times 3}} \to \mathbb{R}^{m \times n}$ is a *perception-compatible choice function* if and only if it satisfies the following properties:

- For $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $r \in [\![1, \min\{m, n\}]\!]$, $\mathfrak{f}(\operatorname{LR}(\boldsymbol{A}, r)) \in \operatorname{LR}(\boldsymbol{A}, r)$.
- For $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $1 \leq r_1 < r_2 \leq \min\{m, n\}$, $\mathfrak{f}(\operatorname{LR}(\boldsymbol{A}, r_1)) = \mathfrak{f}(\operatorname{LR}(\mathfrak{f}(\operatorname{LR}(\boldsymbol{A}, r_2)), r_1))$.
- For $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $r \in [\![1, \min\{m, n\}]\!]$, $\mathfrak{f}(\operatorname{LR}(-\boldsymbol{A}, r)) = -\mathfrak{f}(\operatorname{LR}(\boldsymbol{A}, r))$. ◁

It easy to verify that setting $\alpha(\boldsymbol{A}, c) = \widetilde{\boldsymbol{A}}^{(\min\{c,m,n\})} = \mathfrak{f}(\operatorname{LR}(\boldsymbol{A}, \min\{c, m, n\}))$ for any perception-compatible choice function $\mathfrak{f}(\cdot)$ yields an odd payoff perception function $\alpha(\cdot, \cdot)$.

One way to define a perception-compatible choice function is to impose a preference over the basis vectors in the SVD. For example, define

$$p_i^j(S) \coloneqq \left\{ (\boldsymbol{U},\, \boldsymbol{\Sigma},\, \boldsymbol{V}) \in S \;\middle|\; U_{j,\,i} = \inf_{(\boldsymbol{U}',\, \boldsymbol{\Sigma}',\, \boldsymbol{V}') \in S} U'_{j,\,i} \right\},$$
$$q_i(S) \coloneqq \left(p_i^1 \circ \cdots \circ p_i^m\right)(S), \quad f(S) \coloneqq (q_1 \circ \cdots \circ q_m)(S),$$

where $(f_1 \circ f_2)(x)$ denotes function composition $f_1(f_2(x))$. Since $U_{i,\,j}$ is continuous in $\boldsymbol{U}$, the set $p_i^j(S)$ is nonempty and compact if S is nonempty and compact. Intuitively, the function $f(S)$ selects the SVD whose $\boldsymbol{U}$ has the smallest lexicographically ordered columns. One can verify that if $(\boldsymbol{U},\, \boldsymbol{\Sigma},\, \boldsymbol{V}) \in q_i(S)$ and $(\boldsymbol{U}',\, \boldsymbol{\Sigma}',\, \boldsymbol{V}') \in q_i(S)$, then $\boldsymbol{U}_{:,\,i} = \boldsymbol{U}'_{:,\,i}$; with similar reasoning, if $(\boldsymbol{U},\, \boldsymbol{\Sigma},\, \boldsymbol{V}) \in f(S)$ and $(\boldsymbol{U}',\, \boldsymbol{\Sigma}',\, \boldsymbol{V}') \in f(S)$, then $\boldsymbol{U} = \boldsymbol{U}'$. Since $\boldsymbol{\Sigma}$ is unique and $\boldsymbol{V}$ is uniquely determined by $\boldsymbol{U}$ via $\boldsymbol{V}^\intercal = \boldsymbol{\Sigma}^{-1}\boldsymbol{U}^\intercal\boldsymbol{A}$, $f(S)$ contains exactly one item. Define $\mathfrak{f}(S) \coloneqq f(S)$ for any nonempty and compact S. Then $\mathfrak{f}(\cdot)$ is a perception-compatible choice function.

Given the above results, the limited-rank games are formally defined as follows:

Definition 3.3 (Limited-rank game)
A limited-rank game is a two-player OLP-Game (Definitions 2.3 and 4.4) where the payoff perception function is defined as $\alpha(\boldsymbol{A},\, c) = \mathfrak{f}(\mathrm{LR}\,(\boldsymbol{A},\, \min\{c,\, m,\, n\}))$ for some perception-compatible choice function $\mathfrak{f}(\cdot)$, and $m \coloneqq |\mathcal{S}_1|$ and $n \coloneqq |\mathcal{S}_2|$ are the numbers of strategies for the row and column players, respectively. The first player is the row player and the second player is the column player.

In an instance of a limited-rank game, the perceived payoff functions are typically denoted by matrix pairs $(\boldsymbol{A}_1,\, \boldsymbol{B}_1)$ and $(\boldsymbol{A}_2,\, \boldsymbol{B}_2)$ for the row and column players, respectively. If the game is zero-sum, then the perceived payoff functions are denoted by $(\boldsymbol{A}_1,\, -\boldsymbol{A}_1)$ and $(\boldsymbol{A}_2,\, -\boldsymbol{A}_2)$ for the row and column players, respectively. ◁

There is no known closed-form formula for the SVD of arbitrarily sized matrices. Practical algorithms use iterative methods to compute the SVD numerically [TB97]. Below is a closed-form formula for the rank-1 approximation of $n \times 2$ matrices, which reveals how columns in the approximation depend on the interior angle and relative lengths of the columns of the original matrix.

Lemma 3.2 (Rank-1 approximation of $n \times 2$ matrices)
Let $\boldsymbol{A} = \begin{bmatrix}\boldsymbol{a}_1 & \boldsymbol{a}_2\end{bmatrix} \in \mathbb{R}^{n\times 2}$ be a $n \times 2$ matrix. It holds that:

$$\widetilde{\boldsymbol{A}}^{(1)} = \begin{cases} \begin{bmatrix}\boldsymbol{0} & \boldsymbol{a}_2\end{bmatrix} & \text{if } Q_{\boldsymbol{A}} = 0 \text{ and } R_{\boldsymbol{A}} < 0 \\ \boldsymbol{A}\begin{bmatrix}\alpha \\ \beta\end{bmatrix}\begin{bmatrix}\alpha & \beta\end{bmatrix} & \text{for some } \alpha^2+\beta^2 = 1 \text{ if } Q_{\boldsymbol{A}} = 0 \text{ and } R_{\boldsymbol{A}} = 0 \\ \begin{bmatrix}\boldsymbol{v} & t\boldsymbol{v}\end{bmatrix} & \text{if } Q_{\boldsymbol{A}} \neq 0 \text{ or } (Q_{\boldsymbol{A}} = 0 \text{ and } R_{\boldsymbol{A}} > 0) \end{cases}$$

$$\text{where } Q_{\boldsymbol{A}} \coloneqq \boldsymbol{a}_1^\intercal \boldsymbol{a}_2, \quad R_{\boldsymbol{A}} \coloneqq \|\boldsymbol{a}_1\|_2^2 - \|\boldsymbol{a}_2\|_2^2, \quad \boldsymbol{v} \coloneqq \frac{1}{t^2+1}(\boldsymbol{a}_1 + t\boldsymbol{a}_2) \tag{3.2}$$

$$t \coloneqq \begin{cases} 0 & \text{if } Q_{\boldsymbol{A}} = 0 \text{ and } R_{\boldsymbol{A}} > 0 \\ \frac{1}{2}(-s+m) & \text{if } Q_{\boldsymbol{A}} \neq 0 \end{cases}$$

$$s \coloneqq \frac{R_{\boldsymbol{A}}}{Q_{\boldsymbol{A}}}, \quad m \coloneqq \begin{cases} \sqrt{s^2+4} & \text{if } Q_{\boldsymbol{A}} > 0 \\ -\sqrt{s^2+4} & \text{if } Q_{\boldsymbol{A}} < 0 \end{cases}$$

Furthermore, if $Q_{\boldsymbol{A}} \neq 0$, then

$$\operatorname{sign}(t) = \operatorname{sign}(Q_{\boldsymbol{A}}), \quad \operatorname{sign}(|t|-1) = \operatorname{sign}(-R_{\boldsymbol{A}}) \tag{3.3}$$

Analogous results hold by switching the rows and columns if $\boldsymbol{A} \in \mathbb{R}^{2\times n}$.

Proof. Let $\widetilde{\boldsymbol{A}}^{(1)} = \begin{bmatrix}\boldsymbol{v} & t\boldsymbol{v}\end{bmatrix}$. To ease the exposition, let's extend $\boldsymbol{v}$ and t to be hyperreals (i.e., $\boldsymbol{v} \in {}^*\mathbb{R}^n$ and $t \in {}^*\mathbb{R}$) so that they can take infinitesimal/infinite values while $t\boldsymbol{v}$ remains finite [Gol98]. Define $L \coloneqq \left\|\boldsymbol{A} - \widetilde{\boldsymbol{A}}^{(1)}\right\|_F^2$ which should be minimized. It holds that:

$$\begin{aligned} L &= \|\boldsymbol{a}_1 - \boldsymbol{v}\|_2^2 + \|\boldsymbol{a}_2 - t\boldsymbol{v}\|_2^2 \\ &= \left(t^2+1\right)\boldsymbol{v}^\intercal\boldsymbol{v} - 2(\boldsymbol{a}_1 + t\boldsymbol{a}_2)^\intercal\boldsymbol{v} + \boldsymbol{a}_1^\intercal\boldsymbol{a}_1 + \boldsymbol{a}_2^\intercal\boldsymbol{a}_2 \\ &= \left(t^2+1\right)\left\|\boldsymbol{v} - \frac{\boldsymbol{a}_1 + t\boldsymbol{a}_2}{t^2+1}\right\|_2^2 - \frac{\|\boldsymbol{a}_1 + t\boldsymbol{a}_2\|_2^2}{t^2+1} + \boldsymbol{a}_1^\intercal\boldsymbol{a}_1 + \boldsymbol{a}_2^\intercal\boldsymbol{a}_2 \\ &= \left(t^2+1\right)\left\|\boldsymbol{v} - \frac{\boldsymbol{a}_1 + t\boldsymbol{a}_2}{t^2+1}\right\|_2^2 - \frac{2\boldsymbol{a}_1^\intercal\boldsymbol{a}_2 t + \boldsymbol{a}_1^\intercal\boldsymbol{a}_1 - \boldsymbol{a}_2^\intercal\boldsymbol{a}_2}{t^2+1} + \boldsymbol{a}_1^\intercal\boldsymbol{a}_1 \\ &= \left(t^2+1\right)\left\|\boldsymbol{v} - \frac{\boldsymbol{a}_1 + t\boldsymbol{a}_2}{t^2+1}\right\|_2^2 - \frac{2Qt + R}{t^2+1} + \boldsymbol{a}_1^\intercal\boldsymbol{a}_1 \end{aligned} \tag{3.4}$$

where $Q \coloneqq Q_{\boldsymbol{A}}$, $R \coloneqq R_{\boldsymbol{A}}$

To minimize L, we need to set $\boldsymbol{v} = \frac{\boldsymbol{a}_1 + t\boldsymbol{a}_2}{t^2+1}$ while maximizing $f(t) \coloneqq \frac{2Qt+R}{t^2+1}$. Consider the following cases:

- When $Q = R = 0$, we have $f(t) = 0$ for all $t \in {}^*\mathbb{R}$, including $t = \infty$. With $\alpha \coloneqq \frac{1}{\sqrt{t^2+1}}$ and $\beta \coloneqq \frac{t}{\sqrt{t^2+1}}$, we rewrite $\begin{bmatrix}\boldsymbol{v} & t\boldsymbol{v}\end{bmatrix} = \boldsymbol{A}\begin{bmatrix}\alpha \\ \beta\end{bmatrix}\begin{bmatrix}\alpha & \beta\end{bmatrix}$. The equivalent constraint on α and β is $\alpha^2 + \beta^2 = 1$ that does not involve hyperreals.

- When $Q = 0$ and $R > 0$, $f(t)$ is maximized at $t = 0$.
- When $Q = 0$ and $R < 0$, $f(t)$ is maximized at $t = \infty$.
- When $Q \neq 0$, we have $\frac{\partial f}{\partial t} = -2\frac{Qt^2+Rt-Q}{(t^2+1)^2} = -2Q\frac{t^2+st-1}{(t^2+1)^2}$. Setting $\frac{\partial f}{\partial t} = 0$ gives $t = \frac{1}{2}\left(-s \pm \sqrt{s^2+4}\right)$. When $Q > 0$, $f(t)$ is maximized at $t = \frac{1}{2}\left(-s + \sqrt{s^2+4}\right)$; when $Q < 0$, $f(t)$ is maximized at $t = \frac{1}{2}\left(-s - \sqrt{s^2+4}\right)$.

Eq. (3.2) summarizes the above results. Note that since $|s| < |m| < |s|+2$, we have $\operatorname{sign}(t) = \operatorname{sign}(m) = \operatorname{sign}(Q)$. Also note that $|m| > 2$, which gives $\operatorname{sign}(|t| - 1) = \operatorname{sign}(|m - s| - 2) = \operatorname{sign}(-ms) = \operatorname{sign}(-R)$. Thus we have also proved (3.3). □

3.2 Payoff bounds of limited-rank games

This section analyzes the bounds of true payoff given a perceived payoff function in limited-rank games. Some results are devoted to the fully-capability-aware setting that is discussed in Chapter 4.

The analysis uses the following notation:

- For $r = \overline{\mathbb{Z}}_+$ and $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\sigma_r(\boldsymbol{A})$ denotes the r-th largest singular value of $\boldsymbol{A}$. Note that the value is unique. If $r > \min\{m, n\}$ (including $r = \infty$), then $\sigma_r(\boldsymbol{A}) = 0$.
- For $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\mathcal{N}(\boldsymbol{A})$ is a matrix whose columns form an orthonormal basis for the null space of $\boldsymbol{A}$. More precisely, $\mathcal{N}(\boldsymbol{A}) = \begin{bmatrix}\boldsymbol{b}_1 & \cdots & \boldsymbol{b}_k\end{bmatrix} \in \mathbb{R}^{n\times k}$ where $\boldsymbol{b}_i \in \mathbb{R}^n$, $\boldsymbol{b}_i^\intercal \boldsymbol{b}_j = \mathbb{1}_{i=j}$ for $i, j \in [\![1, k]\!]$, and $\boldsymbol{A}\boldsymbol{x} = 0$ if and only if $\boldsymbol{x} \in \operatorname{span}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$. Specially, if $\operatorname{rank}(\boldsymbol{A}) = m$, then $\mathcal{N}(\boldsymbol{A}) = \boldsymbol{0}$.

The main result of this section is the following theorem:

Theorem 3.3 (Payoff bounds of limited-rank games)
Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and two vectors $\boldsymbol{x} \in \mathbb{R}^m$ and $\boldsymbol{y} \in \mathbb{R}^n$, if $\boldsymbol{A}$ is the perceived payoff function of a player with capability level c in a limited-rank game (Definition 3.3), then the true payoff bounds as defined in Definition 2.7 are given by:

$$\begin{aligned} \mathcal{P}^-(\boldsymbol{A}, c, \boldsymbol{x}, \boldsymbol{y}) &= \boldsymbol{x}^\intercal \boldsymbol{A}\boldsymbol{y} - \mathcal{U}(\boldsymbol{A}, c, \boldsymbol{x}, \boldsymbol{y}) \\ \mathcal{P}^+(\boldsymbol{A}, c, \boldsymbol{x}, \boldsymbol{y}) &= \boldsymbol{x}^\intercal \boldsymbol{A}\boldsymbol{y} + \mathcal{U}(\boldsymbol{A}, c, \boldsymbol{x}, \boldsymbol{y}) \\ \text{where } \mathcal{U}(\boldsymbol{A}, c, \boldsymbol{x}, \boldsymbol{y}) &\coloneqq \sigma_c(\boldsymbol{A}) \cdot \left\|\mathcal{N}^\intercal(\boldsymbol{A}^\intercal)\boldsymbol{x}\right\|_2 \cdot \left\|\mathcal{N}^\intercal(\boldsymbol{A})\boldsymbol{y}\right\|_2 \end{aligned} \tag{3.5}$$

Moreover, the limited-rank game is universally narrowly reversible (Definition 4.23).

Consequently, zero-sum limited-rank games can be efficiently solved using Second-Order Cone Programs (SOCPs). An ϵ-approximate solution to a SOCP can be found in $O(-\log\epsilon)$ iterations using interior-point methods [BV04, Chapter 11].

Corollary 3.4 (Solving zero-sum limited-rank games using SOCP)
In an instance of a zero-sum fully-capability-aware limited-rank game (Definitions 3.3 and 4.12), assume $c_1 < c_2$. Let $(s^*, u^*, \boldsymbol{x}^*)$ be a solution to the following Second-Order Cone Program (SOCP):

$$\begin{aligned} &\max_{s,\,u,\,\boldsymbol{x}} \; s \\ \text{subject to } & s \le \boldsymbol{x}^\intercal \boldsymbol{A}_1 \boldsymbol{e}_i - \sigma_{c_1}(\boldsymbol{A}_1) \cdot u \cdot \left\| \mathcal{N}^\intercal(\boldsymbol{A}_1)\boldsymbol{e}_i \right\|_2 \quad \forall i \in [\![1,\, |\mathcal{S}_2|]\!] \\ & \left\| \mathcal{N}^\intercal(\boldsymbol{A}_1^\intercal)\boldsymbol{x} \right\|_2 \le u \\ & \boldsymbol{x} \in \Delta_{\mathcal{S}_1} \end{aligned} \tag{3.6}$$

Given $\boldsymbol{x}^*$, let $(t^*, v^*, \boldsymbol{y}^*)$ be a solution to the following SOCP:

$$\begin{aligned} &\min_{t,\,v,\,\boldsymbol{y}} \; t \\ \text{subject to } & t \ge \boldsymbol{x}^{*\intercal} \boldsymbol{A}_2 \boldsymbol{y} + \sigma_{c_2}(\boldsymbol{A}_2) \cdot \left\| \mathcal{N}^\intercal(\boldsymbol{A}_2^\intercal)\boldsymbol{x}^* \right\|_2 \cdot v \\ & \left\| \mathcal{N}^\intercal(\boldsymbol{A}_2)\boldsymbol{y} \right\|_2 \le v \\ & \boldsymbol{y} \in \Delta_{\mathcal{S}_2} \end{aligned} \tag{3.7}$$

Then $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is a Stackelberg equilibrium of this instance where the row player is the leader. The Stackelberg value and the Nash value of the row player are both s^*. Furthermore, if the solutions to both (3.6) and (3.7) are unique, then $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is also a Nash equilibrium.

Proof. Theorem 3.3 shows that zero-sum limited-rank games are universally narrowly reversible. Theorem 4.23 then implies that such games are maximin-attainable. Theorem 4.15 yields (3.6) and (3.7) for maximin-attainable games with the payoff bounds given by (3.5). When (3.6) has a unique solution, the game has a unique maximin strategy. Theorem 4.23 implies that the Stackelberg equilibrium is also unique. Since a Nash equilibrium is also a Stackelberg equilibrium in maximin-attainable games, the Nash equilibrium is unique and coincides with the Stackelberg equilibrium. □

As will be argued in Section 4.3.3.2, computation-bounded players should be satisfied with $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ in the Nash setting even if $\boldsymbol{y}^*$ is not used by a Nash equilibrium.

The rest of this section is devoted to proving Theorem 3.3.

Lemma 3.5
For a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and an integer $r \in [\![1, \min\{m, n\} - 1]\!]$, let $\widetilde{\boldsymbol{A}}^{(r)} \in \mathrm{LR}(\boldsymbol{A}, r)$ be a rank-r approximation (not necessarily the result of a perception-compatible choice function).

Given any $\boldsymbol{x} \in \mathbb{R}^m$ and $\boldsymbol{y} \in \mathbb{R}^n$, it holds that

$$\begin{aligned} -s \le \boldsymbol{x}^\intercal \boldsymbol{A} \boldsymbol{y} - \boldsymbol{x}^\intercal \widetilde{\boldsymbol{A}}^{(r)} \boldsymbol{y} \le s \\ \text{where } s \coloneqq \sigma_{r+1}(\boldsymbol{A}) \cdot \left\| \mathcal{N}^\intercal\left(\widetilde{\boldsymbol{A}}^{(r)\,\intercal} \right) \boldsymbol{x} \right\|_2 \cdot \left\| \mathcal{N}^\intercal\left(\widetilde{\boldsymbol{A}}^{(r)} \right) \boldsymbol{y} \right\|_2 \end{aligned} \tag{3.8}$$

The bound can be relaxed to $s \le \sigma_{r+1}(\boldsymbol{A}) \cdot \|\boldsymbol{x}\|_2 \cdot \|\boldsymbol{y}\|_2$.

Proof. Let $f \coloneqq \boldsymbol{x}^\intercal\Big(\boldsymbol{A} - \widetilde{\boldsymbol{A}}^{(r)}\Big)\boldsymbol{y}$ be the value to be bounded. The following proof only deals with the upper bound of f. The lower bound can be proved similarly.

Let $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\intercal$ be the SVD of $\boldsymbol{A}$ that is truncated to obtain $\widetilde{\boldsymbol{A}}^{(r)}$. Then we have $f = \sum_{i=r+1}^{\min\{m,n\}} \sigma_i(\boldsymbol{A})\boldsymbol{x}^\intercal\boldsymbol{U}_{:,i}\boldsymbol{V}_{:,i}^\intercal\boldsymbol{y}$. The value of f can be increased if we negate some columns of $\boldsymbol{U}$ to ensure that $\boldsymbol{x}^\intercal\boldsymbol{U}_{:,i}\boldsymbol{V}_{:,i}^\intercal\boldsymbol{y} > 0$ for $r+1 \le i \le \min\{m, n\}$ and set $\sigma_i(\boldsymbol{A}) = \sigma_{r+1}(\boldsymbol{A})$ for $r+2 \le i \le \min\{m, n\}$. More precisely, we have $f \le f^*$ where

$$
\begin{aligned}
f^* &\coloneqq \boldsymbol{x}^\intercal\Big(\boldsymbol{A}^* - \widetilde{\boldsymbol{A}}^{(r)}\Big)\boldsymbol{y}, \quad \boldsymbol{A}^* \coloneqq \boldsymbol{U}\boldsymbol{S}\boldsymbol{\Sigma}^*\boldsymbol{V}^\intercal, \quad \boldsymbol{S} \coloneqq \operatorname{diag}\left(s_1, s_2, \ldots, s_{\min\{m,n\}}\right) \\
s_i &\coloneqq \begin{cases} -1 & \text{if } i \ge r+1 \text{ and } \boldsymbol{x}^\intercal\boldsymbol{U}_{:,i}\boldsymbol{V}_{:,i}^\intercal\boldsymbol{y} < 0 \\ 1 & \text{otherwise} \end{cases} \\
\boldsymbol{\Sigma}^* &\coloneqq \operatorname{diag}\left(\sigma_1(\boldsymbol{A}), \sigma_2(\boldsymbol{A}), \ldots, \sigma_{r+1}(\boldsymbol{A}), \sigma_{r+1}(\boldsymbol{A}), \ldots, \sigma_{r+1}(\boldsymbol{A})\right)
\end{aligned}
$$

Let $\boldsymbol{X}_k \coloneqq \boldsymbol{U}_{:,k:m}$ be the k-th and subsequent columns of $\boldsymbol{U}$. Similarly define $\boldsymbol{Y}_k \coloneqq \boldsymbol{V}_{:,k:n}$ and $\boldsymbol{X}_k^* \coloneqq \boldsymbol{U}\boldsymbol{S}_{:,k:m}$. Since $\boldsymbol{S}$ is a diagonal matrix, we have $\boldsymbol{X}_k^* = \boldsymbol{X}_k\boldsymbol{S}_{k:m,\,k:m}$. Rewrite f^* as:

$$
\begin{aligned}
f^* &= \boldsymbol{x}^\intercal\boldsymbol{U}\boldsymbol{S}(\boldsymbol{\Sigma}^* - \operatorname{diag}\left(\sigma_1(\boldsymbol{A}), \sigma_2(\boldsymbol{A}), \ldots, \sigma_r(\boldsymbol{A})\right))\boldsymbol{V}^\intercal\boldsymbol{y} \\
&= \boldsymbol{x}^\intercal\boldsymbol{X}_{r+1}^*\sigma_{r+1}(\boldsymbol{A})\boldsymbol{I}^{(m-r)\times(n-r)}\boldsymbol{Y}_{r+1}^\intercal\boldsymbol{y} \\
&= \sigma_{r+1}(\boldsymbol{A})\big(\boldsymbol{X}_{r+1}^{*\intercal}\boldsymbol{x}\big)^\intercal\big(\boldsymbol{I}^{(m-r)\times(n-r)}\boldsymbol{Y}_{r+1}^\intercal\boldsymbol{y}\big) \\
&\overset{①}{\le} \sigma_{r+1}(\boldsymbol{A})\big\|\boldsymbol{X}_{r+1}^{*\intercal}\boldsymbol{x}\big\|_2\big\|\boldsymbol{I}^{(m-r)\times(n-r)}\boldsymbol{Y}_{r+1}^\intercal\boldsymbol{y}\big\|_2 \\
&\le \sigma_{r+1}(\boldsymbol{A})\big\|\boldsymbol{X}_{r+1}^{*\intercal}\boldsymbol{x}\big\|_2\big\|\boldsymbol{Y}_{r+1}^\intercal\boldsymbol{y}\big\|_2 \\
&= \sigma_{r+1}(\boldsymbol{A})\big\|\boldsymbol{S}_{r+1:m,\,r+1:m}^\intercal\boldsymbol{X}_{r+1}^\intercal\boldsymbol{x}\big\|_2\big\|\boldsymbol{Y}_{r+1}^\intercal\boldsymbol{y}\big\|_2 \\
&= \sigma_{r+1}(\boldsymbol{A})\big\|\boldsymbol{X}_{r+1}^\intercal\boldsymbol{x}\big\|_2\big\|\boldsymbol{Y}_{r+1}^\intercal\boldsymbol{y}\big\|_2,
\end{aligned}
\tag{3.9}
$$

where $\overset{①}{\le}$ follows from the Cauchy-Schwarz inequality. Since $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices, (3.8) can be proven by the plugging following inequalities into (3.9):

$$
\begin{aligned}
\boldsymbol{X}_{r+1} = \mathcal{N}\big(\boldsymbol{U}_{:,1:r}^\intercal\big) = \mathcal{N}\Big(\widetilde{\boldsymbol{A}}^{(r)\,\intercal}\Big), \quad \big\|\boldsymbol{X}_{r+1}^\intercal\boldsymbol{x}\big\|_2 \le \|\boldsymbol{x}\|_2 \\
\boldsymbol{Y}_{r+1} = \mathcal{N}\big(\boldsymbol{V}_{:,1:r}^\intercal\big) = \mathcal{N}\Big(\widetilde{\boldsymbol{A}}^{(r)}\Big), \quad \big\|\boldsymbol{Y}_{r+1}^\intercal\boldsymbol{y}\big\|_2 \le \|\boldsymbol{y}\|_2
\end{aligned}
$$

□

Lemma 3.6

Given a rank-r matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and two vectors $\boldsymbol{x} \in \mathbb{R}^m$ and $\boldsymbol{y} \in \mathbb{R}^n$, the true payoff bounds satisfy:

$$
\begin{aligned}
\mathcal{P}^-(\boldsymbol{A}, r, \boldsymbol{x}, \boldsymbol{y}) &= \boldsymbol{x}^\intercal\boldsymbol{A}\boldsymbol{y} - \mathcal{U}(\boldsymbol{A}, r, \boldsymbol{x}, \boldsymbol{y}) \\
\mathcal{P}^+(\boldsymbol{A}, r, \boldsymbol{x}, \boldsymbol{y}) &= \boldsymbol{x}^\intercal\boldsymbol{A}\boldsymbol{y} + \mathcal{U}(\boldsymbol{A}, r, \boldsymbol{x}, \boldsymbol{y})
\end{aligned}
\tag{3.10}
$$

When $r < \min\{m, n\}$, for any $\epsilon \in \mathbb{R}_+$, there exist matrices $\boldsymbol{A}_\epsilon^-$ and $\boldsymbol{A}_\epsilon^+$ such that

$$
\begin{aligned}
&\operatorname{rank}(\boldsymbol{A}_\epsilon^-) = \operatorname{rank}(\boldsymbol{A}_\epsilon^+) = r+1, \quad \operatorname{LR}\left(\boldsymbol{A}_\epsilon^-, r\right) = \operatorname{LR}\left(\boldsymbol{A}_\epsilon^+, r\right) = \{\boldsymbol{A}\} \\
&\mathcal{P}^-\big(\boldsymbol{A}_\epsilon^-, r+1, \boldsymbol{x}, \boldsymbol{y}\big) = \mathcal{P}^+\big(\boldsymbol{A}_\epsilon^-, r+1, \boldsymbol{x}, \boldsymbol{y}\big) = \boldsymbol{x}^\intercal\boldsymbol{A}\boldsymbol{y} - (1-\epsilon)s \\
&\mathcal{P}^-\big(\boldsymbol{A}_\epsilon^+, r+1, \boldsymbol{x}, \boldsymbol{y}\big) = \mathcal{P}^+\big(\boldsymbol{A}_\epsilon^+, r+1, \boldsymbol{x}, \boldsymbol{y}\big) = \boldsymbol{x}^\intercal\boldsymbol{A}\boldsymbol{y} + (1-\epsilon)s \\
&\text{where } s \coloneqq \mathcal{U}(\boldsymbol{A}, r, \boldsymbol{x}, \boldsymbol{y})
\end{aligned}
\tag{3.11}
$$

Proof. Let $s \coloneqq \mathcal{U}(\boldsymbol{A}, r, \boldsymbol{x}, \boldsymbol{y})$. When $\boldsymbol{A}$ is full-rank (i.e., $r = \min\{m, n\}$), either $\mathcal{N}(\boldsymbol{A}^\intercal)$ or $\mathcal{N}(\boldsymbol{A})$ is the zero vector, which implies $s = 0$. Therefore, (3.10) holds when $r = \min\{m, n\}$.

Now assume $r < \min\{m, n\}$. Take an arbitrary $\boldsymbol{A}' \in \alpha^{-1}(\boldsymbol{A}, r)$. Since $\sigma_{r+1}(\boldsymbol{A}') \leq \sigma_r(\boldsymbol{A}') = \sigma_r(\boldsymbol{A})$, Lemma 3.5 yields $|\boldsymbol{x}^\intercal(\boldsymbol{A} - \boldsymbol{A}')\boldsymbol{y}| \leq s$, which implies

$$\mathcal{P}^-(\boldsymbol{A}, r, \boldsymbol{x}, \boldsymbol{y}) \geq \boldsymbol{x}^\intercal \boldsymbol{A}\boldsymbol{y} - s, \quad \mathcal{P}^+(\boldsymbol{A}, r, \boldsymbol{x}, \boldsymbol{y}) \leq \boldsymbol{x}^\intercal \boldsymbol{A}\boldsymbol{y} + s \tag{3.12}$$

Eqs. (3.11) and (3.12) together imply (3.10). The rest of the proof focuses on (3.11).

Assume the SVD of $\boldsymbol{A}$ is $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\intercal$ where $\Sigma_{i,i} = 0$ for $i \geq r+1$. Define $\widetilde{\boldsymbol{U}} \coloneqq \boldsymbol{U}_{:,1:r}$ and $\widetilde{\boldsymbol{V}} \coloneqq \boldsymbol{V}_{:,1:r}$ to be the left or right singular vectors of $\boldsymbol{A}$ corresponding to nonzero singular values. Decompose $\boldsymbol{x}$ as $\boldsymbol{x} = \widetilde{\boldsymbol{U}}\boldsymbol{x}_0 + \boldsymbol{x}_b$ where $\boldsymbol{x}_0 \coloneqq \widetilde{\boldsymbol{U}}^\intercal \boldsymbol{x}$ so that $\boldsymbol{x}_b$ is orthogonal to the column space of $\widetilde{\boldsymbol{U}}$ (i.e., $\widetilde{\boldsymbol{U}}^\intercal \boldsymbol{x}_b = \boldsymbol{0}$). It follows that $\boldsymbol{x}_b$ is in the column space of $\boldsymbol{U}_{:,r+1:m}$, which implies that there is an orthogonal matrix $\boldsymbol{P}$ such that $\boldsymbol{x}_b^\intercal(\boldsymbol{U}_{:,r+1:m}\boldsymbol{P}) = \begin{bmatrix}\|\boldsymbol{x}_b\|_2 & \boldsymbol{0}\end{bmatrix}$. It also holds that $\|\boldsymbol{x}_b\|_2 = \|\mathcal{N}^\intercal(\boldsymbol{A}^\intercal)\boldsymbol{x}\|_2$. Similarly, decompose $\boldsymbol{y}$ as $\boldsymbol{y} = \widetilde{\boldsymbol{V}}\boldsymbol{y}_0 + \boldsymbol{y}_b$ and find an orthogonal matrix $\boldsymbol{Q}$ such that $(\boldsymbol{V}_{:,r+1:n}\boldsymbol{Q})^\intercal \boldsymbol{y}_b = \begin{bmatrix}\|\boldsymbol{y}_b\|_2 & \boldsymbol{0}\end{bmatrix}^\intercal$.

For $q \in \mathbb{R}$, define

$$\boldsymbol{A}_q \coloneqq \begin{bmatrix}\widetilde{\boldsymbol{U}} & \boldsymbol{U}_{:,r+1:m}\boldsymbol{P}\end{bmatrix} \operatorname{diag}(\sigma_1(\boldsymbol{A}), \ldots, \sigma_r(\boldsymbol{A}), q, 0, \ldots, 0) \begin{bmatrix}\widetilde{\boldsymbol{V}} & \boldsymbol{V}_{:,r+1:n}\boldsymbol{Q}\end{bmatrix}^\intercal$$

One can verify that for $|q| < \sigma_r(\boldsymbol{A})$, the following holds:

$$\operatorname{rank}(\boldsymbol{A}_q) = r+1, \quad \operatorname{LR}(\boldsymbol{A}_q, r) = \{\boldsymbol{A}\}, \quad \mathcal{U}(\boldsymbol{A}_q, r+1, \boldsymbol{x}, \boldsymbol{y}) = 0$$
$$\boldsymbol{x}^\intercal \boldsymbol{A}_q \boldsymbol{y} = \boldsymbol{x}^\intercal \boldsymbol{A}\boldsymbol{y} + q\|\boldsymbol{x}_b\|_2\|\boldsymbol{y}_b\|_2 = \boldsymbol{x}^\intercal \boldsymbol{A}\boldsymbol{y} + q \cdot \|\mathcal{N}^\intercal(\boldsymbol{A}^\intercal)\boldsymbol{x}\|_2 \cdot \|\mathcal{N}^\intercal(\boldsymbol{A})\boldsymbol{y}\|_2$$

Therefore, (3.11) is satisfied by setting $\boldsymbol{A}_\epsilon^- = \boldsymbol{A}_{-w}$ and $\boldsymbol{A}_\epsilon^+ = \boldsymbol{A}_w$ where $w \coloneqq (1-\epsilon)\sigma_r(\boldsymbol{A})$.□

Proof (Theorem 3.3). Since $\boldsymbol{A}$ is the perceived payoff function of a player with capability level c in a limited-rank game, we have $\operatorname{rank}(\boldsymbol{A}) \leq c$. If $\operatorname{rank}(\boldsymbol{A}) = c$, then (3.10) directly implies (3.5). Otherwise when $\operatorname{rank}(\boldsymbol{A}) < c$, we have $\sigma_c(\boldsymbol{A}) = 0$ and $\alpha^{-1}(\boldsymbol{A}, c) = \{\boldsymbol{A}\}$, which gives $\mathcal{U}(\boldsymbol{A}, c, \boldsymbol{x}, \boldsymbol{y}) = 0$ and thus (3.5) still holds.

Let's proceed to show that (4.69) hold for $c_1 = c$ and $c_2 = c+1$, and therefore the limited-rank game is universally narrowly reversible. Take arbitrary values of $\boldsymbol{x} \in \Delta_m$ and $\boldsymbol{y} \in \Delta_n$.

We consider two cases: $\operatorname{rank}(\boldsymbol{A}) = c$ and $\operatorname{rank}(\boldsymbol{A}) < c$. If $\operatorname{rank}(\boldsymbol{A}) = c$, then take an arbitrary sequence $\epsilon_i \to 0$. Lemma 3.6 implies that there exists $\boldsymbol{A}_{\epsilon_i}^- \in \mathrm{N_C}(\boldsymbol{A}, c, c+1)$ satisfying (3.11), which yields

$$\begin{aligned}\inf_{\boldsymbol{A}' \in \mathrm{N_C}(\boldsymbol{A}, c, c+1)} \mathcal{P}^+(\boldsymbol{A}', c+1, \boldsymbol{x}, \boldsymbol{y}) &\leq \inf_{i \to \infty} \mathcal{P}^+\big(\boldsymbol{A}_{\epsilon_i}^-, c+1, \boldsymbol{x}, \boldsymbol{y}\big) \\ &= \inf_{i \to \infty} (\boldsymbol{x}^\intercal \boldsymbol{A}\boldsymbol{y} - (1-\epsilon_i)s) = \mathcal{P}^-(\boldsymbol{A}, c, \boldsymbol{x}, \boldsymbol{y})\end{aligned}$$

Combining the above inequality with (4.70) yields

$$\inf_{\boldsymbol{A}' \in \mathrm{N_C}(\boldsymbol{A}, c, c+1)} \mathcal{P}^+(\boldsymbol{A}', c+1, \boldsymbol{x}, \boldsymbol{y}) = \mathcal{P}^-(\boldsymbol{A}, c, \boldsymbol{x}, \boldsymbol{y})$$

Analogously, one can prove $\sup_{\boldsymbol{A}' \in \mathrm{N_C}(\boldsymbol{A}, c, c+1)} \mathcal{P}^-(\boldsymbol{A}', c+1, \boldsymbol{x}, \boldsymbol{y}) = \mathcal{P}^+(\boldsymbol{A}, c, \boldsymbol{x}, \boldsymbol{y})$.

If $\operatorname{rank}(\boldsymbol{A}) < c$, then

$$\sup_{\boldsymbol{A}' \in \mathrm{N_C}(\boldsymbol{A}, c, c+1)} \mathcal{P}^-(\boldsymbol{A}', c+1, \boldsymbol{x}, \boldsymbol{y}) = \inf_{\boldsymbol{A}' \in \mathrm{N_C}(\boldsymbol{A}, c, c+1)} \mathcal{P}^+(\boldsymbol{A}', c+1, \boldsymbol{x}, \boldsymbol{y})$$
$$= \mathcal{P}^-(\boldsymbol{A}, c, \boldsymbol{x}, \boldsymbol{y}) = \mathcal{P}^+(\boldsymbol{A}, c, \boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\intercal \boldsymbol{A} \boldsymbol{y}$$

Therefore, the limited-rank game is universally narrowly reversible. □

3.3 Properties of capability-oblivious limited-rank games

Recall that as defined in Definition 3.1, capability-oblivious players do not know that they may have limited capabilities and therefore reason as if their perceived payoff functions are the true payoff functions. This section discusses a few properties related to games where both players are capability-oblivious.

3.3.1 Expressiveness of low-rank payoff matrices

A player's strategy space is effectively reduced to one dimension when the player's capability level is 1 in a limited-rank game. Assume without loss of generality that this is the column player. The player perceives payoff matrices in the form of $\widetilde{\boldsymbol{B}}^{(1)} = \sigma_1(\boldsymbol{B}) \boldsymbol{u} \boldsymbol{v}^\intercal$. Let $s \coloneqq \operatorname{argmin}^*_i v_i$ and $t \coloneqq \operatorname{argmax}^*_i v_i$. If all entries of $\boldsymbol{u}$ are nonnegative, then $\boldsymbol{e}_t$ is the column player's dominant strategy in the perceived game. Otherwise, given the row player's strategy $\boldsymbol{x}$, the column player's best response is either $\boldsymbol{e}_s$ or $\boldsymbol{e}_t$ depending on $\operatorname{sign}(\boldsymbol{u}^\intercal \boldsymbol{x})$. Therefore, the column player's best responses are restricted to $\Delta_{\{\boldsymbol{e}_s, \boldsymbol{e}_t\}}$ in the perceived game.

By contrast, rank-2 matrices are expressive enough to encode any best response functions as shown below.

Theorem 3.7 (Expressiveness of rank-2 payoff matrices)
Given an arbitrary function $\mathrm{BR} : [\![1, m]\!] \to [\![1, n]\!]$ *representing the best response of the column player in an* $m \times n$ *bimatrix game, there is a rank-2 payoff matrix* $\boldsymbol{B} \in \mathbb{R}^{m \times n}$ *that has the same best responses:*

$$\forall 1 \le i \le m : \operatorname*{argmax}_{1 \le j \le n} B_{i,j} = \{\mathrm{BR}(i)\}$$

Analogous results hold for the row player.

Proof. Let $f(x) \coloneqq x$ and $g(x) \coloneqq -x^2$. Define their linear combination $h_p(x) \coloneqq pf(x) + (1-p)g(x)$ for $0 \le p \le 1$. We have

$$h_p(x) = px - (1-p)x^2 = -(1-p)\left(\left(x - \frac{p}{2(1-p)}\right)^2 - \frac{p^2}{4(1-p)^2}\right)$$

Therefore, $h_p(x)$ achieves its maximum at $x = \frac{p}{2(1-p)}$, or equivalently $p = \frac{2x}{2x+1}$. Define constant vectors $\boldsymbol{a} \coloneqq \begin{bmatrix} f(1) & f(2) & \cdots & f(n) \end{bmatrix}$ and $\boldsymbol{b} \coloneqq \begin{bmatrix} g(1) & g(2) & \cdots & g(n) \end{bmatrix}$. Set $\boldsymbol{B}_i = p_i \boldsymbol{a} + (1-p_i)\boldsymbol{b}$ where $p_i = \frac{2\,\mathrm{BR}(i)}{2\,\mathrm{BR}(i)+1}$. It is then easy to verify that $\operatorname{argmax}_{1 \le j \le n} B_{i,j} = \{\mathrm{BR}(i)\}$. We also have $\operatorname{rank}(\boldsymbol{B}) = 2$ since each row is a linear combination of two constant vectors. □

Although rank-2 matrices can encode any best response functions, given a true payoff matrix $\boldsymbol{B}$, its best rank-2 approximation $\widetilde{\boldsymbol{B}}^{(2)}$ as defined in Section 3.1 does not necessarily preserve the best response function induced by $\boldsymbol{B}$. A sufficient condition for the best response function to be preserved is that the payoff gap between the best and the second-best responses is large enough to survive the low-rank approximation. In this case, the pure Nash equilibrium of the true game is also a Nash equilibrium of the perceived game. The following theorem formalizes this intuition.

Theorem 3.8 (When a perceived game preserves the pure Nash equilibrium) *For $\boldsymbol{x} \in \mathbb{R}^d$, define $\gamma(\boldsymbol{x})$ as the gap between the largest and the second-largest elements of $\boldsymbol{x}$ if the largest element is unique; otherwise $\gamma(\boldsymbol{x}) = 0$. Formally,*

$$\gamma(\boldsymbol{x}) \coloneqq \begin{cases} \left(\max_{i \in [\![1,\, d]\!]} x_i\right) - \left(\max_{j \in [\![1,\, d]\!] \setminus \operatorname{argmax}_{k \in [\![1,\, d]\!]} x_k} x_j\right) & \text{if } \left|\operatorname*{argmax}_{i \in [\![1,\, d]\!]} x_i\right| = 1 \\ 0 & \text{otherwise} \end{cases}$$

Let $\textsc{Pne}(\boldsymbol{A}, \boldsymbol{B})$ denote the set of pure Nash equilibria of the game with payoff matrices $\boldsymbol{A}$ and $\boldsymbol{B}$.

In an $m \times n$ limited-rank game with true payoff matrices $(\boldsymbol{A}, \boldsymbol{B})$, if there exists an integer $r \in [\![1,\, \min\{m,\, n\} - 1]\!]$ such that

$$\min_{j \in [\![1,\, n]\!]} \gamma(\boldsymbol{A}_{:,\, j}) > \sqrt{2}\sigma_{r+1}(\boldsymbol{A}), \quad \min_{i \in [\![1,\, m]\!]} \gamma(\boldsymbol{B}_i) > \sqrt{2}\sigma_{r+1}(\boldsymbol{B})$$

Then $\textsc{Pne}(\boldsymbol{A}, \boldsymbol{B}) = \textsc{Pne}\left(\widetilde{\boldsymbol{A}}^{(c)}, \widetilde{\boldsymbol{B}}^{(c)}\right)$ for any $c \geq r$.

Proof. Consider the row player. Take an arbitrary $j \in [\![1,\, n]\!]$. Since $\sigma_{r+1}(\boldsymbol{A}) \geq 0$, we have $\gamma(\boldsymbol{A}_{:,\, j}) > 0$ and therefore the best response to $\boldsymbol{A}_{:,\, j}$ is unique. Let $i = \operatorname{argmax}^{*}_{i \in [\![1,\, m]\!]} A_{i,\, j}$. For any $i' \neq i$ and $c \geq r$, it holds that

$$\begin{aligned} \widetilde{A}^{(c)}_{i,\, j} - \widetilde{A}^{(c)}_{i',\, j} &= (\boldsymbol{e}_i - \boldsymbol{e}_{i'})^{\intercal} \widetilde{\boldsymbol{A}}^{(c)} \boldsymbol{e}_j \\ &\overset{①}{\geq} A_{i,\, j} - A_{i',\, j} - \sigma_{c+1}(\boldsymbol{A}) \cdot \|\mathcal{N}^{\intercal}(\boldsymbol{A}^{\intercal})(\boldsymbol{e}_i - \boldsymbol{e}_{i'})\|_2 \cdot \|\mathcal{N}^{\intercal}(\boldsymbol{A})\boldsymbol{e}_j\|_2 \\ &\geq A_{i,\, j} - A_{i',\, j} - \sigma_{c+1}(\boldsymbol{A}) \cdot \|\boldsymbol{e}_i - \boldsymbol{e}_{i'}\|_2 \cdot \|\boldsymbol{e}_j\|_2 \\ &= A_{i,\, j} - A_{i',\, j} - \sqrt{2}\sigma_{c+1}(\boldsymbol{A}) \geq \gamma(\boldsymbol{A}_{:,\, j}) - \sqrt{2}\sigma_{r+1}(\boldsymbol{A}) > 0, \end{aligned}$$

where $\overset{①}{\geq}$ follows from Lemma 3.5. Therefore, the row player's best response function in the true game is preserved in the perceived game. Analogous results hold for the column player. Since $(\boldsymbol{A}, \boldsymbol{B})$ and $\left(\widetilde{\boldsymbol{A}}^{(c)}, \widetilde{\boldsymbol{B}}^{(c)}\right)$ share the same best response functions, their pure Nash equilibria are the same. □

3.3.2 Quality of solutions

In an instance of a limited-rank game, a capability-oblivious player perceives an approximate payoff function. Intuitively, an approximate perception should result in some type of approximate solution to the true game. The following theorem quantifies the approximation error.

Theorem 3.9 (Approximate best response in limited-rank games) *In an instance of a limited-rank game, if $\boldsymbol{x}^*$ is the best response to $\boldsymbol{y}$ in the game perceived by the row player, then $\boldsymbol{x}^*$ is an ϵ-approximate best response to $\boldsymbol{y}$ in the true game.*

Formally, given $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, for $\widetilde{\boldsymbol{A}}^{(c_1)} \in \mathrm{LR}(\boldsymbol{A}, c_1)$ (which is not necessarily the result of a perception-compatible choice function) and $\boldsymbol{y} \in \Delta_n$, it holds that

$$\begin{aligned} \max_{\boldsymbol{x}\in\Delta_m} \boldsymbol{x}^\intercal \boldsymbol{A}\boldsymbol{y} &\le \boldsymbol{x}^{*\intercal}\boldsymbol{A}\boldsymbol{y} + \epsilon \\ \text{where } \boldsymbol{x}^* &\coloneqq \operatorname*{argmax}^*_{\boldsymbol{x}\in\Delta_m} \boldsymbol{x}^\intercal \widetilde{\boldsymbol{A}}^{(c_1)}\boldsymbol{y} \\ \epsilon &\coloneqq 2\sigma_{c_1}(\widetilde{\boldsymbol{A}}^{(c_1)}) \cdot \left\| \mathcal{N}^\intercal\left(\widetilde{\boldsymbol{A}}^{(c_1)}\right)\boldsymbol{y}\right\|_2 \end{aligned} \tag{3.13}$$

Of note, $\epsilon \le 2\sigma_{c_1}(\widetilde{\boldsymbol{A}}^{(c_1)}) = 2\sigma_{c_1}(\boldsymbol{A})$. Analogous results hold for the column player.

Proof. For any $\boldsymbol{x} \in \Delta_m$, it holds that

$$\begin{aligned} \boldsymbol{x}^\intercal\boldsymbol{A}\boldsymbol{y} &\overset{①}{\le} \boldsymbol{x}^\intercal\widetilde{\boldsymbol{A}}^{(c_1)}\boldsymbol{y} + s_1 \overset{②}{\le} \boldsymbol{x}^{*\intercal}\widetilde{\boldsymbol{A}}^{(c_1)}\boldsymbol{y} + s_1 \overset{③}{\le} \boldsymbol{x}^{*\intercal}\boldsymbol{A}\boldsymbol{y} + s_1 + s_2 \\ \text{where } s_1 &\coloneqq \sigma_{c_1}(\widetilde{\boldsymbol{A}}^{(c_1)}) \cdot \left\| \mathcal{N}^\intercal\left(\widetilde{\boldsymbol{A}}^{(c_1)\intercal}\right)\boldsymbol{x}\right\|_2 \cdot \left\| \mathcal{N}^\intercal\left(\widetilde{\boldsymbol{A}}^{(c_1)}\right)\boldsymbol{y}\right\|_2 \\ s_2 &\coloneqq \sigma_{c_1}(\widetilde{\boldsymbol{A}}^{(c_1)}) \cdot \left\| \mathcal{N}^\intercal\left(\widetilde{\boldsymbol{A}}^{(c_1)\intercal}\right)\boldsymbol{x}^*\right\|_2 \cdot \left\| \mathcal{N}^\intercal\left(\widetilde{\boldsymbol{A}}^{(c_1)}\right)\boldsymbol{y}\right\|_2 \end{aligned}$$

Here $\overset{①}{\le}$ and $\overset{③}{\le}$ follow from Lemma 3.5 and $\overset{②}{\le}$ follows from the definition of $\boldsymbol{x}^*$. Since $\mathcal{N}^\intercal(\boldsymbol{M})$ contains orthonormal basis vectors, $\|\mathcal{N}^\intercal(\boldsymbol{M})\boldsymbol{x}\|_2 \le \|\boldsymbol{x}\|_2 \le 1$ for any matrix $\boldsymbol{M}$ and $\boldsymbol{x} \in \Delta_m$. Therefore, $s_1 + s_2 \le \epsilon$. □

If both players perceive the same payoff matrices (e.g., when they have the same capability levels or if the true payoff matrices' ranks are no more than the lower capability level), then Theorem 3.9 implies that their strategy profile is an approximate Nash equilibrium of the true game. Another setting when players perceive the same payoff matrices is the uncoupled dynamics in repeated games, which is discussed in Section 3.3.3.

Furthermore, in a zero-sum game, a player can guarantee that their maximin strategy in a perceived game is an ϵ-approximate maximin strategy in the true game:

Theorem 3.10 (Approximate maximin strategy in limited-rank zero-sum games) *Given a zero-sum game $(\boldsymbol{A}, -\boldsymbol{A})$ where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, define $\boldsymbol{x}^*_c$ as the maximin strategy of the row player in the perceived game when the row player has capability level c:*

$$\boldsymbol{x}^*_c \coloneqq \operatorname*{argmax}^*_{\boldsymbol{x}\in\Delta_m} \min_{j\in[\![1,n]\!]} \boldsymbol{x}^\intercal\widetilde{\boldsymbol{A}}^{(c)}\boldsymbol{e}_j$$

Define $v \coloneqq \max_{\boldsymbol{x}\in\Delta_m}\min_{j\in[\![1,n]\!]}\boldsymbol{x}^\intercal\boldsymbol{A}\boldsymbol{e}_j$ as the value of the true game. It holds that

$$\min_{j\in[\![1,n]\!]} \boldsymbol{x}^{*\intercal}_c\boldsymbol{A}\boldsymbol{e}_j \ge v - 2\sigma_c(\widetilde{\boldsymbol{A}}^{(c)}) \tag{3.14}$$

Analogously, let $\boldsymbol{y}^*_c \coloneqq \operatorname{argmin}^*_{\boldsymbol{y}\in\Delta_n}\max_{i\in[\![1,m]\!]}\boldsymbol{e}_i^\intercal\widetilde{\boldsymbol{A}}^{(c)}\boldsymbol{y}$ be the minimax strategy of the column player in the perceived game; then

$$\max_{i\in[\![1,m]\!]} \boldsymbol{e}_i^\intercal\boldsymbol{A}\boldsymbol{y}^*_c \le v + 2\sigma_c(\widetilde{\boldsymbol{A}}^{(c)}) \tag{3.15}$$

Proof. Let $v_c \coloneqq \max_{\boldsymbol{x}\in\Delta_m} \min_{j\in[\![1,n]\!]} \boldsymbol{x}^\intercal \widetilde{\boldsymbol{A}}^{(c)} \boldsymbol{e}_j$ be the value of the perceived game when the player has capability level c. Let $\boldsymbol{x}^* \coloneqq \boldsymbol{x}^*_{\mathrm{rank}(\boldsymbol{A})}$ be the strategy of the row player in the true game.

Given c, let $j' \coloneqq \operatorname{argmin}^*_{j\in[\![1,n]\!]} \boldsymbol{x}^{*\intercal} \widetilde{\boldsymbol{A}}^{(c)} \boldsymbol{e}_j$. It holds that

$$\begin{aligned} v &= \min_{j\in[\![1,n]\!]} \boldsymbol{x}^{*\intercal} \boldsymbol{A} \boldsymbol{e}_j \le \boldsymbol{x}^{*\intercal} \boldsymbol{A} \boldsymbol{e}_{j'} \overset{①}{\le} \boldsymbol{x}^{*\intercal} \widetilde{\boldsymbol{A}}^{(c)} \boldsymbol{e}_{j'} + \sigma_c(\boldsymbol{A}) \\ &= \min_{j\in[\![1,n]\!]} \boldsymbol{x}^{*\intercal} \widetilde{\boldsymbol{A}}^{(c)} \boldsymbol{e}_j + \sigma_c(\boldsymbol{A}) \\ &\le \min_{j\in[\![1,n]\!]} \boldsymbol{x}_c^{*\intercal} \widetilde{\boldsymbol{A}}^{(c)} \boldsymbol{e}_j + \sigma_c(\boldsymbol{A}) = v_c + \sigma_c(\boldsymbol{A}), \\ \min_{j\in[\![1,n]\!]} \boldsymbol{x}_c^{*\intercal} \boldsymbol{A} \boldsymbol{e}_j &\overset{②}{\ge} \min_{j\in[\![1,n]\!]} \left(\boldsymbol{x}_c^{*\intercal} \widetilde{\boldsymbol{A}}^{(c)} \boldsymbol{e}_j - \sigma_c(\boldsymbol{A}) \right) = v_c - \sigma_c(\boldsymbol{A}), \end{aligned}$$

where $\overset{①}{\le}$ and $\overset{②}{\ge}$ both follow from Lemma 3.5 with $\|\boldsymbol{x}^*\|_2 \le 1$, $\|\boldsymbol{x}_c^*\|_2 \le 1$, and $\|\boldsymbol{e}_{j'}\|_2 = 1$. Eq. (3.14) is obtained by combining the above inequalities. Eq. (3.15) can be proved analogously. □

When players perceive different payoff matrices, even if the Nash equilibrium is unique in each perceived game, their combined strategy profile may not be a best response for either player, as shown in the following example.

Example 3.1 (Payoff change of a capability-oblivious full-capability player) Here is an example where a capability-oblivious player with full capability suffers from an arbitrarily large loss compared to their best response when the opponent has a limited capability.

Pick a constant $C \in \mathbb{R}_+$. Define the following true payoff matrices for a limited-rank game:

$$\boldsymbol{A} \coloneqq \begin{bmatrix} C & 1 \\ -1 & -1 \\ 0 & C \end{bmatrix}, \quad \boldsymbol{B} \coloneqq \begin{bmatrix} 2 & 1 \\ 1 & 5 \\ 4 & 3 \end{bmatrix}$$

The true game has a unique NE with the pure strategy profile $(\boldsymbol{e}_1, \boldsymbol{e}_1)$. The row player receives a payoff of C.

Let the capability profile be $c_1 \ge 2$ and $c_2 = 1$ in an instance of the limited-rank game. The row player still perceives the true payoff matrices. The column player perceives $\widetilde{\boldsymbol{B}}^{(1)} \approx \begin{bmatrix} 1.12 & 2.67 & 2.72 \end{bmatrix}^\intercal \begin{bmatrix} 1 & 1.43 \end{bmatrix}$ as given by Lemma 3.2. Therefore, the second column is the column player's dominant strategy in this instance. The row player still plays the first row due to the ignorance of their opponent's rank capability limitation, which results in $C - 1$ worse payoff for the row player compared to the best response (the third row). By comparison, the column player is already playing their best response of their perceived game, which is an ϵ-approximate best response in the true game with $\epsilon \le 2\sigma_1(\widetilde{\boldsymbol{B}}^{(1)}) < 13.9$.◁

3.3.3 Repeated games with uncoupled dynamics

When capability-oblivious players have different capability levels, they can perceive different games; if they play equilibrium strategies in their respectively perceived games, the combined

strategy profile may not be an equilibrium. In the zero-sum case, Theorem 3.10 provides a bound of the true payoff when a player uses the maximin strategy of the perceived game. In the general-sum case, no useful bound has been obtained so far.

This section discusses an alternative approach of modeling capability-oblivious players. I consider the uncoupled dynamics in repeated games where each player only has access to their own payoff function and all players' history, but no access to their opponents' payoff functions. This model allows players to focus on their own strategy to maximize their own payoff. In uncoupled dynamics, instead of considering the opponents' incentives by accessing the opponents' payoff functions, each player assumes that their opponents' strategies can be described by a certain model (e.g., stationary mixed strategies) and updates their beliefs of their opponents' strategies based on the observed history. In this way, players with the capability profile (c_1, c_2) and the true game $(\boldsymbol{A}, \boldsymbol{B})$ can be treated as in the stage game $\left(\widetilde{\boldsymbol{A}}^{(c_1)}, \widetilde{\boldsymbol{B}}^{(c_2)}\right)$, where previous results on uncoupled dynamics can be applied.

Uncoupled dynamics has been extensively studied under various settings [HM03; Ber05; HM06; Bab12; Say+21]. One result is that there exists no uncoupled dynamics that uses bounded length of history, is stationary (i.e., time independent), and guarantees almost sure convergence of the behavior probabilities (i.e., period-by-period behavior) to ϵ-equilibrium of the stage game for sufficiently small ϵ [HM06]. However, this impossibility result can be circumvented by introducing randomized strategy switching that breaks the stationary assumption [FY06]. Another celebrated result is that there exists a class of learning dynamics called no-regret learning that converges to correlated equilibria, with extensive research on no-regret learning algorithms under various settings [FV97; FV98; FL99; HM00; FY06; SL07; Mor+21; Far+22b; Far+22a].

I consider fictitious play [Bro51], one of the simplest uncoupled dynamics. Each player assumes that their opponent's strategy is stationary and plays their best response against their observed opponent's empirical behavior. Formally, in an $m \times n$ bimatrix game, the players start with any strategy profile $(\boldsymbol{p}_0 \in \Delta_m, \boldsymbol{q}_0 \in \Delta_n)$. At each time step $t \geq 1$, they choose a strategy $\boldsymbol{x}_t \in \mathrm{BR}_1(\boldsymbol{q}_{t-1})$ and $\boldsymbol{y}_t \in \mathrm{BR}_2(\boldsymbol{p}_{t-1})$, where $\mathrm{BR}_i(\boldsymbol{z})$ is the set of best responses of player i against the opponent's strategy $\boldsymbol{z}$. The players update their beliefs of their opponent's strategy by $\boldsymbol{p}_t = \frac{1}{t}((t-1)\boldsymbol{p}_{t-1} + \boldsymbol{x}_t)$ and $\boldsymbol{q}_t = \frac{1}{t}((t-1)\boldsymbol{q}_{t-1} + \boldsymbol{y}_t)$.

Fictitious play converges to a Nash equilibrium in specific cases including zero-sum games $(\boldsymbol{A} = -\boldsymbol{B})$ [Rob51], games with identical utility functions $(\boldsymbol{A} = \boldsymbol{B})$ [MS96b], potential games [MS96c], and $2 \times n$ bimatrix games [Ber05], assuming some "natural" tie-breaking rule (e.g., never change a strategy if it is already a best response) [MS96a]. Convergence is defined by convergence of players' strategy beliefs to some Nash equilibrium. There are 3×3 games where fictitious play does not converge [Sha64].

As discussed in Section 3.3.1, a player's strategy space is effectively reduced to two strategies when the player's capability level is 1. Since fictitious play converges for $2 \times n$ bimatrix games [Ber05], we have the following result:

Theorem 3.11

In a repeated bimatrix game with true payoff matrices $(\boldsymbol{A}, \boldsymbol{B})$, if the stage game is a limited-rank game with players' capability levels c_1 and c_2, both players are capability-oblivious, and

$\min\{c_1, c_2\} = 1$, *then fictitious play converges to an* ϵ*-Nash of the true game, where*

$$\epsilon = 2\max\left\{\sigma_{c_1}(\widetilde{\boldsymbol{A}}^{(c_1)}),\, \sigma_{c_2}(\widetilde{\boldsymbol{B}}^{(c_2)})\right\}$$

On the other hand, if both players' capability levels are least 2, then there are games where fictitious play does not converge, as shown in the following example.

Example 3.2 (A rank-2 game where fictitious play does not converge)
Consider the 3×3 game $G = (\boldsymbol{A}, \boldsymbol{B})$ where:

$$\boldsymbol{A} \coloneqq \begin{bmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 1 & 2 \end{bmatrix}, \quad \boldsymbol{B} \coloneqq \begin{bmatrix} 1 & 0 & 2 \\ 2 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

Shapley [Sha64, Section 5.2] shows that fictitious play does not converge in G. We derive a rank-2 game $G' = (\boldsymbol{A}', \boldsymbol{B}')$ with the same best responses as G for all strategy profiles by subtracting the first row of $\boldsymbol{A}$ from each row of $\boldsymbol{A}$ and the first column of $\boldsymbol{B}$ from each column of $\boldsymbol{B}$:

$$\boldsymbol{A}' \coloneqq \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ -2 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{B}' \coloneqq \begin{bmatrix} 0 & -1 & 1 \\ 0 & -1 & -2 \\ 0 & 2 & 1 \end{bmatrix}$$

It is easy to verify that $\operatorname{rank}(\boldsymbol{A}') = \operatorname{rank}(\boldsymbol{B}') = 2$, $\boldsymbol{A}'\boldsymbol{y} = \boldsymbol{A}\boldsymbol{y} + c_1$, and $\boldsymbol{x}^\intercal \boldsymbol{B}' = \boldsymbol{x}^\intercal \boldsymbol{B} + c_2$ for constants $c_1 \in \mathbb{R}$ and $c_2 \in \mathbb{R}$ and all $\boldsymbol{x} \in \Delta_3$ and $\boldsymbol{y} \in \Delta_3$. Thus G and G' have identical paths of fictitious play, which implies fictitious play does not converge for the game G'. ◁

3.4 Capability transfer properties in limited-rank games

One crucial question related to the competitiveness of low-rank agents is how the game outcome changes when a player has different capability levels. This section considers capability-oblivious players in limited-rank games. One of the players has a fixed capability level. The goal is to compare the game outcomes when the other player has different capability levels.

A challenge in formalizing the change of payoff is that when there are multiple Nash equilibria, an equilibrium selection rule is needed; otherwise the payoff changes are coupled with contributions from playing mismatched strategies of different equilibria. To address the equilibrium selection issue, this section assumes both players agree to use the *simplest equilibrium*, which is the strategy profile with minimum support size and ties broken with lexicographical order of the support and lexicographical order of the probability values, as formalized below.

Definition 3.4 (Simplest equilibrium)
Let $\textsc{Ne}$ be the set of all Nash equilibria of a given $m \times n$ bimatrix game G. Define $f : \textsc{Ne} \mapsto V$ with $V \coloneqq \mathbb{N} \times \mathbb{N} \times \mathbb{N}^{\max\{m,n\}} \times \Delta_m \times \Delta_n$ that maps a Nash equilibrium to a tuple for lexicographical comparison:

$$f(\boldsymbol{x}, \boldsymbol{y}) \coloneqq (a+b,\, |a-b|,\, \operatorname{maxv}(\operatorname{supp}^{\mathrm{v}}(\boldsymbol{x}), \operatorname{supp}^{\mathrm{v}}(\boldsymbol{y})),\, \boldsymbol{x},\, \boldsymbol{y}),$$
$$\text{where } a \coloneqq |\operatorname{supp}(\boldsymbol{x})|, \quad b \coloneqq |\operatorname{supp}(\boldsymbol{y})|$$

Here $\operatorname{maxv}(\boldsymbol{x}, \boldsymbol{y})$ returns the element-wise maximum of $\boldsymbol{x}$ and $\boldsymbol{y}$ with $-\infty$ appended to the shorter vector to make them the same length.

Define a total order $<$ on V as the lexicographical order (with recursion into $\mathbb{N}^k$ and Δ_k). Clearly $f(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{x}', \boldsymbol{y}')$ if and only if $\boldsymbol{x} = \boldsymbol{x}'$ and $\boldsymbol{y} = \boldsymbol{y}'$.

Since $\textsc{Ne}$ is compact, there is a unique minimum $(\boldsymbol{x}^*, \boldsymbol{y}^*) \in \textsc{Ne}$ such that $f(\boldsymbol{x}^*, \boldsymbol{y}^*) \leq f(\boldsymbol{x}, \boldsymbol{y})$ for all $(\boldsymbol{x}, \boldsymbol{y}) \in \textsc{Ne}$. The strategy profile $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is the *simplest equilibrium* of the game G. ◁

Remark 3.1 The total order is symmetric with respect to the row and column players (i.e., if $f(\boldsymbol{x}, \boldsymbol{y}) < f(\boldsymbol{x}', \boldsymbol{y}')$, then $f(\boldsymbol{y}, \boldsymbol{x}) < f(\boldsymbol{y}', \boldsymbol{x}'))$ unless we need to break ties by the numerical values of $\boldsymbol{x}$ and $\boldsymbol{y}$.

Since the size of the support of a Nash equilibrium can be bounded by the rank of the payoff matrix [LMM03], there is a chance to find the same simplest equilibrium when players have different capability levels. In particular, if the true game has a pure Nash equilibrium and the payoff matrices satisfy the conditions of Theorem 3.8, then the same pure Nash equilibrium is selected by any limited-rank player of capability level at least r. Of note, finding the simplest equilibrium for a given game is NP-hard [CMG17].

The change of payoff is formalized as follows.

Definition 3.5 (Change of payoff)
In a rank-limited game, assume both players are capability-oblivious and play the simplest equilibrium of their respective perceived games.

Assume without loss of generality that the row player has different capability levels c_1^1 and c_1^2, and the column player has capability level c_2. Let $(\boldsymbol{x}_1, \boldsymbol{y})$ be the strategy profile in the instance with the capability profile (c_1^1, c_2), and $(\boldsymbol{x}_2, \boldsymbol{y})$ be the strategy profile in the instance with the capability profile (c_1^2, c_2).

Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be the true payoff matrices. The change of payoff for player i is denoted by δ_{ui} and is defined as:

$$\delta_{u1} \coloneqq \boldsymbol{x}_2^\intercal \boldsymbol{A}\boldsymbol{y} - \boldsymbol{x}_1^\intercal \boldsymbol{A}\boldsymbol{y}, \quad \delta_{u2} \coloneqq \boldsymbol{x}_2^\intercal \boldsymbol{B}\boldsymbol{y} - \boldsymbol{x}_1^\intercal \boldsymbol{B}\boldsymbol{y}$$

Moreover, define $\delta_W \coloneqq \delta_{u1} + \delta_{u2}$ as the change of social welfare.

The symbols δ_{u1}, δ_{u2}, and δ_W are used when the values of c_1^1, c_1^2, and c_2 are clear from the context.

The change of payoff is defined analogously when the row player has a fixed capability level c_1 and the column player has different capability levels c_2^1 and c_2^2. ◁

3.4.1 Hardness of determining the sign of the change of payoff

Computing the numerical values of δ_{u1} and δ_{u2} defined in Definition 3.5 seems not easier than solving the game. In practice, one may be interested in determining $\operatorname{sign}(\delta_{u1})$ and $\operatorname{sign}(\delta_{u2})$ with the hope that the sign of the change of payoff is easier to determine than the numerical value. This section shows that determining the sign is as hard as computing a Nash equilibrium.

The sign of a payoff change or social welfare change is called the *payoff change type* for short. Define the *payoff change type function* $\mathcal{T}(\boldsymbol{A}, \boldsymbol{B}, c_1^1, c_1^2, c_2) = (\operatorname{sign}(\delta_{u1}), \operatorname{sign}(\delta_{u2}))$

that determines the payoff change types for both players, where the row player has capability levels c_1^1 or c_1^2, and the column player has a fixed capability level c_2. Analogously define $\mathcal{T}(\boldsymbol{A}, \boldsymbol{B}, c_1, c_2^1, c_2^2)$ when the row player has a fixed capability level c_1 and the column player has capability levels c_2^1 or c_2^2. This section proves the following result:

Theorem 3.12 (Hardness of determining payoff change types)
Computing the payoff change type function $\mathcal{T}(\cdot)$ is PPAD*-hard.*

More specifically, for any matrices $\boldsymbol{P}, \boldsymbol{Q} \in \mathbb{R}^{r\times s}$, define $\mathcal{T}'(\boldsymbol{P}, \boldsymbol{Q}) \coloneqq \operatorname{sign}(\delta_{u2})$ with $\operatorname{sign}(\delta_{u2})$ given by $\mathcal{T}(\boldsymbol{P}, \boldsymbol{Q}, c_1 = \min\{r, s\}, c_2^1 = \min\{r, s\}, c_2^2 = 1)$. Then given an arbitrary $m \times n$ bimatrix game $(\boldsymbol{A}, \boldsymbol{B})$, a Nash equilibrium of this game can be found by making $m + n$ queries to $\mathcal{T}'(\cdot)$ and then solving a linear program with $\mathcal{O}\big(\max\{m, n\}^2\big)$ constraints and $\mathcal{O}(m + n)$ variables.

Of note, the value of $\mathcal{T}'(\boldsymbol{A}, \boldsymbol{B})$ defined Theorem 3.12 is non-positive since the column player plays a best response to the row player's strategy in the instance (c_1, c_2^1). The hardness is essentially about deciding if a player has strictly better payoff when they have full rank compared to when they have the lowest rank against a full-rank opponent.

The proof is based on constructing a payoff matrix such that the best response of its rank-1 approximation can be manipulated. Then by querying the sign of payoff change of a player in full-rank and rank-1 instances, one can determine the support of the simplest equilibrium of the true game.

Lemma 3.13
Let $a \in \mathbb{R}_+$ and $x, y \in \mathbb{R}$. If $x \le y$, then $(a - x)^2 + y^2 \ge \frac{a^2}{2}$.

Proof. If $x < 0$, then $(a - x)^2 + y^2 \ge (a - 0)^2 \ge \frac{a^2}{2}$. If $0 \le x \le y$, then $(a - x)^2 + y^2 \ge (a - x)^2 + x^2 = 2\big(x - \frac{a}{2}\big)^2 + \frac{a^2}{2} \ge \frac{a^2}{2}$. □

Lemma 3.14
Given $a \in \mathbb{R}_+$, $B \in \mathbb{R}^{m\times n}$, and $k \in [\![1, n]\!]$, define the matrix $\boldsymbol{M} \coloneqq \begin{bmatrix} a\boldsymbol{e}_k^\intercal \\ \boldsymbol{B} \end{bmatrix}$. Let $\boldsymbol{P} \in \operatorname{LR}(\boldsymbol{M}, 1)$ be a rank-1 approximation of $\boldsymbol{M}$. If $a > \sqrt{2}\|\boldsymbol{B}\|_F$, then $P_{1,k} > P_{1,i}$ for all $i \ne k$.

Proof. Prove the contraposition. Assume there exists $i \ne k$ such that $P_{1,k} \le P_{1,i}$. By Lemma 3.13,

$$\|\boldsymbol{M} - \boldsymbol{P}\|_F \ge \sqrt{(a - P_{1,k})^2 + P_{1,i}^2} \ge \frac{a}{\sqrt{2}}$$

Define $\boldsymbol{P}' \coloneqq \begin{bmatrix} a\boldsymbol{e}_k^\intercal \\ \boldsymbol{0} \end{bmatrix}$. Then $\|\boldsymbol{M} - \boldsymbol{P}'\|_F = \|\boldsymbol{B}\|_F$. By the definition of the low-rank approximation, $\|\boldsymbol{M} - \boldsymbol{P}\|_F \le \|\boldsymbol{M} - \boldsymbol{P}'\|_F$, which implies $a \le \sqrt{2}\|\boldsymbol{B}\|_F$. □

Lemma 3.15
Given an $m \times n$ bimatrix game $(\boldsymbol{A}, \boldsymbol{B})$, let $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ be its simplest equilibrium. Then the best response set of the column player, i.e., $\operatorname{argmax}_{j\in[\![1,n]\!]} \boldsymbol{x}^{*\intercal}\boldsymbol{B}\boldsymbol{e}_j$, can be computed by making n queries to the payoff change sign function $\mathcal{T}'(\cdot)$. Analogous results hold for the row player.

Proof. Define the matrices $\boldsymbol{A}^+ \coloneqq \boldsymbol{A} - \min \boldsymbol{A} + 1$ and $\boldsymbol{B}^+ \coloneqq \boldsymbol{B} - \min \boldsymbol{B} + 1$, where $\min \boldsymbol{M} \coloneqq \min_{i \in [\![1,\, m]\!],\, j \in [\![1,\, n]\!]} M_{i,\, j}$ denotes the minimum element of the matrix $\boldsymbol{M}$. It is easy to verify that $(\boldsymbol{x}^*,\, \boldsymbol{y}^*)$ is the simplest equilibrium of the game $(\boldsymbol{A}^+,\, \boldsymbol{B}^+)$.

For $k \in [\![2,\, n+1]\!]$, define the following matrices

$$\boldsymbol{A}' \coloneqq \begin{bmatrix} C & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}^+ \end{bmatrix}, \quad \boldsymbol{B}^k \coloneqq \begin{bmatrix} D\boldsymbol{e}_k^\intercal \\ \boldsymbol{0} \quad \boldsymbol{B}^+ \end{bmatrix}$$
$$\text{where } C > \sigma_1(\boldsymbol{A}^+), \quad D > \sqrt{2} \left\| \boldsymbol{B}^+ \right\|_F$$

Then the rank-1 approximation of $\boldsymbol{A}'$ is $\widetilde{\boldsymbol{A}'}^{(1)} = \begin{bmatrix} C & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$. Lemma 3.14 implies $\widetilde{B^k}^{(1)}_{1,\, k} > \widetilde{B^k}^{(1)}_{1,\, i}$ for $i \neq k$. Recall that the simplest equilibrium prefers pure strategies and strategies that use the smallest-numbered actions. In the game $\left(\widetilde{\boldsymbol{A}'}^{(1)},\, \widetilde{\boldsymbol{B}^k}^{(1)}\right)$, since there is no equilibrium in the form of $(\boldsymbol{x},\, \boldsymbol{e}_1)$, the simplest equilibrium is $(\boldsymbol{e}_1,\, \boldsymbol{e}_k)$.

Let $\left(\boldsymbol{x}^k,\, \boldsymbol{y}^k\right)$ be the simplest equilibrium of the game $\left(\boldsymbol{A}',\, \boldsymbol{B}^k\right)$. Since the first column of $\boldsymbol{B}^k$ is all zero, which is strictly dominated by at least one other column, the first column can not be a best response, i.e., $y_1^k = 0$. Let $\boldsymbol{y}^k_{2:n+1}$ be the all but the first element of $\boldsymbol{y}^k$. Then

$$\boldsymbol{A}'\boldsymbol{y}^k = \begin{bmatrix} 0 \\ \boldsymbol{A}^+\boldsymbol{y}^k_{2:n+1} \end{bmatrix} \qquad \text{with } \boldsymbol{y}^k_{2:n+1} \in \Delta_n$$

Since $\min \boldsymbol{A}^+ = 1$, the first row can not be a best response, i.e., $x_1^k = 0$. Therefore, $\left(\boldsymbol{x}^k,\, \boldsymbol{y}^k\right)$ corresponds to the simplest equilibrium of the game $(\boldsymbol{A}^+,\, \boldsymbol{B}^+)$, and it follows that $\boldsymbol{x}^* = \boldsymbol{x}^k_{2:m+1}$ and $\boldsymbol{y}^* = \boldsymbol{y}^k_{2:n+1}$.

Consider two instances of the limited-rank game with true payoff matrices $\left(\boldsymbol{A}',\, \boldsymbol{B}^k\right)$ where the row player has the fixed capability level $c_1 = \min\{m,\, n\}$ and the column player has the capability levels $c_2^1 = \min\{m,\, n\}$ and $c_2^2 = 1$ in the first and second instances, respectively. The player's strategy profiles are $\left(\boldsymbol{x}^k,\, \boldsymbol{y}^k\right)$ and $\left(\boldsymbol{x}^k,\, \boldsymbol{e}_k\right)$ in the first and second instances, respectively. We have $\delta_{u2} = \boldsymbol{x}^{k\intercal}\boldsymbol{B}^k\left(\boldsymbol{e}_k - \boldsymbol{y}^k\right) = \boldsymbol{x}^{*\intercal}\boldsymbol{B}^+(\boldsymbol{e}_{k-1} - \boldsymbol{y}^*)$, which implies $\operatorname{sign}(\delta_{u2}) = 0$ if and only if $\boldsymbol{e}_{k-1}$ is a best response to $\boldsymbol{x}^*$ in the original game $(\boldsymbol{A},\, \boldsymbol{B})$. Therefore, for $k \in [\![1,\, n]\!]$, one can decide whether $k \in \operatorname{argmax}_{j \in [\![1,\, n]\!]} \boldsymbol{x}^{*\intercal}\boldsymbol{B}\boldsymbol{e}_j$ by querying $\operatorname{sign}(\delta_{u2})$ via $\mathcal{T}\left(\boldsymbol{A}',\, \boldsymbol{B}^{k+1},\, c_1,\, c_2^1,\, c_2^2\right)$. □

Proof (Theorem 3.12). With Lemma 3.15, one can determine the best response sets $P \coloneqq \operatorname{argmax}_{i \in [\![1,\, m]\!]} \boldsymbol{e}_i^\intercal \boldsymbol{A}\boldsymbol{y}^*$ $Q \coloneqq \operatorname{argmax}_{j \in [\![1,\, n]\!]} \boldsymbol{x}^{*\intercal}\boldsymbol{B}\boldsymbol{e}_j$ by making $m+n$ queries to $\mathcal{T}(\cdot)$. Consider the following polytope defined for $\boldsymbol{x} \in \Delta_m$ and $\boldsymbol{y} \in \Delta_n$:

$$\begin{aligned}
x_i = 0,\, y_j = 0 \qquad & \forall i \notin P, \quad \forall j \notin Q \\
x_i \geq 0,\, y_j \geq 0 \qquad & \forall i \in P, \quad \forall j \in Q \\
\sum_{i \in P} x_i = 1,\, \sum_{j \in Q} y_j = 1 \qquad & \\
\boldsymbol{A}^\intercal_{:,\, j}\boldsymbol{x} \geq \boldsymbol{A}^\intercal_{:,\, j'}\boldsymbol{x} \qquad & \forall j \in Q, \quad \forall j' \notin Q \\
\boldsymbol{B}_i\boldsymbol{y} \geq \boldsymbol{B}_{i'}\boldsymbol{y} \qquad & \forall i \in P, \quad \forall i' \notin P
\end{aligned} \tag{3.16}$$

Any solution to (3.16) is a Nash equilibrium of the game $(\boldsymbol{A}, \boldsymbol{B})$. Since the simplest equilibrium of $(\boldsymbol{A}, \boldsymbol{B})$ is also a solution to (3.16), the polytope is nonempty. Therefore, one Nash equilibrium of the game $(\boldsymbol{A}, \boldsymbol{B})$ can be found by finding one feasible solution to (3.16) using linear programming in polynomial time. □

3.4.2 Analytical results of a defense game

The hardness result of Theorem 3.12 suggests that there is unlikely to be a useful closed-form expression for the payoff change type. This section analyzes a simple defense game that exhibits rich behavior of payoff change types. The goal is to characterize whether or when one player benefits from limiting themselves to a lower capability level if their opponent has a fixed non-full capability level, assuming both players are capability-oblivious.

The defense game is a $n \times 2$ bimatrix game. The row player is the defender. The column player is the attacker. The defender tries to defend a fixed-value target by selecting a nonnegative integer as the defense level. The attacker can choose to attack the target or not. A fraction (depending on the defense level) of the target value is transferred to the attacker if they choose to attack. The defender's payoff is determined by the cost of defense (linear in the defense level) and the target value remained after the transfer. The attacker's payoff is is transferred target value (if they choose to attack) minus the cost of attacking (linear in the defense level).

Formally, the game is parameterized by a quintuple (C, n, k_1, k_2, f) where $C \in \mathbb{R}_+$ is the value of the target, $n \in \mathbb{Z}_+$ is the number of available defense levels, $k_1 \in \mathbb{R}_+$ is the cost to the defender for one more defense level, $k_2 \in \mathbb{R}_+$ is cost to the attacker for one more defense level, and $f : \mathbb{R}_{\geq 0} \mapsto [0, 1]$ is the fraction of target value transferred to the attacker for each defense level, satisfying $f(0) = 0$, $f'(x) \geq 0$ and $f''(x) \leq 0$ for $x \in \mathbb{R}_{\geq 0}$, $f'(x) > 0$ and $f''(x) < 0$ if $f(x) < 1$, and $\lim_{x \to +\infty} f(x) = 1$. The function f models decreasing marginal returns to defense investment. An example of f is $f(x) = 1 - \frac{1}{\alpha x + 1}$ for some $\alpha \in \mathbb{R}_+$. The strategy space of the defender is $[\![0, n-1]\!]$. The strategy space of the attacker is $\{0, 1\}$. Let $d \in [\![0, n-1]\!]$ be the defense level chosen by the defender, $a \in \{0, 1\}$ be whether the attacker chooses to attack $(a = 1)$ or not $(a = 0)$, and $\mathrm{U}_1(d, a)$ and $\mathrm{U}_2(d, a)$ be the payoff functions of the defender and attacker, respectively. Define:

$$\begin{aligned} &\mathrm{U}_1(d, 0) \coloneqq -dk_1 + C, \quad &&\mathrm{U}_1(d, 1) \coloneqq -dk_1 + Cf(d) \\ &\mathrm{U}_2(d, 0) \coloneqq 0, &&\mathrm{U}_2(d, 1) \coloneqq -dk_2 + C(1 - f(d)) \end{aligned} \tag{3.17}$$

Assume different defense levels result in different payoffs, i.e., $\mathrm{U}_1(d_1, 1) \neq \mathrm{U}_1(d_2, 1)$ and $\mathrm{U}_2(d_1, 1) \neq \mathrm{U}_2(d_2, 1) \neq 0$ for $d_1 \neq d_2$. If the attacker does not attack, the defender should choose the lowest defense level, resulting in a defender payoff of C and attacker payoff of 0. The attacker is then incentivized to attack. Therefore, the pure strategy $a = 0$ can not be a Nash equilibrium.

The defender needs to choose a defense level to maximize their payoff when the attacker attacks. Since $\frac{\partial \mathrm{U}_1(d,1)}{\partial d} = Cf'(d) - k_1$, the payoff function $\mathrm{U}_1(d, 1)$ can be monotonically increasing, monotonically decreasing, or increasing followed by decreasing. Let $d^* = \operatorname{argmax}^*_{d \in [\![0, n-1]\!]} \mathrm{U}_1(d, 1)$ be the optimal defense level when the attacker attacks. If $\mathrm{U}_2(d^*, 1) \geq 0$, then $(d = d^*, a = 1)$ is the unique Nash equilibrium in this game.

This paragraph shows that if $\mathrm{U}_2(d^*, 1) < 0$, then the game has a unique mixed Nash equilibrium. Assume the attacker's strategy is $\boldsymbol{s_2} = \begin{bmatrix} 1-q \\ q \end{bmatrix}$. Use $\mathrm{U}_1(d, \boldsymbol{s_2}) = -dk_1 + C - q(1 - f(d))C$ to represent the defender's expected payoff, which is still a unimodal function. Since $\mathrm{U}_2(n-1, 1) \leq \mathrm{U}_2(d^*, 1) < 0$, the attacker should not incentivize the defender to choose the highest defense level. Therefore, the attacker adjusts q so that there is $d_q^* \in [\![0, n-2]\!]$ such that $\mathrm{U}_1\left(d_q^*, \boldsymbol{s_2}\right) = \mathrm{U}_1\left(d_q^* + 1, \boldsymbol{s_2}\right) = \max_{d \in [\![0, n-1]\!]} \mathrm{U}_1(d, \boldsymbol{s_2})$. The defender plays a mixed strategy that selects d_q^* with probability $1 - p$ and $d_q^* + 1$ with probability p. The defender needs to satisfy $(1-p)\,\mathrm{U}_2\left(d_q^*, \boldsymbol{s_2}\right) + p\,\mathrm{U}_1\left(d_q^* + 1, \boldsymbol{s_2}\right) = 0$, which requires $\mathrm{U}_2\left(d_q^*, \boldsymbol{s_2}\right) > 0$ and $\mathrm{U}_2\left(d_q^* + 1, \boldsymbol{s_2}\right) < 0$. Since $\mathrm{U}_2(d^*, 1) < 0 < \mathrm{U}_2(0, 1) = C$, there is a unique value $\hat{d}$ such that $\mathrm{U}_2\left(\hat{d} + 1, 1\right) < 0 < \mathrm{U}_2\left(\hat{d}, 1\right)$. The attacker's strategy is $q = \frac{k_1}{(f(\hat{d}+1) - f(\hat{d}))C}$. The attacker's expected payoff is $\mu_2 = 0$. The defender's expected payoff is $\mu_1 = C + \frac{(\hat{d}+1)f(\hat{d}) - \hat{d}f(\hat{d}+1) - 1}{(f(\hat{d}+1) - f(\hat{d}))}k_1$. Note that $\mu_1 > 0$ since $\hat{d} < d^*$ and thus $\mathrm{U}_1\left(\hat{d}, 0\right) \geq \mathrm{U}_1\left(\hat{d}, 1\right) \geq \mathrm{U}_1(0, 1) > 0$.

Define the *two-level defense game* as a special case of the defense game where the defender can only choose from two levels (i.e., to defend or not to defend). The two-level defense game can be solved analytically in the limited-rank setting. Assume one of the players has a fixed capability level of 1. The following theorem fully characterizes the payoff change types in the two-level defense game when the other player has capability levels 2 or 1.

Theorem 3.16 (Payoff change types in the two-level defense game)
In the two-level defense game $(C, n = 2, k_1, k_2, f)$, assume the players are rank-limited and capability-oblivious. Let c_1 and c_2 be the capability levels of the defender and attacker, respectively. Then:

- *When $c_1 = 1$, the payoff change types for $c_2^1 = 2$ and $c_2^2 = 1$ are:*
  - $(\delta_{u1} \geq 0, \delta_{u2} = 0)$
  - $(\delta_{u1} > 0, \delta_{u2} > 0)$
  - $(\delta_{u1} < 0, \delta_{u2} > 0, \delta_W = 0)$
- *When $c_2 = 1$, the payoff change types for $c_1^1 = 2$ and $c_1^2 = 1$ are:*
  - $(\delta_{u1} \leq 0, \delta_{u2} \leq 0)$
  - $(\delta_{u1} < 0, \delta_{u2} > 0, \delta_W > 0)$

Remark 3.2 Theorem 3.16 considers all games with C, k_1, k_2, and f being the free parameters. The tuples describing payoff change types are complete and sound characterizations of possible outcomes given all possible values of C, k_1, k_2, and f. For example, when $c_1 = 1$, there is a choice of parameter such that $\delta_{u1} > 0$ and $\delta_{u2} = 0$, but there is no game such that $\delta_{u1} = 0$ and $\delta_{u2} > 0$.

Theorem 3.16 shows that it is sometimes beneficial to limit oneself to a lower capability level if their opponent has a fixed non-full capability level. In particular, when the defender

is rank-1, a rank-1 attacker will always get no worse payoff than a rank-2 attacker (i.e., $\delta_{u2} \geq 0$ when $c_1 = 1$), and the social welfare with both players being rank-1 is no worse than that when defender is rank-1 and attacker is rank-2. When the attacker is rank-1, a rank-1 defender will not get better payoff than a rank-2 defender, but the social welfare increases when both players are rank-1 compared to a rank-2 defender and a rank-1 attacker in certain games.

Proof (Theorem 3.16). This proof uses the following notation:

- $\boldsymbol{\mu}_{ij} \in \mathbb{R}^2$ denotes the expected true payoffs of the defender and attacker, respectively, when the defender's capability level is i and the attacker's capability level is j. The goal is to characterize the signs of entries of $\boldsymbol{\mu}_{11} - \boldsymbol{\mu}_{21}$ and $\boldsymbol{\mu}_{11} - \boldsymbol{\mu}_{12}$.
- $(d,\, a)$ denotes the strategy profile where the defender chooses a defense level $d \in \{0,\, 1\}$ and the attacker chooses to attack ($a = 1$) or not ($a = 0$). $(d \sim B(p),\, a)$ denotes a mixed strategy where the defender chooses $d = 1$ with probability p and $d = 0$ with probability $1 - p$. Similarly for $(d,\, a \sim B(q))$.
- Since the action space of both players are $\{0,\, 1\}$, this proof uses zero-based indexing for matrices and vectors. For example, $\boldsymbol{A}_0$ and $\boldsymbol{A}_1$ are the first and second rows of the matrix $\boldsymbol{A}$, respectively.

The following 2×2 matrices describe the payoff functions defined by (3.17) in a two-level defense game:

$$\begin{aligned} \boldsymbol{A} &= \begin{bmatrix} C & 0 \\ C - k_1 & C\beta - k_1 \end{bmatrix} \\ \boldsymbol{B} &= \begin{bmatrix} 0 & C \\ 0 & C(1-\beta) - k_2 \end{bmatrix} \\ &\text{where } \beta \coloneqq f(1) \in (0,\, 1) \end{aligned}$$

Since $\operatorname{rank}(\boldsymbol{B}) = 1$, when either player has a limited rank, they make their strategy decisions based on the perceived rank-1 matrices $\left(\widetilde{\boldsymbol{A}}^{(1)},\, \boldsymbol{B}\right)$. The proof thus focuses on the rank-1 approximation of $\boldsymbol{A}$.

Applying Lemma 3.2 to the rows of $\boldsymbol{A}$ (i.e., columns of $\boldsymbol{A}^\intercal$) yields:

$$Q_{\boldsymbol{A}^\intercal} = \boldsymbol{A}_0 \cdot \boldsymbol{A}_1^\intercal = C(C - k_1) \tag{3.18}$$

$$\begin{aligned} R_{\boldsymbol{A}^\intercal} &= \boldsymbol{A}_0 \cdot \boldsymbol{A}_0^\intercal - \boldsymbol{A}_1 \cdot \boldsymbol{A}_1^\intercal \\ &= C^2 - (C - k_1)^2 - (C\beta - k_1)^2 \\ &= -C^2 \left(\left(\frac{k_1}{C} - 1 \right)^2 + \left(\beta - \frac{k_1}{C} \right)^2 - 1 \right) \end{aligned} \tag{3.19}$$

Consider the following cases based on the signs of $Q_{\boldsymbol{A}^\intercal}$ and $R_{\boldsymbol{A}^\intercal}$.

Case 1: $C \leq k_1$

The condition is equivalent to $Q_{\boldsymbol{A}^\intercal} \leq 0$, which also implies $A_{1,1} = C\beta - k_1 < 0$. Consider the column-based representation of $\widetilde{\boldsymbol{A}}^{(1)}$ with $\widetilde{\boldsymbol{A}}^{(1)} = \begin{bmatrix} \boldsymbol{v} & t\boldsymbol{v} \end{bmatrix}$ where $\boldsymbol{v} \coloneqq (\boldsymbol{A}_{:,0} + t\boldsymbol{A}_{:,1})/(t^2+1)$. Since $Q_{\boldsymbol{A}} = (C-k_1)(C\beta - k_1) \geq 0$, Lemma 3.2 implies that $t \geq 0$, and consequently $v_1 \leq 0 < v_0$. Therefore, the perceived game $\left(\widetilde{A}^{(1)}, B\right)$ has a unique pure equilibrium $(d=0,\, a=1)$, which is the same strategy profile as the true game. Both players' payoffs are unchanged in this case.

Case 2: $C > k_1$

The condition is equivalent to $Q_{\boldsymbol{A}^\intercal} > 0$. Represent $\widetilde{\boldsymbol{A}}^{(1)}$ as:

$$\widetilde{\boldsymbol{A}}^{(1)} = \begin{bmatrix} v_0 & v_1 \\ tv_0 & tv_1 \end{bmatrix} \tag{3.20}$$

Lemma 3.2 implies that $t > 0$ given $Q_{\boldsymbol{A}^\intercal} > 0$. Further consider two cases based on the sign of $A_{1,1} = C\beta - k_1$.

Case 2.1: $C\beta \leq k_1 < C$

The condition is equivalent to $\beta \leq \frac{k_1}{C} < 1$. Since $a^2 + b^2 \leq (a+b)^2$ when $\min\{a,\, b\} \geq 0$, the right hand side of (3.19) can be bounded by:

$$\begin{aligned} \left(\frac{k_1}{C} - 1\right)^2 + \left(\beta - \frac{k_1}{C}\right)^2 - 1 &\leq \left(1 - \frac{k_1}{C} + \frac{k_1}{C} - \beta\right)^2 - 1 \\ &= (1-\beta)^2 - 1 < 0 \end{aligned}$$

We now have $Q_{\boldsymbol{A}^\intercal} > 0$ and $R_{\boldsymbol{A}^\intercal} > 0$, which implies $0 < t < 1$, $v_0 > 0$, and $v_1 < 0$ in (3.20). Therefore, the defender's best response in the perceived game is to choose $d = 0$ if the attacker does not attack and choose $d = 1$ otherwise. In the true game, the defender has the dominant strategy of $d = 0$. Consider the cases based on the sign of $B_{1,1} = C(1-\beta) - k_2$.

Case 2.1.1: $C\beta \leq k_1 < C$ and $k_2 \leq C(1-\beta)$

The attacker has the dominant strategy of always attacking. The true game has a unique equilibrium $(d=0,\, a=1)$. The perceived game has a unique equilibrium $(d=1,\, a=1)$. We have $\boldsymbol{\mu}_{21} = \begin{bmatrix} 0 \\ C \end{bmatrix}$ and $\boldsymbol{\mu}_{11} = \boldsymbol{\mu}_{12} = \begin{bmatrix} C\beta - k_1 \\ C(1-\beta) - k_2 \end{bmatrix}$. The nonzero payoff change is $\boldsymbol{\mu}_{11} - \boldsymbol{\mu}_{21} = \begin{bmatrix} C\beta - k_1 \\ -C\beta - k_2 \end{bmatrix}$ when the attacker has limited rank capability, with attainable payoff change types being $(\delta_{u1} \leq 0,\, \delta_{u2} < 0)$.

Case 2.1.2: $C\beta \leq k_1 < C$ and $k_2 > C(1-\beta)$

The attacker's best response is to attack if $d = 0$ and not attack if $d = 1$. The true game has a unique equilibrium $(d=0,\, a=1)$. The perceived game becomes a case of the matching pennies game with a unique mixed equilibrium. Assume the mixed strategy is

$\boldsymbol{s_1} = \begin{bmatrix} 1-p_1 \\ p_1 \end{bmatrix}$ and $\boldsymbol{s_2} = \begin{bmatrix} 1-p_2 \\ p_2 \end{bmatrix}$. It holds that:

$$0 = (1-p_1)C + p_1(C(1-\beta) - k_2), \quad (1-p_2)v_0 + p_2 v_1 = (1-p_2)tv_0 + p_2 t v_1$$
$$v_0 = \frac{C + t(C-k_1)}{t^2+1}, \quad v_1 = \frac{t(C\beta - k_1)}{t^2+1}$$

which yields

$$p_1 = \frac{1}{\beta + \frac{k_2}{C}}, \quad p_2 = \frac{1 + t\left(1 - \frac{k_1}{C}\right)}{1 + t(1-\beta)}$$

Eqs. (3.18) and (3.19) and Lemma 3.2 then imply:

$$t = \frac{-s + \sqrt{s^2+4}}{2}, \quad s = \frac{R_{\boldsymbol{A}^\intercal}}{Q_{\boldsymbol{A}^\intercal}} = \frac{1 - \left(1 - \frac{k_1}{C}\right)^2 - \left(\frac{k_1}{C} - \beta\right)^2}{1 - \frac{k_1}{C}}$$

Let's first analyze the feasible regions of p_1 and p_2. Let P_1 denote the set of feasible values of p_1 and P_2 for p_2. Since k_2 is independent of β and C while the only constraint is $\frac{k_2}{C} + \beta > 1$, we have $\inf P_1 = 0$ and $\sup P_1 = 1$. Setting $\beta = \frac{1}{2}$ and $\frac{k_1}{C} \to 1^-$ leads to $p_2 \to 1^-$, which yields $\sup P_2 = 1$. Since $\frac{\partial p_2}{\partial t} = \frac{\beta - \frac{k_1}{C}}{(1+t(1-\beta))^2} < 0$, we have $p_2 > p_2\big|_{t=1} = \frac{2 - \frac{k_1}{C}}{2-\beta} > \frac{1}{2}$. By setting $\frac{k_1}{C} = 1 - \frac{1}{n}$ and $\beta = \frac{1}{n^2}$, we have $\lim_{n\to+\infty} p_2 = \frac{\sqrt{2}}{2}$. It follows that $\frac{1}{2} \le \inf P_2 \le \frac{\sqrt{2}}{2}$. In conclusion, $P_1 = (0,\, 1)$ and $\left(\frac{\sqrt{2}}{2},\, 1\right) \subseteq P_2 \subseteq \left(\frac{1}{2},\, 1\right)$, while all pairs of values $(p_1,\, p_2) \in P_1 \times P_2$ are feasible.

The expected payoffs are:

- If $c_1 = 1$ and $c_2 = 2$, the strategy profile is $(d \sim B(p_1),\, a = 1)$, with the payoffs $\boldsymbol{\mu}_{12} = \begin{bmatrix} -C\frac{\frac{k_1}{C} - \beta}{\frac{k_2}{C} + \beta} \\ 0 \end{bmatrix}$.
- If $c_1 = 2$ and $c_2 = 1$, the strategy profile is $(d = 0,\, a \sim B(p_2))$, with the payoffs $\boldsymbol{\mu}_{21} = \begin{bmatrix} C(1-p_2) \\ Cp_2 \end{bmatrix}$.
- If $c_1 = 1$ and $c_2 = 1$, the strategy profile is $(d \sim B(p_1),\, a \sim B(p_2))$, with the payoffs

$$\boldsymbol{\mu}_{11} = \begin{bmatrix} \mu_0 \\ 0 \end{bmatrix}$$
$$\text{where } \mu_0 \coloneqq (1-p_1)(1-p_2)C + p_1(1-p1)(C-k_1) + p_1p_2(C\beta - k_1)$$
$$= C - \frac{k_1 + k_2 p_2}{\beta + \frac{k_2}{C}}$$

Finally, let's analyze the attainable payoff change types:

- When $c_1 = 1$, the payoff change is $\boldsymbol{\mu}_{11} - \boldsymbol{\mu}_{12} = \begin{bmatrix} \frac{(1-p_2)k_2}{\beta + \frac{k_2}{C}} \\ 0 \end{bmatrix}$. The attainable payoff change type is ($\delta_{u1} > 0$, $\delta_{u2} = 0$).
- When $c_2 = 1$, the payoff change is $\boldsymbol{\mu}_{11} - \boldsymbol{\mu}_{21} = \begin{bmatrix} \frac{p_2 C\beta - k_1}{\beta + \frac{k_2}{C}} \\ -Cp_2 \end{bmatrix}$. Since $p_2 C\beta - k_1 \leq p_2 k_1 - k_1 < 0$, the attainable payoff change type is ($\delta_{u1} < 0$, $\delta_{u2} < 0$).

Case 2.2: $k_1 < C\beta$

In the true game, the defender's best response to a is $d = a$.

- When $C(1-\beta) \geq k_2$, the attacker has a dominant strategy $a = 1$. The equilibrium is ($d = 1$, $a = 1$) with the payoffs $\boldsymbol{\mu}_{22} = \begin{bmatrix} C\beta - k_1 \\ C(1-\beta) - k_2 \end{bmatrix}$.
- When $C(1-\beta) < k_2$, there is a unique mixed equilibrium $\left(d \sim B\left(\frac{C}{C\beta + k_2}\right),\ a \sim B\left(\frac{k_1}{C\beta}\right)\right)$ with the expected payoffs $\boldsymbol{\mu}_{22} = \begin{bmatrix} C - \frac{k_1}{\beta} \\ 0 \end{bmatrix}$.

In the perceived game, we have $v_0 > 0$ and $v_1 > 0$ in (3.20). The defender has a dominant strategy depending on $t > 1$ or $t < 1$.

Case 2.2.1: $k_1 < C\beta$ and $\left(1 - \frac{k_1}{C}\right)^2 + \left(\beta - \frac{k_1}{C}\right)^2 \leq 1$ and $C(1-\beta) \geq k_2$

In this case, $R_{\boldsymbol{A}^\intercal} \geq 0$ and $t \leq 1$, the perceived game has the unique equilibrium ($d = 0$, $a = 1$). We have $\boldsymbol{\mu}_{21} = \begin{bmatrix} C\beta - k_1 \\ C(1-\beta) - k_2 \end{bmatrix}$ and $\boldsymbol{\mu}_{12} = \boldsymbol{\mu}_{11} = \begin{bmatrix} 0 \\ C \end{bmatrix}$. The nonzero payoff change is $\boldsymbol{\mu}_{11} - \boldsymbol{\mu}_{21} = \begin{bmatrix} k_1 - C\beta \\ k_2 + C\beta \end{bmatrix}$ with the payoff change type ($\delta_{u1} < 0$, $\delta_{u2} > 0$, $\delta_W > 0$).

Case 2.2.2: $k_1 < C\beta$ and $\left(1 - \frac{k_1}{C}\right)^2 + \left(\beta - \frac{k_1}{C}\right)^2 \leq 1$ and $C(1-\beta) < k_2$.

Similar to the case above, the perceived game has the unique equilibrium ($d = 0$, $a = 1$).

- If $c_1 = 1$ and $c_2 = 2$, the strategy profile is $\left(d = 0,\ a \sim B\left(\frac{k_1}{C\beta}\right)\right)$, with the expected payoffs $\boldsymbol{\mu}_{12} = \begin{bmatrix} C - \frac{k_1}{\beta} \\ \frac{k_1}{\beta} \end{bmatrix}$.
- If $c_1 = 2$ and $c_2 = 1$, the strategy profile is $\left(d \sim B\left(\frac{C}{C\beta + k_2}\right),\ a = 1\right)$, with the expected payoffs $\boldsymbol{\mu}_{21} = \begin{bmatrix} C\frac{C\beta - k_1}{C\beta + k_2} \\ 0 \end{bmatrix}$.
- If $c_1 = 1$ and $c_2 = 1$, the strategy profile is ($d = 0$, $a = 1$), with the payoffs $\boldsymbol{\mu}_{11} = \begin{bmatrix} 0 \\ C \end{bmatrix}$.

Therefore,

- If $c_1 = 1$, the payoff change is $\boldsymbol{\mu}_{11} - \boldsymbol{\mu}_{12} = \begin{bmatrix} \frac{k_1}{\beta} - C \\ C - \frac{k_1}{\beta} \end{bmatrix}$. The attainable payoff change type is ($\delta_{u1} < 0$, $\delta_{u2} > 0$, $\delta_W = 0$).
- If $c_2 = 1$, the payoff change is $\boldsymbol{\mu}_{11} - \boldsymbol{\mu}_{21} = \begin{bmatrix} C\frac{k_1 - C\beta}{C\beta + k_2} \\ C \end{bmatrix}$. The attainable payoff change type is ($\delta_{u1} < 0$, $\delta_{u2} > 0$, $\delta_W > 0$).

Case 2.2.3: $k_1 < C\beta$ and $\left(1 - \frac{k_1}{C}\right)^2 + \left(\beta - \frac{k_1}{C}\right)^2 > 1$ and $C(1-\beta) \geq k_2$

In this case, $R_{\boldsymbol{A}^\intercal} < 0$ and $t > 1$, the perceived game has the unique equilibrium ($d = 1$, $a = 1$). The strategy profile is the same as in the true game.

Case 2.2.4: $k_1 < C\beta$ and $\left(1 - \frac{k_1}{C}\right)^2 + \left(\beta - \frac{k_1}{C}\right)^2 > 1$ and $C(1-\beta) < k_2$

The perceived game has the unique equilibrium ($d = 1$, $a = 0$).

- If $c_1 = 1$ and $c_2 = 2$, the strategy profile is $\left(d = 1,\, a \sim B\left(\frac{k_1}{C\beta}\right)\right)$, with the expected payoffs $\boldsymbol{\mu}_{12} = \begin{bmatrix} C - \frac{k_1}{\beta} \\ (C(1-\beta) - k_2)\frac{k_1}{C\beta} \end{bmatrix}$.
- If $c_1 = 2$ and $c_2 = 1$, the strategy profile is $\left(d \sim B\left(\frac{C}{C\beta + k_2}\right),\, a = 0\right)$, with the expected payoffs $\boldsymbol{\mu}_{21} = \begin{bmatrix} C\frac{C\beta - k_1 + k_2}{C\beta + k_2} \\ 0 \end{bmatrix}$.
- If $c_1 = 1$ and $c_2 = 1$, the strategy profile is ($d = 1$, $a = 0$), with the expected payoffs $\boldsymbol{\mu}_{11} = \begin{bmatrix} C - k_1 \\ 0 \end{bmatrix}$.

Therefore,

- If $c_1 = 1$, the payoff change is $\boldsymbol{\mu}_{11} - \boldsymbol{\mu}_{12} = \begin{bmatrix} k_1\left(\frac{1}{\beta} - 1\right) \\ (k_2 - C(1-\beta))\frac{k_1}{C\beta} \end{bmatrix}$. The attainable payoff change type is ($\delta_{u1} > 0$, $\delta_{u2} > 0$).
- If $c_2 = 1$, the payoff change is $\boldsymbol{\mu}_{11} - \boldsymbol{\mu}_{21} = \begin{bmatrix} \frac{C(1-\beta) - k_2}{C\beta + k_2} k_1 \\ 0 \end{bmatrix}$. The attainable payoff change type is ($\delta_{u1} < 0$, $\delta_{u2} = 0$).

Theorem 3.16 is thus obtained by combining all the cases discussed above. □

Chapter 4

Capability-aware players in limited-perception games

This chapter discusses *capability-aware* players in limited-perception games. A capability-aware player knows their and their opponents' capabilities and thus considers the bounds of payoffs over the set of possible true payoff functions compatible with their perceived payoff functions.

Definition 4.1 (Capability-aware player)
In an OLP-Game as defined in Definition 2.3, a *capability-aware* player knows everyone's capabilities $(c_j)_{j\in\mathcal{N}}$ in any game instance. ◁

Section 4.1 discusses the potential objectives of a capability-aware player and argues that maximizing the worst-case expected payoff is a reasonable objective. Section 4.2 analyzes the equilibrium and some extended settings of fully-capability-aware games, where all players are capability-aware, maximize the worst-case expected payoff, and know these conditions as common knowledge. Section 4.3 studies the zero-sum case and identifies efficiently solvable zero-sum fully-capability-aware games. Section 4.4 discusses the capability transfer properties in limited-perception games.

4.1 Rational behavior of capability-aware players

A capability-aware player may seek to maximize different objectives, including the lower or upper bounds of their expected payoff as defined in Section 3.2. Let's consider three possible objectives of a capability-aware player: maximizing the worst possible payoff, maximizing the best possible payoff, or minimizing the uncertainty of possible payoffs:

Definition 4.2 (Response types of a capability-aware player)
In an OLP-Game as defined in Definition 2.3, a capability-aware player i with a capability of c_i and a perceived payoff function $v_i^i(\cdot)$ may define their best response set as one of the following given the mixed strategy profile of the other players $\boldsymbol{\sigma}_{-i}$:

- *Pessimistic response*: maximize the worst possible payoff given the perceived game

$$\mathrm{BR}_i^{\mathrm{ca,\,p}}\left(\boldsymbol{\sigma}_{-i}\right) \coloneqq \operatorname*{argmax}_{\boldsymbol{\sigma}_i \in \Delta_{\mathcal{S}_i}} \mathcal{P}^{-}\left(v,\, c,\, \boldsymbol{\sigma}_i \mid \boldsymbol{\sigma}_{-i}\right) \tag{4.1}$$

- *Optimistic response*: maximize the best possible payoff given the perceived game

$$\mathrm{BR}_i^{\mathrm{ca,\,o}}\left(\boldsymbol{\sigma}_{-i}\right) \coloneqq \operatorname*{argmax}_{\boldsymbol{\sigma}_i \in \Delta_{\mathcal{S}_i}} \mathcal{P}^{+}\left(v,\, c,\, \boldsymbol{\sigma}_i \mid \boldsymbol{\sigma}_{-i}\right) \tag{4.2}$$

- *Uncertainty-averse response*: minimize the uncertainty of all possible payoffs given the perceived game

$$\mathrm{BR}_i^{\mathrm{ca,\,u}}\left(\boldsymbol{\sigma}_{-i}\right) \coloneqq \operatorname*{argmin}_{\boldsymbol{\sigma}_i \in \Delta_{\mathcal{S}_i}} \left(\mathcal{P}^{+}\left(v,\, c,\, \boldsymbol{\sigma}_i \mid \boldsymbol{\sigma}_{-i}\right) - \mathcal{P}^{-}\left(v,\, c,\, \boldsymbol{\sigma}_i \mid \boldsymbol{\sigma}_{-i}\right)\right) \tag{4.3}$$

The player who uses the above response types is called a *pessimistic*, *optimistic*, or *uncertainty-averse* player, respectively. ◁

The above definition derives related equilibrium concepts. Since the lower bound of the expected true payoff is concave in $\boldsymbol{\sigma}_i$, there exists an equilibrium $(\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n)$ associated with the pessimistic response such that

$$\forall i \in \mathcal{N} : \; \boldsymbol{\sigma}_i \in \mathrm{BR}_i^{\mathrm{ca,\,p}}\left(\boldsymbol{\sigma}_{-i}\right) \tag{4.4}$$

The equilibrium exists since $\boldsymbol{\sigma}_i \mapsto \mathcal{P}^{-}\left(v,\, c,\, \boldsymbol{\sigma}_i \mid \boldsymbol{\sigma}_{-i}\right)$ is continuous and concave [Ros65]. However, when players have different perceptions due to different capability levels, the equilibrium is not a reasonable solution concept since a lower-capability player cannot predict

the best response of a higher-capability player, and thus cannot solve the equilibrium from their private information. Section 4.2 proposes a suitable solution concept for games where players have different capability levels.

Numerically solving the equilibrium in (4.4) is challenging. See Chernov [Che19] and references therein for a discussion on related algorithms. Many of the algorithms have additional assumptions, such as strict concavity [Che19] or single-valued best response functions [KU00]. Computing the equilibrium in (4.4) is PPAD-complete since it is harder than the bimatrix game, which is PPAD-complete [Nis+07], but not harder than the general-case concave game, which is also PPAD-complete [PVZ23].

Similarly, there also exists an equilibrium associated with the uncertainty-averse response since it minimizes a convex function. However, for the optimistic response, a related equilibrium may not exist since each player tries to maximize a convex function; of note, Daskalakis [Das21] provides an example of a convex maximization game without a Nash equilibrium. Example 4.1 demonstrates a limited-perception game without an equilibrium when players use the optimistic response.

Example 4.1 (A limited-perception game without an optimistic equilibrium) Consider a two-player game where the payoff functions are represented by matrices, the first player is the row player, and the second player is the column player. Define the following matrices:

$$\boldsymbol{A}_1 \coloneqq \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad \boldsymbol{A}_2 \coloneqq -\boldsymbol{A}_1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad \boldsymbol{B} \coloneqq \boldsymbol{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Define a function $\alpha : \mathbb{R}^{2\times 2} \times \overline{\mathbb{Z}}_+ \to \mathbb{R}^{2\times 2}$ as:

$$\alpha(\boldsymbol{M},\, c) \coloneqq \begin{cases} \boldsymbol{A}_1 & \text{if } c = 1 \text{ and } \boldsymbol{M} \in \{\boldsymbol{A}_1,\, \boldsymbol{A}_2\} \\ \boldsymbol{M} & \text{otherwise} \end{cases}$$

It is easy to verify that $\alpha(\cdot,\, \cdot)$ is a payoff perception function.

Consider an instance where the capability profile is $c_1 = c_2 = 1$, and both players perceive the payoff functions as $(\boldsymbol{A}_1,\, \boldsymbol{B})$. We have $\alpha^{-1}(\boldsymbol{A}_1,\, 1) = \{\boldsymbol{A}_1,\, \boldsymbol{A}_2\}$ and $\alpha^{-1}(\boldsymbol{B},\, 1) = \{\boldsymbol{B}\}$. Let $\begin{bmatrix} x & 1-x \end{bmatrix}^\intercal$ be the row player's mixed strategy and $\begin{bmatrix} y & 1-y \end{bmatrix}^\intercal$ be the column player's mixed strategy. The payoff bounds are given by:

$$\begin{aligned} \mathcal{P}^+\left(\boldsymbol{A}_1,\, 1,\, x \mid y\right) &= \max\left\{ \begin{bmatrix} x & 1-x \end{bmatrix} \boldsymbol{A}_1 \begin{bmatrix} y \\ 1-y \end{bmatrix},\, \begin{bmatrix} x & 1-x \end{bmatrix} \boldsymbol{A}_2 \begin{bmatrix} y \\ 1-y \end{bmatrix} \right\} \\ &= \max\left\{x - y,\, y - x\right\} = |x - y| \\ \mathcal{P}^+\left(\boldsymbol{B},\, 1,\, y \mid x\right) &= \begin{bmatrix} x & 1-x \end{bmatrix} \boldsymbol{B} \begin{bmatrix} y \\ 1-y \end{bmatrix} = xy + (1-x)(1-y) \end{aligned}$$

Assume both players use the optimistic response. Use $\mathrm{BR}_i : \mathbb{R} \rightrightarrows \mathbb{R}$ to denote the set of best responses of player i given their opponent's strategy. It holds that

$$\mathrm{BR}_1(y) = \begin{cases} \{1\} & \text{if } y < 0.5 \\ \{0,\, 1\} & \text{if } y = 0.5 \\ \{0\} & \text{if } y > 0.5 \end{cases}, \quad \mathrm{BR}_2(x) = \begin{cases} \{x\} & \text{if } x \neq 0.5 \\ [0, 1] & \text{if } x = 0.5 \end{cases}$$

Note that $\mathrm{BR}_1(0.5)$ is not a convex set. Since the row player tries to stay away from y but the column player tries to be close to x, the game has no equilibrium. In other words, there is no x and y such that $x \in \mathrm{BR}_1(y)$ and $y \in \mathrm{BR}_2(x)$. ◁

Although the equilibrium related to the optimistic response may not exist, the response itself is easy to compute, as shown by the following lemma:

Lemma 4.1 (Bauer maximum principle [Bau58])
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous convex function. Let $C \subseteq \mathbb{R}^n$ be a compact convex set. Then there exists an extreme point $\boldsymbol{x}^*$ in C such that $f(\boldsymbol{x}^*) = \sup_{\boldsymbol{x} \in C} f(\boldsymbol{x})$.

Proof. See Aliprantis and Border [AB13, Theorem 7.69]. □

Corollary 4.2 (Computing the optimistic response)
With the notation in Definition 4.2, some optimistic response strategies can be obtained with a pure strategy. Formally,

$$\max_{\boldsymbol{\sigma}_i \in \Delta_{\mathcal{S}_i}} \mathcal{P}^+(v, c, \boldsymbol{\sigma}_i \mid \boldsymbol{\sigma}_{-i}) = \max_{1 \le k \le |\mathcal{S}_i|} \mathcal{P}^+(v, c, \boldsymbol{e}_k \mid \boldsymbol{\sigma}_{-i}) \tag{4.5}$$

Proof. By Theorem 2.8 and Lemma 4.1, we only need to consider the extreme points of the simplex $\Delta_{\mathcal{S}_i}$ to find the maximizer of the optimistic response. Those extreme points are the standard basis vectors $\boldsymbol{e}_k$ for $1 \le k \le |\mathcal{S}_i|$. □

The uncertainty-averse response is less desirable than the pessimistic response in typical settings. Let $\boldsymbol{\sigma}_i$ be the uncertainty-averse response and $\boldsymbol{\sigma}_i'$ the pessimistic response of player i. Let $l \coloneqq \mathcal{P}^-(v, c, \boldsymbol{\sigma}_i \mid \boldsymbol{\sigma}_{-i})$, $u \coloneqq \mathcal{P}^+(v, c, \boldsymbol{\sigma}_i \mid \boldsymbol{\sigma}_{-i})$, $l' \coloneqq \mathcal{P}^-(v, c, \boldsymbol{\sigma}_i' \mid \boldsymbol{\sigma}_{-i})$, and $u' \coloneqq \mathcal{P}^+(v, c, \boldsymbol{\sigma}_i' \mid \boldsymbol{\sigma}_{-i})$ be the corresponding payoff bounds. Then $l \le l'$ and $u - l \le u' - l'$ by definition. It follows that $u \le u'$. Therefore, the uncertainty-averse response is dominated by the pessimistic response. The uncertainty-averse response may still be useful in certain scenarios. For example, in a cooperative setting, if a player communicates their expected payoff to their teammates, and the team gets penalized by the difference in the member's true payoffs, then reducing the payoff uncertainty helps other team members make decisions.

The above discussion shows that the optimistic response may not yield an equilibrium, and the payoff bounds of the uncertainty-averse response are dominated by the pessimistic response. Therefore, I focus on the pessimistic response in this thesis.

4.2 Fully-capability-aware games

4.2.1 Setup

As discussed in Section 4.1, the equilibrium directly derived by the pessimistic responses of capability-aware players as given in (4.4) may not be solvable by the players in an OLP-Game instance when they have different capabilities. This section analyzes the rational behavior of capability-aware players when they know that all players use the pessimistic response as common knowledge. Such games are called *fully-capability-aware games*. This section first formalizes the fully-capability-aware game, then defines Nash equilibrium and Stackelberg equilibrium in such games, proves the existence of the equilibrium, and finally discusses the computational hardness of the equilibrium.

Definition 4.3 (Fully-capability-aware game)
In an OLP-Game as defined in Definition 2.1, the game is called a *fully-capability-aware game* if the following conditions hold for any game instance (Definition 2.3):

- Each player knows the capability profile $(c_i)_{i\in\mathcal{N}}$.
- Each player knows the payoff perception function $\alpha(\cdot, \cdot)$.
- Each player is pessimistic: they try to maximize the worst possible payoff given the perceived game as defined in (4.1).
- The above three conditions are common knowledge among all players. ◁

4.2.2 Nash equilibrium in fully-capability-aware games

This section begins with defining Nash equilibrium for two-player fully-capability-aware games for easier exposition. The definition is then extended to the general case with a finite number of players.

The Nash equilibrium is defined for games where all players move simultaneously [Nas50]. At the equilibrium, no player has an incentive to unilaterally deviate from their strategy given the strategy of the other players. In a conventional bimatrix game, Nash equilibrium is a pair of mixed strategies. Players sample their strategies from the distributions defined by the mixed strategies. At the equilibrium, no player can deviate from their mixed strategy to improve their expected payoff.

However, in two-player fully-capability-aware games, a pair of mixed strategies $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ cannot constitute an equilibrium. Assume the first player has a lower capability. They do not know the perceived game of the second player, and thus they can not solve $\boldsymbol{y}^*$ from their private information. The second player can reconstruct the first player's perception and predict $\boldsymbol{x}^*$. Even if we allow players to communicate, without a commitment mechanism like in a Stackelberg game, the first player should not believe in the second player's announced strategy since they may be strategically misleading the first player.

A mixed strategy can be understood as a first-order function that maps a pure strategy to the probability of choosing that strategy. By comparison, in a fully-capability-aware game, the equilibrium should be defined by a higher-order function that encodes best responses of all possible perceived games of the higher-capability player. This function is called the *response function*. The lower-capability player defends against the worst outcome over possible strategies defined by the response function of the higher-capability player. The higher-capability player applies the response function to their privately perceived game to obtain the mixed strategy.

Before formalizing Nash equilibrium of two-player fully-capability-aware games, let's first introduce simplified notation for two-player games that will also be used in other parts of this chapter.

Definition 4.4 (Simplified notation in a two-player game)
In a two-player OLP-Game instance as defined in Definition 2.3, let (c_1, c_2) be the capability profile of the two players. The following notation is used:

- u^* and v^* denote the first and the second player's true payoff functions, respectively.
- (u_1, v_1) and (u_2, v_2) denote the players' respective perceived payoff functions; i.e.,

$$u_1 \coloneqq \alpha(u^*, c_1), \quad v_1 \coloneqq \alpha(v^*, c_1), \quad u_2 \coloneqq \alpha(u^*, c_2), \quad v_2 \coloneqq \alpha(v^*, c_2)$$

- $\boldsymbol{x}$ and $\boldsymbol{y}$ denote the players' respective mixed strategies; i.e., $\boldsymbol{x}$ for $\boldsymbol{\sigma}_1$ and $\boldsymbol{y}$ for $\boldsymbol{\sigma}_2$. ◁

Definition 4.5 (Nash equilibrium of two-player fully-capability-aware games) In an instance of a two-player fully-capability-aware game as defined in Definition 4.3 with the capability profile (c_1, c_2), assume without loss of generality that $c_1 \leq c_2$. Let $U \coloneqq \mathbb{R}^{\mathcal{S}_1 \times \mathcal{S}_2}$ be the set of all possible payoff functions. Let (u_1, v_1) and (u_2, v_2) be the perceived payoff functions of the first and the second player, respectively.

Define T_1^2 as the set of possible perceived payoff functions of the second player from the first player's perspective:

$$T_1^2 \coloneqq \{v_2' \in U \mid \exists v^* \in U : \ \alpha(v^*, c_2) = v_2' \text{ and } \alpha(v_2', c_1) = v_1\} \tag{4.6}$$

A Nash equilibrium of this instance is a pair $(\boldsymbol{x}^*, \boldsymbol{Y})$ where $\boldsymbol{x}^* \in \Delta_{\mathcal{S}_1}$ is a mixed strategy of the first player, and $\boldsymbol{Y} : T_1^2 \mapsto \Delta_{\mathcal{S}_2}$ is a response function of the second player. The equilibrium condition is that $\boldsymbol{x}^*$ maximizes the first player's worst-case payoff over perceptions in T_1^2, and $\boldsymbol{Y}$ encodes a best response of the second player to any possibility in T_1^2. Formally, $(\boldsymbol{x}^*, \boldsymbol{Y})$ is a Nash equilibrium if and only if the following conditions hold:

- For all possible perceived payoff functions $v_2' \in T_1^2$, the second player has no incentive to deviate from the strategy $\boldsymbol{Y}(v_2')$ given that the first player plays $\boldsymbol{x}^*$:

$$\forall v_2' \in T_1^2 : \ \boldsymbol{Y}(v_2') \in \operatorname*{argmax}_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^-(v_2', c_2, \boldsymbol{x}^*, \boldsymbol{y}) \tag{4.7}$$

- Let $f_{\boldsymbol{Y}}(\boldsymbol{x})$ be the first player's worst payoff when they play $\boldsymbol{x}$ against $\boldsymbol{Y}$, defined as

$$f_{\boldsymbol{Y}}(\boldsymbol{x}) \coloneqq \inf_{v_2' \in T_1^2} \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{Y}(v_2')) \tag{4.8}$$

  Then $\boldsymbol{x}^*$ is a maximizer of $f_{\boldsymbol{Y}}(\boldsymbol{x})$:

$$\boldsymbol{x}^* \in \operatorname*{argmax}_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} f_{\boldsymbol{Y}}(\boldsymbol{x}) \tag{4.9}$$

Given a Nash equilibrium $(\boldsymbol{x}^*, \boldsymbol{Y})$, the first player samples an action according to $\boldsymbol{x}^*$, and the second player samples an action according to $\boldsymbol{Y}(v_2)$. The values $f_{\boldsymbol{Y}}(\boldsymbol{x}^*)$ and $\mathcal{P}^-(v_2, c_2, \boldsymbol{x}^*, \boldsymbol{Y}(v_2))$ are the *Nash values* of the first and second player at the equilibrium, respectively. ◁

Remark 4.1 In Definition 4.5, since T_1^2 only depends on v_1, c_1 and c_2, and the second player can reconstruct the first player's perception (i.e., $u_1 = \alpha(u_2, c_1)$ and $v_1 = \alpha(v_2, c_1)$), both players can solve $(\boldsymbol{x}^*, \boldsymbol{Y})$ using their private information. Both players have no incentive to deviate from the equilibrium.

Remark 4.2 As with the conventional Nash equilibrium, Definition 4.5 does not address the equilibrium selection problem. Instead, it assumes both players come to the identical equilibrium solution $(\boldsymbol{x}^*, \boldsymbol{Y})$ using their private information if there are multiple equilibria.

Remark 4.3 In the special case of $c_1 = c_2$, the specification of T_1^2 in (4.6) requires $v_1 = \alpha(v_2', c_1)$ and $\bar{c}(v_2') \leq c_1$, which implies $v_2' = \alpha(v_2', \bar{c}(v_2')) = \alpha(v_2', c_1) = v_1 = v_2$. Therefore, we have $T_1^2 = \{v_1\}$. Let $\boldsymbol{y}^* \coloneqq \boldsymbol{Y}(v_2) = \boldsymbol{Y}(v_1)$; the pair $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is solvable by both players and coincides with the conventional Nash equilibrium of (4.4).

Definition 4.5 handles the two players separately to facilitate the comprehensibility of the definition. In the general case, we can define a response function for each player and define Nash equilibrium by imposing symmetric constraints to the response functions.

Before defining Nash equilibrium for the general case, let's first formalize the *narrow concretization set* as a generalization of T_1^2 in (4.6) to any pair of players. Suppose one player has a capability level of c_1 and perceives the payoff function of a c_2-capability player as v, then $\mathrm{N_C}(v, c_1, c_2)$ denotes the set of possible perceived payoff functions of the c_2-capability player compatible with the perceived payoff function of the c_1-capability player.

Definition 4.6 (Narrow concretization set)
In an OLP-Game as defined in Definition 2.3, define the *narrow concretization set* $\mathrm{N_C} : U \times \mathbb{Z}_+^2 \mapsto 2^U$ as:

$$\mathrm{N_C}(v, c_1, c_2) \coloneqq \left\{ u \in U \;\middle|\; \begin{array}{l} \bar{c}(u) \leq c_2 \text{ and} \\ \alpha(v, \min\{c_1, c_2\}) = \alpha(u, \min\{c_1, c_2\}) \end{array} \right\} \tag{4.10}$$

◁

Theorem 4.3 (Property of the narrow concretization set)
The narrow concretization set $\mathrm{N_C}(v, c_1, c_2)$ *satisfies the following properties:*

- *If* $\bar{c}(v) \leq c_1 < c_2$, *then* $\mathrm{N_C}(v, c_1, c_2) \subseteq \alpha^{-1}(v, c_1)$.
- *If* $c_1 \geq c_2$, *then* $\mathrm{N_C}(v, c_1, c_2) = \{\alpha(v, c_2)\}$.
- *If* $c_1' \leq c_1$, *then* $\mathrm{N_C}(v, c_1, c_2) \subseteq \mathrm{N_C}(\alpha(v, c_1'), c_1', c_2)$.

Proof. If $\bar{c}(v) \leq c_1 < c_2$ and $u \in \mathrm{N_C}(v, c_1, c_2)$, then $\alpha(u, c_1) = \alpha(v, c_1) = v$, which means $u \in \alpha^{-1}(v, c_1)$.

If $c_1 \geq c_2$ and $u \in \mathrm{N_C}(v, c_1, c_2)$, then $u \overset{①}{=} \alpha(u, c_2) = \alpha(v, c_2)$, where $\overset{①}{=}$ is due to $\bar{c}(u) \leq c_2$ and Theorem 2.1.

To prove the third property, consider a few cases. If $c_1 \leq c_2$, let $u \in \mathrm{N_C}(v, c_1, c_2)$. Then $\alpha(u, c_1) = \alpha(v, c_1)$ and $\bar{c}(u) \leq c_2$. Given $c_1' \leq c_1$, the capability path independence implies that $\alpha(u, c_1') = \alpha(\alpha(u, c_1), c_1') = \alpha(\alpha(v, c_1), c_1') = \alpha(\alpha(v, c_1'), c_1')$, which yields $u \in \mathrm{N_C}(\alpha(v, c_1'), c_1', c_2)$. If $c_1 > c_2$, then $\mathrm{N_C}(v, c_1, c_2) = \{w\}$ where $w \coloneqq \alpha(v, c_2)$. If $c_2 \leq c_1' \leq c_1$, then $\mathrm{N_C}(\alpha(v, c_1'), c_1', c_2) = \{\alpha(\alpha(v, c_1'), c_2)\} = \{w\}$. If $c_1' < c_2 < c_1$, then $\alpha(w, \min\{c_1', c_2\}) = \alpha(v, c_1') = \alpha(\alpha(v, c_1'), \min\{c_1', c_2\})$, implying $w \in \mathrm{N_C}(\alpha(v, c_1'), c_1', c_2)$. Combining all cases yields $\mathrm{N_C}(v, c_1, c_2) \subseteq \mathrm{N_C}(\alpha(v, c_1'), c_1', c_2)$. □

Definition 4.7 (Nash equilibrium of fully-capability-aware games) In an instance of a fully-capability-aware game as defined in Definition 4.3, define $T_i^j \coloneqq \mathrm{N_C}\left(v_j^i,\, c_i,\, c_j\right)$ as the set of possible perceived payoff functions of player j from the perspective of player i. Let $m = \operatorname{argmin}^*_{i\in\mathcal{N}} c_i$ be a player with the lowest capability level. For $i \in \mathcal{N}$, let $\boldsymbol{R}_i : T_m^i \mapsto \Delta_{\mathcal{S}_i}$ be the response function of player i. The tuple $(\boldsymbol{R}_1, \ldots, \boldsymbol{R}_n)$ is called a *response profile*.

A response profile is a Nash equilibrium of the instance if and only if the following condition holds for each $i \in \mathcal{N}$:

$$\begin{aligned} &\forall u \in T_m^i :\ \boldsymbol{R}_i(u) \in \operatorname*{argmax}_{\boldsymbol{\sigma}\in\Delta_{\mathcal{S}_i}} f_i\left(u,\, \boldsymbol{\sigma} \mid \boldsymbol{R}_{-i}\right) \\ &\text{where } f_i\left(u,\, \boldsymbol{\sigma} \mid \boldsymbol{R}_{-i}\right) \coloneqq \inf_{v_j'\in T_i^j \text{ for } j\in N_i} \mathcal{P}^-\left(u,\, c_i,\, \boldsymbol{\sigma} \;\middle|\; \boldsymbol{R}_j\left(v_j'\right) \text{ for } j \in N_i\right) \\ &N_i \coloneqq \mathcal{N} \setminus \{i\} \end{aligned} \tag{4.11}$$

At Nash equilibrium, player i samples an action from the distribution defined by $\boldsymbol{R}_i(v_i^i)$. The value $f_i\left(v_i^i,\, \boldsymbol{R}_i(v_i^i) \mid \boldsymbol{R}_{-i}\right)$ is called the *Nash value* of player i at the equilibrium. ◁

Remark 4.4 For any $i \in \mathcal{N}$, since $T_j^i \subseteq T_m^i$ for all $j \in \mathcal{N}$ (third property in Theorem 4.3), T_m^i is sufficient to contain the values of u used in the infimum of other players' f_j for $j \in \mathcal{N}\setminus\{i\}$. Therefore, (4.11) only needs to ensure that $\boldsymbol{R}_i(u)$ is a best response to u and $\boldsymbol{R}_{-i}$ for $u \in T_m^i$. Of note, T_m^i is bounded due to the bounded concretization property (Definition 2.2).

In the two-player case with $c_1 \le c_2$, the case of $i = 1$ in (4.11) corresponds to (4.8) and (4.9), and the case of $i = 2$ corresponds to the constraints on $\boldsymbol{Y}(v_2')$ for $v_2' \in T_1^2$ in (4.7) (note that $T_1^1 = T_2^1 = \{u_1\}$, $\boldsymbol{Y} = \boldsymbol{R}_2$, and $\boldsymbol{x}^* = \boldsymbol{R}_1(u_1)$). Therefore, the definition is consistent with Definition 4.5 in the two-player case.

If $c_i = c_j$ for all $i, j \in \mathcal{N}$, then $T_i^j = \left\{v_j^j\right\}$ for all $i, j \in \mathcal{N}$. The definition is consistent with (4.4) in this case.

Remark 4.5 In the two-player case, both players can solve Nash equilibrium from their private information. However, in the multi-player case, only the two players with the highest capability levels can solve Nash equilibrium from their private information. Suppose player i has the highest capability level, and player j has the second-highest capability level; i.e., $i \in \operatorname{argmax}_{p\in\mathcal{N}} c_p$ and $j \in \operatorname{argmax}_{p\in\mathcal{N}\setminus\{i\}} c_p$. Then both players i and j can compute T_k^l for all $k, l \in \mathcal{N}$ and $k \ne l$. Both players can compute T_m^k for $k \in \mathcal{N}$. If there are more than one player with the highest capability level (i.e., $\left|\operatorname{argmax}_{p\in\mathcal{N}} c_p\right| > 1$), then all such highest-capability players can solve Nash equilibrium from their private information. For another player t such that $c_t < c_j$, player t cannot compute T_j^i and thus cannot define f_j that is required to solve Nash equilibrium.

Although the lower-capability players may be unable to solve Nash equilibrium, they can still agree on a response profile being a Nash equilibrium if all players' agreement is common knowledge built with an inductive argument. To simplify the discussion, let's consider a three-player game and an instance where $c_1 < c_2 < c_3$. Assume all players announce that they agree to use the response profile $(\boldsymbol{R}_1, \boldsymbol{R}_2, \boldsymbol{R}_3)$ with the agreement being common knowledge. Although player 1 can not solve the equilibrium from their private information

and can only verify that (4.11) holds for $i = 1$, player 1 knows that player 2 verifies (4.11) for $i = 1$ since player 2 knows that player 1 can verify for $i = 1$ and would deviate if the condition does not hold; with a similar argument, player 1 knows that player 2 also verifies (4.11) for $i = 2$ and $i = 3$. Similarly, player 1 knows that player 3 verifies (4.11) for $i \in \mathcal{N}$. Since players 2 and 3 know that their opponent can verify all conditions, none of them is motivated to "cheat" by deviating from the announcement. Therefore, given the common knowledge of the agreement on $(\boldsymbol{R}_1, \boldsymbol{R}_2, \boldsymbol{R}_3)$, player 1 can conclude that all players have no incentive to deviate.

In a non-Nash setup, if the two players with the highest capabilities agree to collaborate, they may play a non-Nash strategy that benefits them. The lower-capability player is unable to detect the collusion using their private information. In the above example, the players 2 and 3 could agree to use a non-Nash strategy $\boldsymbol{R}'_2$ and $\boldsymbol{R}'_3$ such that (4.11) does not hold for $i = 2$ and/or $i = 3$, but still holds for $i = 1$ for some $\boldsymbol{R}'_1$. Players 2 and 3 can be incentivized to collude if $(\boldsymbol{R}'_1, \boldsymbol{R}'_2, \boldsymbol{R}'_3)$ gives them both better payoffs than Nash equilibrium. This deviation from Nash is not detectable by player 1. However, this collusion is unstable since at least one of the players 2 and 3 can increase their payoff by deviating from the collusion (if none of them can increase their own payoff, then $(\boldsymbol{R}'_1, \boldsymbol{R}'_2, \boldsymbol{R}'_3)$ is a Nash equilibrium by definition).

Next comes the proof of Nash equilibrium's existence in fully-capability-aware games. The proof needs a generalization of the Kakutani's fixed-point theorem to infinite-dimensional spaces, which is stated below:

Theorem 4.4 ([Fan52, Theorem 1])
Let L be a locally convex topological vector space and K a compact convex set in L. Let $\mathfrak{K}(K)$ be the family of all closed convex nonempty subsets of K. Then for any upper hemicontinuous point-to-set transformation f from K into $\mathfrak{K}(K)$, there exists a point $x_0 \in K$ such that $x_0 \in f(x_0)$.

Theorem 4.5 (Existence of Nash equilibrium in fully-capability-aware games)
The Nash equilibrium of a fully-capability-aware game as defined in Definition 4.7 always exists.

Proof. Define the vector space $X \coloneqq \prod_{i\in\mathcal{N}} \mathcal{B}(U \times \mathcal{S}_i)$ over $\mathbb{R}$ where $\mathcal{B}(A)$ denotes the set of bounded real-valued functions on A and the addition and scalar multiplication are defined point-wise on the functions. For any point $p \in X$, use the tuple $p = (\boldsymbol{R}_1, \ldots, \boldsymbol{R}_n)$ to denote the components of p where $\boldsymbol{R}_i \in \mathcal{B}(U \times \mathcal{S}_i)$; $\boldsymbol{R}_i$ is interpreted as a function from U to $\mathbb{R}^{\mathcal{S}_i}$. Extend X to a metric space by defining the distance function using the uniform convergence norm defined as $\|(\boldsymbol{R}_1, \ldots, \boldsymbol{R}_n)\|_\infty \coloneqq \max_{i\in[\![1,n]\!]} \sup_{u\in U} \max_{j\in[\![1,|\mathcal{S}_i|]\!]} |R_{ij}(u)|$ where $R_{ij}(u)$ is the j-th component of $\boldsymbol{R}_i(u)$. Of note, the supremum in the norm's definition is finite since $\boldsymbol{R}_i(\cdot)$ is a bounded function; the norm topology induces a locally convex topological vector space [AB13, Section 6.1]. Define the subset $S \coloneqq \prod_{i\in\mathcal{N}} \Delta^U_{\mathcal{S}_i} \subseteq X$. It is easy to verify that S is compact, convex, and nonempty.

Given the capability profile $(c_1, \ldots, c_n)$ and the perceived payoff functions $\left(v_i^j\right)_{(i,j)\in\mathcal{N}^2}$, with $f_i(\cdot, \cdot \mid \cdot)$ and T_m^i defined as in Definition 4.7, define the function $g : S^2 \mapsto \mathbb{R}$ as:

$$g\big((\boldsymbol{R}_i)_{i\in\mathcal{N}}, (\boldsymbol{R}'_i)_{i\in\mathcal{N}}\big) \coloneqq \sum_{i\in\mathcal{N}} \inf_{u\in T_m^i} \left(f_i(u, \boldsymbol{R}'_i(u) \mid \boldsymbol{R}_{-i}) - \max_{\boldsymbol{\sigma}\in\Delta_{\mathcal{S}_i}} f_i(u, \boldsymbol{\sigma} \mid \boldsymbol{R}_{-i}) \right) \tag{4.12}$$

The function $g(\cdot, \cdot)$ is bounded since the bounded concretization property (Definition 2.2) implies that T_m^i is bounded. Since $\mathcal{P}^-\left(u, c_i, \boldsymbol{\sigma} \,\middle|\, \boldsymbol{R}_j(u_j')\right)$ is continuous in both $\boldsymbol{\sigma}$ and $\boldsymbol{R}_{-i}$ and concave in $\boldsymbol{\sigma}$ (Theorem 2.8), $f_i\left(u, \boldsymbol{\sigma} \mid \boldsymbol{R}_{-i}\right)$ is continuous in both $\boldsymbol{\sigma}$ and $\boldsymbol{R}_{-i}$ and concave in $\boldsymbol{\sigma}$. Therefore, the supremum of $\boldsymbol{\sigma} \mapsto f_i\left(u, \boldsymbol{\sigma} \mid \boldsymbol{R}_{-i}\right)$ is always attainable. It also follows that $g(p, q)$ is continuous in both p and q and concave in q.

Define a set-valued function $\Phi : S \mapsto 2^S$ as:

$$\Phi(p) \coloneqq \{q \in S \mid g(p, q) = 0\} \tag{4.13}$$

The definition of $g(\cdot, \cdot)$ implies that $g(p, q) \leq 0$. By setting $\boldsymbol{R}_i'(u)$ as an arbitrary maximizer of $\boldsymbol{\sigma} \mapsto f_i\left(u, \boldsymbol{\sigma} \mid \boldsymbol{R}_{-i}\right)$ over $\Delta_{\mathcal{S}_i}$, $\Phi(p)$ is nonempty. Since $g(p, q)$ is concave and continuous in q, $\Phi(p)$ is compact and convex. Moreover, since $g(p, q)$ is continuous in both p and q, $\operatorname{Gr}\Phi \coloneqq \{(p, q) \in S^2 \mid q \in \Phi(p)\}$ is closed, and $\Phi(\cdot)$ is thus upper hemicontinuous [AB13, Theorem 17.11]. Theorem 4.4 implies that there exists $p_0 \in S$ such that $p_0 \in \Phi(p_0)$. Let $p_0 = (\boldsymbol{R}_1^*, \ldots, \boldsymbol{R}_n^*)$. One can easily verify that $(\boldsymbol{R}_1^*, \ldots, \boldsymbol{R}_n^*)$ is a Nash equilibrium of the fully-capability-aware game. □

4.2.3 Hardness of equilibrium solving and representation

Since Nash equilibrium and Stackelberg equilibrium both involve the inversion of the payoff perception function, they can be expected to be computationally hard. This intuition can be strengthened. The following theorem states that assuming the best response of the second player is efficiently computable, checking whether a given strategy will be used by the second player as a best response to a given strategy of the first player is NP-hard. The hardness implies that verifying a Nash or Stackelberg equilibrium (see Section 4.2.4.2 below) of a fully-capability-aware game as defined in Definitions 4.7 and 4.9 is also NP-hard.

Theorem 4.6 (Hardness of checking the best response)
Define the following decision problem in the context of an instance of a two-player fully-capability-aware game as defined in Definitions 4.3 and 4.4. Assume all input values of the problem and all input/output values of intermediate Turing machines can be exactly represented as rational numbers of polynomial size. The problem is NP*-complete.*

- ***Input**: (i) $m \in \mathbb{Z}_+$; and (ii) A Turing machine that defines a payoff perception function $\alpha(\cdot, \cdot)$.*
- ***Constraints on input**: (i) The strategy spaces are $\mathcal{S}_1 = [\![1, m]\!]$ and $\mathcal{S}_2 = \{1, 2\}$; (ii) For any $u \in U$, the payoff perception function satisfies $\alpha(u, 2) = u$, $\alpha(u, 1)$ is computable in $\mathsf{poly}(m)$ time, $\alpha(\cdot, \cdot)$ is representable by a Turing machine of size $\mathsf{poly}(m)$, $|\alpha^{-1}(u, 1)|$ is finite, and $\alpha^{-1}(u, 1)$ is enumerable by a Turing machine of size $\mathsf{poly}(m)$ in $\mathsf{poly}(|\alpha^{-1}(u, 1)|)$ time; and (iii) Given $u \in U$, $\boldsymbol{x} \in \Delta_{\mathcal{S}_1}$, and $c \in \mathbb{Z}_+$, both $\mathcal{P}^-(u, c, \boldsymbol{x}, \boldsymbol{y})$ for any $\boldsymbol{y} \in \Delta_{\mathcal{S}_2}$ and $\operatorname{argmax}^*_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^-(u, c, \boldsymbol{x}, \boldsymbol{y})$ are computable in $\mathsf{poly}(m)$ time.*
- ***Constants**: The first player's perceived payoff function of the second player, denoted as v_1, is defined as $v_1(x, y) = -y$ for all $(x, y) \in \mathcal{S}_1 \times \mathcal{S}_2$. Define $\boldsymbol{y}_0 = \begin{bmatrix} 0 & 1 \end{bmatrix}^\intercal$, $c_1 = 1$, and $c_2 = 2$.*

- ***Arbitrary values***: *(i)* $\boldsymbol{x}_0 \in \Delta_{\mathcal{S}_1}$ *is arbitrarily chosen; and (ii) an arbitrary Turing machine that defines a best response function* $\boldsymbol{Y} : U \mapsto \Delta_{\mathcal{S}_2}$ *such that for all* $u \in U$, $\boldsymbol{Y}(u)$ *is computable in* $\mathsf{poly}(m)$ *time, and* $\boldsymbol{Y}(u) \in \operatorname{argmax}_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^-(u, c_2, \boldsymbol{x}_0, \boldsymbol{y})$. *Of note,* $\boldsymbol{x}_0$ *and* $\boldsymbol{Y}(\cdot)$ *can be chosen depending on the input; there is no choice of* $\boldsymbol{x}_0$ *or* $\boldsymbol{Y}(\cdot)$ *that makes the problem easier.*
- ***Output***: *whether* $\boldsymbol{y}_0 \in \mathrm{N_C}(v_1, c_1, c_2)$.

Proof. The proof uses a reduction from the Boolean satisfiability problem (SAT), a well-known NP-complete problem [GJ79]. Let ϕ be a Boolean formula with n variables and $\mathsf{poly}(n)$ clauses. Denote the variables in ϕ as $\{t_1, \ldots, t_n\}$. Let $m = 2n$. Define a set $S_\phi \subseteq U$ as a set of payoff functions that encodes the solution to the SAT problem ϕ. Given a payoff function $u \in U$, $u \in S_\phi$ if and only if all of the following conditions hold:

- $u(x, 1) = 1$ for all $x \in [\![1, 2n]\!]$.
- $\{u(2k-1, 2), u(2k, 2)\} = \{2, 3\}$ for all $k \in [\![1, n]\!]$.
- Let t_k be true if and only if $u(2k, 2) = 3$. The assignment $\{t_1, \ldots, t_n\}$ satisfies the formula ϕ.

Let $v_1(x, y) = -y$ as defined above. Define a function $\alpha : U \times \mathbb{Z}_+ \to U$ as

$$\alpha(u, c) = \begin{cases} v_1 & \text{if } u \in S_\phi \cup \{v_1\} \text{ and } c = 1 \\ u & \text{otherwise} \end{cases} \tag{4.14}$$

It is straightforward to verify that $\alpha(\cdot, \cdot)$ satisfies Definition 2.2 and is thus a payoff perception function. One can also easily verify that $\alpha(u, c)$ is computable in $\mathsf{poly}(n)$ time with a Turing machine of size $\mathsf{poly}(n)$.

The construction above further ensures that ϕ is satisfiable if and only if there exists $u \neq v_1$ such that $\alpha(u, 1) = v_1$. Thus it holds that

$$\alpha^{-1}(v_1, 1) = S_\phi \cup \{v_1\} \tag{4.15}$$

Define $\boldsymbol{y}(a) \coloneqq \begin{bmatrix} a & 1-a \end{bmatrix}^\intercal$ where $0 \leq a \leq 1$. For any $\boldsymbol{x} \in \Delta_{\mathcal{S}_1}$, we have

$$\begin{aligned} v_1(\boldsymbol{x}, \boldsymbol{y}(a)) &= (-1) \cdot a + (-2) \cdot (1-a) = a - 2 < 0 \\ u(\boldsymbol{x}, \boldsymbol{y}(a)) &= a + \lambda(1-a) = (1-\lambda)a + \lambda > 0 \quad \text{for } u \in S_\phi \\ \text{where } \lambda &\coloneqq \sum_{k=1}^{n} \left(u(2k-1, 2)x_{2k-1} + u(2k, 2)x_{2k}\right) \end{aligned} \tag{4.16}$$

The value of λ in (4.16) satisfies $2 \leq \lambda \leq 3$ due to the definition of $u(\cdot, 2)$ for $u \in S_\phi$. It follows that for $c \in \overline{\mathbb{Z}}_+$ and $u \in S_\phi$, $\mathcal{P}^-(v_1, c, \boldsymbol{x}, \boldsymbol{y}(a)) = a - 2$ and $\mathcal{P}^-(u, c, \boldsymbol{x}, \boldsymbol{y}(a)) = (1-\lambda)a + \lambda$. Thus

$$\begin{aligned} \forall c \in \overline{\mathbb{Z}}_+ : &\quad \operatorname*{argmax}_{a \in [0,1]} \mathcal{P}^-(v_1, c, \boldsymbol{x}, \boldsymbol{y}(a)) = \{1\} \\ \forall c \in \overline{\mathbb{Z}}_+, \forall u \in S_\phi : &\quad \operatorname*{argmax}_{a \in [0,1]} \mathcal{P}^-(u, c, \boldsymbol{x}, \boldsymbol{y}(a)) = \{0\} \end{aligned} \tag{4.17}$$

Therefore, $\operatorname{argmax}^*_{\boldsymbol{y}\in\Delta_{\mathcal{S}_2}} \mathcal{P}^-(u, c, \boldsymbol{x}, \boldsymbol{y})$ is computable in $\mathsf{poly}(n)$ time for any $u \in U$, and thus the payoff perception function defined by (4.14) satisfies the constraints specified in Theorem 4.6.

Define a set $J \coloneqq \{\boldsymbol{Y}(v_2') \mid v_2' \in \mathrm{N_C}(v_1, c_1, c_2)\}$. Combining (4.15) and (4.17) with the definition of J yields

$$J = \begin{cases} \left\{\begin{bmatrix}1 & 0\end{bmatrix}^\intercal, \begin{bmatrix}0 & 1\end{bmatrix}^\intercal\right\} & \text{if } \phi \text{ is satisfiable} \\ \left\{\begin{bmatrix}1 & 0\end{bmatrix}^\intercal\right\} & \text{otherwise} \end{cases} \tag{4.18}$$

As a result, whether $\boldsymbol{y}_0 \in J$ is equivalent to whether ϕ is satisfiable, which makes the problem NP-hard. On the other hand, since we assume $\alpha^{-1}(u, 1)$ is finite and enumerable, the problem is in NP. Therefore, the problem is NP-complete. □

A consequence of Theorem 4.6 is that verifying an ϵ-Nash equilibrium or an ϵ-Stackelberg equilibrium of a fully-capability-aware game is both NP-hard and coNP-hard. With the setup in the proof of Theorem 4.6, one can choose a payoff function u_1, a strategy $\boldsymbol{x}_0$, and a value $\epsilon \in \mathbb{R}_+$ such that $(\boldsymbol{x}_0, \boldsymbol{Y}(\cdot))$ is an ϵ-Nash equilibrium (i.e., $\alpha^{-1}(u_1, c_1) = \{u_1\}$, $\min_{y\in J} u_1(\boldsymbol{x}_0, y) \geq \max_{\boldsymbol{x}\in\Delta_{\mathcal{S}_1}} \min_{y\in J} u_1(\boldsymbol{x}, y) - \epsilon$) if and only if $\boldsymbol{y}_0 \in J$. Similarly, one can choose values such that $(\boldsymbol{x}_0, \boldsymbol{Y}(\cdot))$ is an ϵ-Nash equilibrium if and only if $\boldsymbol{y}_0 \notin J$.

Verifying a Nash equilibrium can be harder if we drop some assumptions in Theorem 4.6. In particular, it is undecidable if the payoff perception function is only assumed to be computable. To see this, consider a program P and an input I. We can use a construction similar to the proof of Theorem 4.6 to define $\alpha(\cdot, \cdot)$ such that $\alpha(u, 1) = v_1$ if and only if $u(\cdot, 1) = 1$, $u(\cdot, 2) = 2 + \frac{1}{k}$ for some $k \in \mathbb{Z}_+$, and P halts on input I after k steps. Then whether $\boldsymbol{y}_0 \in J$ is equivalent to whether P halts on input I. The undecidability of the halting problem implies the undecidability of verifying a Nash equilibrium.

Let's turn our attention to the representation of the equilibrium. Assume there is a trusted oracle that finds one Nash equilibrium. If both players are Turing-complete, can the oracle communicate the equilibrium to them? The answer is not obvious since representing $\boldsymbol{Y}(\cdot)$ may require an infinitely sized lookup table. Representation size is also related to the hardness of equilibrium solving. Assume $T_1^2 \coloneqq \mathrm{N_C}(v_1, c_1, c_2)$ is finite, enumerable, but exponentially large. Without an efficient equilibrium representation, an attempt to solve the equilibrium is unlikely in PSPACE.

The following theorem provides a positive result on representing ϵ-Nash equilibria. It shows that if there is a polynomial-time algorithm that solves a convex program involving the payoff bounds ((4.19) below), then there is an approximate Nash equilibrium $(\boldsymbol{x}^*, \boldsymbol{Y})$ whose representation size does not depend on the size of T_1^2, and evaluating $\boldsymbol{Y}(u)$ for any $u \in T_1^2$ is also in polynomial time. Consequently, finding an approximate Nash equilibrium can be in PSPACE assuming T_1^2 is finite and enumerable by brute-force search of all possible formulations compatible with (4.20) below.

Theorem 4.7 (Compact representation of approximate Nash equilibria) *In an instance of a two-player fully-capability-aware game as defined in Definitions 4.3 and 4.4, assume without loss of generality that the players' capability levels satisfy $c_1 \leq c_2$. Let (u_1, v_1) and (u_2, v_2) be the perceived payoff functions of the players. Let $(\boldsymbol{x}^*, \boldsymbol{Y})$ be an ϵ-Nash equilibrium of this instance with $\epsilon \in \mathbb{R}_{\geq 0}$. Define $T_1^2 \coloneqq \mathrm{N_C}(v_1, c_1, c_2)$.*

For $u \in T_1^2$ and $t \in \mathbb{R}$, define an oracle $\mathfrak{O}(u,\, t)$ that returns an arbitrary solution $\boldsymbol{y}$ subject to the following constraints:

$$\begin{aligned} &\mathfrak{O}(u,\, t) \coloneqq \boldsymbol{y} \\ \textit{subject to}\quad &\boldsymbol{y} \in \Delta_{\mathcal{S}_2} \\ &\mathcal{P}^-(u,\, c_2,\, \boldsymbol{x}^*,\, \boldsymbol{y}) \geq \max_{\boldsymbol{y}' \in \Delta_{\mathcal{S}_2}} \mathcal{P}^-(u,\, c_2,\, \boldsymbol{x}^*,\, \boldsymbol{y}') - \epsilon - \epsilon' \\ &\mathcal{P}^-(u_1,\, c_1,\, \boldsymbol{x}^*,\, \boldsymbol{y}) \geq t \end{aligned} \tag{4.19}$$

Take an arbitrary value $\epsilon' \in \mathbb{R}_+$. Then there exists a function $\boldsymbol{Y}' : T_1^2 \mapsto \Delta_{\mathcal{S}_2}$ such that $(\boldsymbol{x}^*,\, \boldsymbol{Y}')$ is an $(\epsilon + \epsilon')$-Nash equilibrium of this instance, where $\boldsymbol{Y}'(\cdot)$ is defined as the following with game-dependent parameters $v_i^* \in T_1^2$ and $\boldsymbol{y}_i^* \in \Delta_{\mathcal{S}_2}$ index by $i \in \mathcal{R}$ for $\mathcal{R} \coloneqq [\![1,\, |\mathcal{S}_1| + 1]\!]$, and $f^* \in \mathbb{R}$:

$$\boldsymbol{Y}'(u) \coloneqq \begin{cases} \boldsymbol{y}_i^* & \text{if } u = v_i^* \text{ for some } i \in \mathcal{R} \\ \mathfrak{O}(u,\, f^*) & \text{if } u \in T_1^2 \setminus \{v_i^* \mid i \in \mathcal{R}\} \end{cases} \tag{4.20}$$

Proof. For a response function $\boldsymbol{Y} : T_1^2 \mapsto \Delta_{\mathcal{S}_2}$, define $f_{\boldsymbol{Y}} : \Delta_{\mathcal{S}_1} \mapsto \mathbb{R}$ as in (4.8), and apply the expression of the payoff bounds given by Theorem 2.7:

$$\begin{aligned} &f_{\boldsymbol{Y}}(\boldsymbol{x}) \coloneqq \inf_{v_2' \in T_1^2} \mathcal{P}^-(u_1,\, c_1,\, \boldsymbol{x},\, \boldsymbol{Y}(v_2')) = \inf_{v_2' \in T_1^2} \min_{\boldsymbol{s} \in \mathcal{C}\left(u_1, c_1 \,\middle|\, \boldsymbol{Y}(v_2')\right)} \boldsymbol{s}^\intercal \boldsymbol{x} = \min_{\boldsymbol{s} \in S_{\boldsymbol{Y}}} \boldsymbol{s}^\intercal \boldsymbol{x} \\ &\text{where } S_{\boldsymbol{Y}} \coloneqq \operatorname{conv}\operatorname{cl} \bigcup_{v_2' \in T_1^2} \mathcal{C}\left(u_1,\, c_1 \mid \boldsymbol{Y}(v_2')\right) \end{aligned} \tag{4.21}$$

Of note, the reason that we can apply the convex hull and closure operators in $S_{\boldsymbol{Y}}$ is similar to the proof of Theorem 2.7. The set $S_{\boldsymbol{Y}}$ is bounded due to the bounded concretization property (Definition 2.2).

Define $\bar{f} \coloneqq \max_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} f_{\boldsymbol{Y}}(\boldsymbol{x})$ as the first player's exact Nash value given $\boldsymbol{Y}(\cdot)$. With the minimax theorem, $\bar{f}$ can be rewritten as:

$$\begin{aligned} &\bar{f} = \max_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} f_{\boldsymbol{Y}}(\boldsymbol{x}) = \max_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} \min_{\boldsymbol{s} \in S_{\boldsymbol{Y}}} \boldsymbol{s}^\intercal \boldsymbol{x} = \min_{\boldsymbol{s} \in S_{\boldsymbol{Y}}} \max_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} \boldsymbol{s}^\intercal \boldsymbol{x} = \max_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} \boldsymbol{s}_{\boldsymbol{Y}}^{*\intercal} \boldsymbol{x} \\ &\text{where } \boldsymbol{s}_{\boldsymbol{Y}}^{*\intercal} \coloneqq \operatorname*{argmin}^*_{\boldsymbol{s} \in S_{\boldsymbol{Y}}} \max_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} \boldsymbol{s}^\intercal \boldsymbol{x} \end{aligned} \tag{4.22}$$

Carathéodory's theorem [ECK93] states that there are points $\boldsymbol{s}_{\boldsymbol{Y}}^{(i)} \in \mathbb{R}^{|\mathcal{S}_1|}$ indexed by $i \in \mathcal{R}$ with $\mathcal{R} \coloneqq [\![1,\, |\mathcal{S}_1| + 1]\!]$ such that $\boldsymbol{s}_{\boldsymbol{Y}}^* = \sum_{i \in \mathcal{R}} \alpha_i \boldsymbol{s}_{\boldsymbol{Y}}^{(i)}$ for $\boldsymbol{\alpha} \in \Delta_{\mathcal{R}}$ and $\boldsymbol{s}_{\boldsymbol{Y}}^{(i)} \in \operatorname{cl} \bigcup_{v_2' \in T_1^2} \mathcal{C}\left(u_1,\, c_1 \mid \boldsymbol{Y}(v_2')\right)$.

Given $\epsilon' > 0$, there exists $v_i^* \in T_1^2$ and $\boldsymbol{s}_{\boldsymbol{Y}}^{(i)\prime} \in \mathcal{C}\left(u_1,\, c_1 \mid \boldsymbol{Y}(v_i^*)\right)$ for $i \in \mathcal{R}$ such that $\left\|\boldsymbol{s}_{\boldsymbol{Y}}^{(i)\prime} - \boldsymbol{s}_{\boldsymbol{Y}}^{(i)}\right\|_2 \leq \epsilon'$. Define $\boldsymbol{s}_{\boldsymbol{Y}}' \coloneqq \sum_{i \in \mathcal{R}} \alpha_i \boldsymbol{s}_{\boldsymbol{Y}}^{(i)\prime}$. Then we have

$$\begin{aligned} &\forall \boldsymbol{x} \in \Delta_{\mathcal{S}_1} : \\ &\left|\boldsymbol{s}_{\boldsymbol{Y}}'^{\intercal} \boldsymbol{x} - \boldsymbol{s}_{\boldsymbol{Y}}^{*\intercal} \boldsymbol{x}\right| \leq \sum_{i \in \mathcal{R}} \alpha_i \left|\boldsymbol{s}_{\boldsymbol{Y}}^{(i)\prime\intercal} \boldsymbol{x} - \boldsymbol{s}_{\boldsymbol{Y}}^{(i)\intercal} \boldsymbol{x}\right| \leq \sum_{i \in \mathcal{R}} \alpha_i \left\|\boldsymbol{s}_{\boldsymbol{Y}}^{(i)\prime} - \boldsymbol{s}_{\boldsymbol{Y}}^{(i)}\right\|_2 \|\boldsymbol{x}\|_2 \leq \sum_{i \in \mathcal{R}} \alpha_i \epsilon' = \epsilon' \end{aligned} \tag{4.23}$$

Let $S^\circ_{\boldsymbol{Y}} \coloneqq \operatorname{conv}\operatorname{cl}\bigcup_{i\in\mathcal{R}} \mathcal{C}\left(u_1,\, c_1 \mid \boldsymbol{Y}(v_i^*)\right)$. Then $\boldsymbol{s}'_{\boldsymbol{Y}} \in S^\circ_{\boldsymbol{Y}}$ and $S^\circ_{\boldsymbol{Y}} \subseteq S_{\boldsymbol{Y}}$.

The response function $\boldsymbol{Y}'(\cdot)$ is constructed so that it agrees with $\boldsymbol{Y}(\cdot)$ on $S^\circ_{\boldsymbol{Y}}$. Let $\boldsymbol{y}_i^* \coloneqq \boldsymbol{Y}(v_i^*)$ for $i \in \mathcal{R}$. Let $f^* \coloneqq f_{\boldsymbol{Y}}(\boldsymbol{x}^*)$. For $u \in T_1^2 \setminus \{v_i^* \mid i \in \mathcal{R}\}$, since $\boldsymbol{Y}(u)$ is a solution to (4.19) given $t = f^*$, the oracle $\mathfrak{O}(u,\, f^*)$ is well-defined. Define $\boldsymbol{Y}'(\cdot)$ as in (4.20). Then $\boldsymbol{Y}'(v_i^*) = \boldsymbol{Y}(v_i^*)$ for $i \in \mathcal{R}$. Since $(\boldsymbol{x}^*,\, \boldsymbol{Y})$ is an ϵ-Nash equilibrium, we have $\bar{f} - \epsilon \le f^* \le \bar{f}$. Given $\boldsymbol{Y}'(\cdot)$, define $f_{\boldsymbol{Y}'}(\cdot)$ and $S_{\boldsymbol{Y}'}$ as in (4.21). We then have $S^\circ_{\boldsymbol{Y}} \subseteq S_{\boldsymbol{Y}'}$, which yields

$$\begin{aligned}\max_{\boldsymbol{x}\in\Delta_{\mathcal{S}_1}} f_{\boldsymbol{Y}'}(\boldsymbol{x}) &= \max_{\boldsymbol{x}\in\Delta_{\mathcal{S}_1}} \min_{\boldsymbol{s}\in S_{\boldsymbol{Y}'}} \boldsymbol{s}^\intercal \boldsymbol{x} \le \max_{\boldsymbol{x}\in\Delta_{\mathcal{S}_1}} \min_{\boldsymbol{s}\in S^\circ_{\boldsymbol{Y}}} \boldsymbol{s}^\intercal \boldsymbol{x} \\ &\le \max_{\boldsymbol{x}\in\Delta_{\mathcal{S}_1}} \boldsymbol{s}'^\intercal_{\boldsymbol{Y}} \boldsymbol{x} \le \max_{\boldsymbol{x}\in\Delta_{\mathcal{S}_1}} \boldsymbol{s}^{*\intercal}_{\boldsymbol{Y}} \boldsymbol{x} + \epsilon' = \bar{f} + \epsilon'\end{aligned} \tag{4.24}$$

Due to the definition of $\mathfrak{O}(\cdot)$ in (4.19), we have

$$\forall u \in T_1^2 \setminus \{v_i^* \mid i \in \mathcal{R}\} :\ \mathcal{P}^-(u_1,\, c_1,\, \boldsymbol{x}^*,\, \boldsymbol{Y}'(u)) \ge f^* \ge \bar{f} - \epsilon \tag{4.25}$$

For $i \in \mathcal{R}$, since $\mathcal{C}\left(u_1,\, c_1 \mid \boldsymbol{Y}'(v_i^*)\right) \subseteq S^\circ_{\boldsymbol{Y}} \subseteq S_{\boldsymbol{Y}}$, we have

$$\mathcal{P}^-(u_1,\, c_1,\, \boldsymbol{x}^*,\, \boldsymbol{Y}'(v_i^*)) = \min_{\boldsymbol{s}\in\mathcal{C}\left(u_1,\, c_1 \,\middle|\, \boldsymbol{Y}'(v_i^*)\right)} \boldsymbol{s}^\intercal \boldsymbol{x}^* \ge \min_{\boldsymbol{s}\in S_{\boldsymbol{Y}}} \boldsymbol{s}^\intercal \boldsymbol{x}^* = f_{\boldsymbol{Y}}(\boldsymbol{x}^*) \ge \bar{f} - \epsilon \tag{4.26}$$

Combining (4.25) and (4.26) yields $f_{\boldsymbol{Y}'}(\boldsymbol{x}^*) \ge \bar{f} - \epsilon$, which further implies that $f_{\boldsymbol{Y}'}(\boldsymbol{x}^*) \ge \max_{\boldsymbol{x}\in\Delta_{\mathcal{S}_1}} f_{\boldsymbol{Y}'}(\boldsymbol{x}) - (\epsilon + \epsilon')$ when combined with (4.24). Thus $\boldsymbol{x}^*$ is an $(\epsilon + \epsilon')$-strategy for the first player. The definitions of $\boldsymbol{Y}'(\cdot)$ and $\mathfrak{O}(\cdot)$ ensure that $\boldsymbol{Y}'(\cdot)$ is an $(\epsilon + \epsilon')$-strategy for the second player. Therefore, $(\boldsymbol{x}^*,\, \boldsymbol{Y}')$ is an $(\epsilon + \epsilon')$-Nash equilibrium. □

4.2.4 Extended settings and equilibrium concepts

This thesis is only an initial exploration of limited-perception games. There are many potential directions for future research. In this section, I discuss some extended settings and equilibrium concepts that may be of interest.

4.2.4.1 Privately solvable Nash equilibrium with multiple players

As discussed in Section 4.2.2, if a fully-capability-aware game has more than two players, then the lower-capability players may be unable to solve Nash equilibrium using their private information. With a communication mechanism, they can agree on the equilibrium if higher-capability players' agreement is common knowledge, and the agreement is extended from higher-capability players to lower-capability players.

It may be desirable to define an equilibrium concept that all players can solve from their private information. One possibility is to constrain the instance to have at most two distinct capability levels (unless some capability levels are redundant; i.e., $\alpha(u,\, c_1) = \alpha(u,\, c_2)$ for all $u \in U$) so that all players can solve Nash equilibrium defined in Definition 4.7. Another possibility is to extend the definition of Nash equilibrium to include the uncertainty of other players' reasoning of higher-capability player's perceptions as detailed below.

Let's consider a three-player game instance with capability levels $c_1 < c_2 < c_3$. Since player 1 does not know player 2's perception of player 3's payoff function, which is required to

define player 2's best response, player 1 models player 2 with a response function $\boldsymbol{Y}(v_2^{2\prime}, v_3^{2\prime})$ that is a best response of player 2 assuming that player 2's perception of player 2's and player 3's payoff functions are $v_2^{2\prime}$ and $v_3^{2\prime}$, respectively. Similar to Definition 4.7, we can define Nash equilibrium as follows:

Definition 4.8 (Privately solvable Nash equilibrium of three-player games) In an instance of a three-player fully-capability-aware game as defined in Definition 4.3, assume without loss of generality that $c_1 \leq c_2 \leq c_3$. As in Definition 2.3, let $v_i^j \coloneqq \alpha(u_i, c_j)$ be player j's perception of player i's payoff function. A Nash equilibrium of this instance is a tuple $(\boldsymbol{x}^*, \boldsymbol{Y}, \boldsymbol{Z})$ if and only if the following conditions hold:

- $\boldsymbol{x}^* \in \Delta_{\mathcal{S}_1}$ is the mixed strategy of player 1, $\boldsymbol{Y} : \mathrm{N_C}\left(v_2^1, c_1, c_2\right) \times \mathrm{N_C}\left(v_3^1, c_1, c_2\right) \mapsto \Delta_{\mathcal{S}_2}$ is the response function of player 2, and $\boldsymbol{Z} : \mathrm{N_C}\left(v_2^1, c_1, c_2\right) \times \mathrm{N_C}\left(v_3^1, c_1, c_3\right) \mapsto \Delta_{\mathcal{S}_3}$ is the response function of player 3. Of note, $\mathrm{N_C}\left(\cdot\right)$ denotes the narrow concretization set defined in Definition 4.6.

- The first player has no incentive to deviate from $\boldsymbol{x}^*$:

$$\boldsymbol{x}^* \in \operatorname*{argmax}_{\boldsymbol{x}\in\Delta_{\mathcal{S}_1}} \inf_{\substack{v_2^{2\prime}\in\mathrm{N_C}\left(v_2^1, c_1, c_2\right)\\ v_3^{3\prime}\in\mathrm{N_C}\left(v_3^1, c_1, c_3\right)}} \mathcal{P}^-\left(v_1^1, c_1, \boldsymbol{x}, \boldsymbol{Y}\left(v_2^{2\prime}, \alpha\left(v_3^{3\prime}, c_2\right)\right), \boldsymbol{Z}\left(v_2^{2\prime}, v_3^{3\prime}\right)\right) \tag{4.27}$$

- The second player's best response is encoded in $\boldsymbol{Y}(\cdot)$:

$$\begin{aligned} &\forall v_2^{2\prime} \in \mathrm{N_C}\left(v_2^1, c_1, c_2\right) : \forall v_3^{2\prime} \in \mathrm{N_C}\left(v_3^1, c_1, c_2\right) : \\ &\boldsymbol{Y}\left(v_2^{2\prime}, v_3^{2\prime}\right) \in \operatorname*{argmax}_{\boldsymbol{y}\in\Delta_{\mathcal{S}_2}} \inf_{v_3^{3\prime}\in\mathrm{N_C}\left(v_3^{2\prime}, c_1, c_3\right)} \mathcal{P}^-\left(v_2^{2\prime}, c_2, \boldsymbol{x}^*, \boldsymbol{y}, \boldsymbol{Z}\left(v_2^{2\prime}, v_3^{3\prime}\right)\right) \end{aligned} \tag{4.28}$$

- The third player's best response is encoded in $\boldsymbol{Z}(\cdot)$:

$$\begin{aligned} &\forall v_2^{2\prime} \in \mathrm{N_C}\left(v_2^1, c_1, c_2\right) : \forall v_3^{3\prime} \in \mathrm{N_C}\left(v_3^1, c_1, c_3\right) : \\ &\boldsymbol{Z}\left(v_2^{2\prime}, v_3^{3\prime}\right) \in \operatorname*{argmax}_{\boldsymbol{z}\in\Delta_{\mathcal{S}_3}} \mathcal{P}^-\left(v_3^{3\prime}, c_3, \boldsymbol{x}^*, \boldsymbol{Y}\left(v_2^{2\prime}, \alpha\left(v_3^{3\prime}, c_2\right)\right), \boldsymbol{z}\right) \end{aligned} \tag{4.29}$$

At the equilibrium, player 1 samples an action according to $\boldsymbol{x}^*$, player 2 samples an action according to $\boldsymbol{y}^*$, and player 3 samples an action according to $\boldsymbol{z}^*$, where

$$\begin{aligned} \boldsymbol{y}^* &\coloneqq \boldsymbol{Y}\left(v_2^2, v_3^2\right) \\ \boldsymbol{z}^* &\coloneqq \boldsymbol{Z}\left(\alpha\left(v_2^3, c_2\right), v_3^3\right) = \boldsymbol{Z}\left(v_2^2, v_3^3\right) \end{aligned} \tag{4.30}$$

◁

Definition 4.8 considers the three-player case. The definition can be extended to more players by adding more parameters to the response functions. The general-case definition involves more convoluted notation and is omitted for brevity. The existence of Nash equilibrium can be proved by extending the proof of Theorem 4.5.

4.2.4.2 Stackelberg equilibrium in fully-capability-aware games

The Stackelberg equilibrium is defined for two-player games where one player is a leader and the other is a follower [Sta+10]. The leader commits to a strategy first. The follower observes the leader's strategy and then chooses their response. Similar to Nash equilibrium case, if the leader has a lower capability, they need to reason about responses of the follower for multiple perceived games consistent with their own perceived game. In conventional finite normal-form games, a pure Stackelberg equilibrium exists. This section considers the case where both players are allowed to use mixed strategies; the pure strategy case is a special and much easier case by constraining the leader's strategy space to a finite set.

Definition 4.9 (Stackelberg equilibrium of fully-capability-aware games)
In an instance of a two-player fully-capability-aware Stackelberg game as defined in Definitions 4.3 and 4.4, assume without loss of generality that the first player is the leader, and the second player is the follower. Let $T_1^2 \coloneqq \mathrm{N_C}\left(v_1,\, c_1,\, c_2\right)$ be the set of possible perceived payoff functions of the follower from the leader's perspective. The tuple $(\boldsymbol{x}^*,\, Y)$ is a Stackelberg equilibrium of this instance if and only if the following conditions hold:

- $\boldsymbol{x}^* \in \Delta_{\mathcal{S}_1}$ is the mixed strategy of the leader.
- $Y : T_1^2 \times \Delta_{\mathcal{S}_1} \rightrightarrows \Delta_{\mathcal{S}_2}$ is the response function of the follower given an assumed perceived game and the leader's mixed strategy, which satisfies

$$\forall (v_2',\, \boldsymbol{x}) \in T_1^2 \times \Delta_{\mathcal{S}_1} :\ \emptyset \neq Y(v_2',\, \boldsymbol{x}) \subseteq \operatorname*{argmax}_{\boldsymbol{y}' \in \Delta_{\mathcal{S}_2}} \mathcal{P}^-(v_2',\, c_2,\, \boldsymbol{x},\, \boldsymbol{y}') \tag{4.31}$$

- The leader obtains maximum worst payoff using $\boldsymbol{x}^*$:

$$\begin{aligned} g_1(\boldsymbol{x}^*) &= \sup_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} g_1(\boldsymbol{x}) \\ \text{where } g_1(\boldsymbol{x}) &\coloneqq \inf_{\boldsymbol{y} \in J_2(\boldsymbol{x})} \mathcal{P}^-(u_1,\, c_1,\, \boldsymbol{x},\, \boldsymbol{y}), \quad J_2(\boldsymbol{x}) \coloneqq \bigcup_{v_2' \in T_1^2} Y(v_2',\, \boldsymbol{x}) \end{aligned} \tag{4.32}$$

  The set $J_2(\boldsymbol{x})$ is the set of the follower's possible responses given the leader's strategy.

The function $g_1(\cdot)$ is called the *Stackelberg objective function* of the leader; the value of $\sup_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} g_1(\boldsymbol{x})$ is called the *Stackelberg value* of the leader. At the equilibrium, the leader commits to a mixed strategy $\boldsymbol{x}^*$, and the follower responds with a mixed strategy $\boldsymbol{y}^*$ such that $\boldsymbol{y}^* \in Y(v_2,\, \boldsymbol{x}^*)$. ◁

Remark 4.6 The Stackelberg value is always finite since $\mathcal{P}^-(u_1,\, c_1,\, \boldsymbol{x},\, \boldsymbol{y})$ is bounded for $(\boldsymbol{x},\, \boldsymbol{y}) \in \Delta_{\mathcal{S}_1} \times \Delta_{\mathcal{S}_2}$ due to the bounded concretization property of the payoff perception function (Definition 2.2).

However, given an arbitrary correspondence $Y(\cdot)$, Stackelberg strategy $\boldsymbol{x}^*$ does not necessarily exist. Even under the assumptions that $Y(u,\, \boldsymbol{x})$ is upper hemicontinuous (uhc) in both u and $\boldsymbol{x}$ (Y can not be continuous for a nondegenerate payoff bound) and that $\mathrm{N_C}\left(v_1,\, c_1,\, c_2\right)$ is compact, one can prove that $J_2(\cdot)$ is uhc. Then $g_1(\cdot)$ is only lower hemicontinuous [AB13, Lemma 17.30], which does not guarantee the attainability of the supremum over $\Delta_{\mathcal{S}_1}$ (unless

when T_1^2 is finite). Characterizing when a Stackelberg equilibrium exists is left as an open problem.

Of note, an ϵ-approximate Stackelberg equilibrium for $\epsilon \in \mathbb{R}_+$ (defined as an equilibrium from which the leader can not deviate to improve their payoff by more than ϵ) always exists. From a computational perspective, the existence of approximate equilibrium is sufficient since the precise real-valued equilibrium, even if it exists, is not computable or representable in general.

4.2.4.3 Games with partial capability awareness

Let's consider *partially capability-aware* players who only know their own capability levels but not the capability levels of other players. A *partially-capability-aware game* is defined with a partition of the player set $\mathcal{N} = \mathcal{N}_f \cup \mathcal{N}_p$ where players in $\mathcal{N}_f$ know the complete capability profile, players in $\mathcal{N}_p$ only know their individual capability levels, and those conditions are common knowledge. A partially capability-aware player needs to reason about possible capability profiles besides perceptions of other players.

Similar to Section 4.2.4.1, a Nash equilibrium can be defined by extending the response function to include uncertainties about the capability levels of other players. A definition of the Nash equilibrium in two-player games with one partially capability-aware player is presented below, which can be extended to more players or games with all players being partially capability-aware.

Definition 4.10 (Nash equilibrium of two-player partially-capability-aware games) In an instance of a two-player partially-capability-aware game with the capability profile (c_1, c_2), assume that the first player is partially capability-aware and the second player is fully capability-aware. Let $U \coloneqq \mathbb{R}^{\mathcal{S}_1 \times \mathcal{S}_2}$ be the set of all possible payoff functions. Let (u_1, v_1) and (u_2, v_2) be the perceived payoff functions of the first and the second player, respectively.

Let $\mathrm{N_C}(\cdot)$ denote the narrow concretization set defined in Definition 4.6. Define $u' \coloneqq \alpha(u_1, 1)$ and $v' \coloneqq \alpha(v_1, 1)$ as the least accurate perceptions of the players' payoff functions. Define $T_1 \coloneqq \alpha^{-1}(u', 1)$ and $T_2 \coloneqq \alpha^{-1}(v', 1)$ as the sets of possible perceived payoff functions of the opponent for the two players without knowledge of the capability levels. For a set $T \subseteq U$, let $\alpha(T, c) \coloneqq \{\alpha(u, c) \mid u \in T\}$ be the set of a capability-c player's perceptions of the items in T.

A Nash equilibrium of this instance is a pair $(\boldsymbol{X}, \boldsymbol{Y})$ if and only if the following conditions hold:

- $\boldsymbol{X} : T_1 \times T_2 \mapsto \Delta_{\mathcal{S}_1}$ is the response function of the first player, and $\boldsymbol{Y} : \overline{\mathbb{Z}}_+ \times T_1 \times T_2 \mapsto \Delta_{\mathcal{S}_2}$ is the response function of the second player.

- The first player has no incentive to deviate from $\boldsymbol{X}$:

$$\forall u_1' \in \alpha(T_1,\, c_1) :\, \forall v_1' \in \alpha(T_2,\, c_1) : \tag{4.33}$$
$$\boldsymbol{X}(u_1',\, v_1') \in \operatorname*{argmax}_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} \left(\inf_{\substack{c_2' \in \overline{\mathbb{Z}}_+ \\ u_2' \in \mathrm{N_C}\left(u_1', c_1, c_2'\right) \\ v_2' \in \mathrm{N_C}\left(v_1', c_1, c_2'\right)}} \mathcal{P}^-(u_1',\, c_1,\, \boldsymbol{x},\, \boldsymbol{Y}(c_2',\, u_2',\, v_2')) \right)$$

- The second player has no incentive to deviate from $\boldsymbol{Y}$:

$$\forall c_2' \in \overline{\mathbb{Z}}_+ :\, \forall u_2' \in \alpha(T_1,\, c_2') :\, \forall v_2' \in \alpha(T_2,\, c_2') : \tag{4.34}$$
$$\boldsymbol{Y}(c_2',\, u_2',\, v_2') \in \operatorname*{argmax}_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \left(\inf_{\substack{u_1' \in \mathrm{N_C}\left(u_2', c_2', c_1\right) \\ v_1' \in \mathrm{N_C}\left(v_2', c_2', c_1\right)}} \mathcal{P}^-(v_2',\, c_2',\, \boldsymbol{X}(u_1',\, v_1'),\, \boldsymbol{y}) \right)$$

At the equilibrium, the first player samples an action according to $\boldsymbol{x}^*$, and the second player samples an action according to $\boldsymbol{y}^*$, where

$$\begin{aligned} \boldsymbol{x}^* &\coloneqq \boldsymbol{X}(u_1,\, v_1) \\ \boldsymbol{y}^* &\coloneqq \boldsymbol{Y}(c_2,\, u_2,\, v_2) \end{aligned} \tag{4.35}$$

◁

Remark 4.7 In Definition 4.10, both players can solve the equilibrium using their private information because both players can define T_1 and T_2, and (4.33) and (4.34) only depend on c_1, T_1, and T_2. The existence of the equilibrium can be proved similarly to Theorem 4.5.

The idea can be generalized to games that simultaneously contain capability-obvious and (fully or partially) capability-aware players. A capability-obvious player solves a standard Nash equilibrium from their perceived game. A capability-aware player reasons with the possible perceived payoff functions and/or capability profiles of the other players.

4.2.4.4 Games with belief on true payoff functions

Our exploration has focused on games without any probabilistic prior. This section briefly demonstrates an extension to games with a shared common prior on the distribution of true payoff functions. Assume all players share a probability density $p^* : U \mapsto \mathbb{R}_{\geq 0}$ over the true payoff functions such that $\int_U p^*(u)\,\mathrm{d}u = 1$ for a proper measure over the set of all possible payoff functions U. The density function $p^*(\cdot)$ and the fact that all payoff functions are drawn independently from $p^*(\cdot)$ are common knowledge among the players. The narrow concretization set given in Definition 4.6 can then be extended to a probability distribution

as follows:

$$p_{\mathrm{N_C}}(u \mid v, c_1, c_2) \coloneqq \begin{cases} 0 & \text{if } u \notin \mathrm{N_C}(v, c_1, c_2) \\ \frac{1}{Z} p^*(u) & \text{otherwise} \end{cases} \tag{4.36}$$
$$\text{where } Z \coloneqq \int_{\mathrm{N_C}(v, c_1, c_2)} p^*(u)\,\mathrm{d}u$$

Players can then reason about the expected outcome instead of the worst outcome by replacing all infimum operators in the equilibrium definitions with expectations over the true payoff functions. For example, in the definition of the two-player Nash equilibrium in Definition 4.5, (4.7) and (4.9) can be replaced with the following:

$$\begin{aligned} \boldsymbol{x}^* &\in \operatorname*{argmax}_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} f_1(\boldsymbol{x}) \\ \forall v_2' \in T_1^2 :\ \boldsymbol{Y}(v_2') &\in \operatorname*{argmax}_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} f_2(\boldsymbol{y} \mid v_2') \end{aligned} \tag{4.37}$$

with the objective functions $f_1(\cdot)$ and $f_2(\cdot \mid \cdot)$ defined as:

$$\begin{aligned} f_1(\boldsymbol{x}) &\coloneqq \int_U \left(\int_U u'(\boldsymbol{x}, \boldsymbol{Y}(v')) p_{\mathrm{N_C}}(v' \mid v_1, c_1, c_2)\,\mathrm{d}v' \right) p_{\mathrm{N_C}}(u' \mid u_1, c_1, \infty)\,\mathrm{d}u' \\ f_2(\boldsymbol{y} \mid v_2') &\coloneqq \int_U v'(\boldsymbol{x}^*, \boldsymbol{y}) p_{\mathrm{N_C}}(v' \mid v_2', c_2, \infty)\,\mathrm{d}v' \end{aligned} \tag{4.38}$$

Since the true payoff functions u^* and v^* are drawn independently from $p^*(\cdot)$, players can simplify $f_1(\cdot)$ and $f_2(\cdot \mid \cdot)$ in (4.38) by factorizing out the expected payoff functions and the expected opponent's strategy:

$$\begin{aligned} f_1(\boldsymbol{x}) = \bar{u}(\boldsymbol{x}, \bar{\boldsymbol{y}}), \quad f_2(\boldsymbol{y} \mid v_2') &= \overline{v_{v_2'}}(\boldsymbol{x}^*, \boldsymbol{y}) \\ \text{where } \bar{u} \coloneqq \mathbb{E}_{u \sim p_{\mathrm{N_C}}(\cdot \mid u_1, c_1, \infty)}[u], \quad \bar{\boldsymbol{y}} &\coloneqq \mathbb{E}_{v \sim p_{\mathrm{N_C}}(\cdot \mid v_1, c_1, c_2)}[\boldsymbol{Y}(v)] \\ \overline{v_{v_2'}} &\coloneqq \mathbb{E}_{v \sim p_{\mathrm{N_C}}(\cdot \mid v_2', c_2, \infty)}[v] \end{aligned} \tag{4.39}$$

Eqs. (4.37) and (4.39) may be efficiently solvable for certain payoff perception functions and payoff function distributions. Identifying such cases and developing efficient solution or approximation algorithms is left for future work.

4.3 Two-player fully-capability-aware zero-sum games

4.3.1 Zero-sum games and equilibrium concepts

Section 4.2 establishes the rational behavior in fully-capability-aware games and shows the hardness of solving the equilibrium in the general case. In conventional bimatrix games, the zero-sum Nash equilibrium is easier to solve than the general-sum case due to the minimax theorem. It is natural to ask whether the zero-sum case is easier to solve for two-player fully-capability-aware games. This section provides a partially positive answer. By imposing

mild assumptions on the payoff perception function, a zero-sum equilibrium can be efficiently computed with nonsmooth convex optimization.

For an arbitrary payoff perception function, even if the true game is zero-sum, the perceived game may be nonzero-sum. To address this issue, we require that the payoff perception function be odd so that the perceived game is also zero-sum.

Definition 4.11 (Odd payoff perception functions)
A payoff perception function $\alpha : U \times \mathbb{Z}_+ \to U$ is called *odd* if and only if

$$\forall u \in U, \quad \forall c \in \mathbb{Z}_+, \quad \alpha(-u, c) = -\alpha(u, c) \tag{4.40}$$

◁

Example 4.2 (Games with odd payoff perception functions)
It is straightforward to verify that the payoff perception functions of the masked payoff game (Example 2.1) and the quantized payoff game (Example 2.2) are odd. ◁

Theorem 4.8 (Properties of odd payoff perception functions)
For an odd payoff perception function $\alpha : U \times \mathbb{Z}_+ \to U$, the following properties hold:

- $\bar{c}(-u) = \bar{c}(u)$ *for any* $u \in U$.
- $\alpha^{-1}(-u, c) = \{-v \mid v \in \alpha^{-1}(u, c)\}$ *for any* $u \in U$ *and* $c \in \mathbb{Z}_+$.
- *For* $v \in U$, $c \in \mathbb{Z}_+$, *and* $\boldsymbol{\sigma}_k \in \Delta_{\mathcal{S}_k}$, *the payoff bounds satisfy*

$$\begin{aligned} \mathcal{P}^-\big(-v, c, (\boldsymbol{\sigma}_k)_{k\in\mathcal{N}}\big) &= -\mathcal{P}^+\big(v, c, (\boldsymbol{\sigma}_k)_{k\in\mathcal{N}}\big) \\ \mathcal{P}^+\big(-v, c, (\boldsymbol{\sigma}_k)_{k\in\mathcal{N}}\big) &= -\mathcal{P}^-\big(v, c, (\boldsymbol{\sigma}_k)_{k\in\mathcal{N}}\big) \end{aligned} \tag{4.41}$$

Proof. Omitted since all properties are straightforward to verify using the definitions. □

Definition 4.12 (TFZolp-Game and its notation)
In a two-player fully-capability-aware zero-sum game (TFZolp-Game for short), besides requirements given in Definition 4.3, there are two additional conditions:

- Players know that the game is zero-sum as common knowledge.
- The payoff perception function is odd (see Definition 4.11).

The following notation is used to simplify the presentation in addition to Definition 4.4:

- u^* and $-u^*$ for the first and the second player's true payoff functions, respectively.
- $(u_1, -u_1)$ and $(u_2, -u_2)$ for the players' respective perceived payoff functions; i.e.,

$$u_1 \coloneqq \alpha(u^*, c_1), \quad u_2 \coloneqq \alpha(u^*, c_2)$$

◁

Similar to the general-sum Nash equilibrium (Definition 4.5), in a zero-sum game, the lower-capability player reasons over all possible true payoff functions and the opponent's corresponding best response to determine their strategy. It differs from the general-sum case in that a possible perception of the higher-capability player is tied to the lower-capability player's true payoff function. The following definition formalizes the zero-sum Nash equilibrium.

Definition 4.13 (Nash equilibrium of TFZOLP-Game)
In an instance of a two-player fully-capability-aware zero-sum game as defined in Definition 4.12 with the capability profile (c_1, c_2), assume without loss of generality that $c_1 \leq c_2$. Let $(u_1, -u_1)$ and $(u_2, -u_2)$ be the perceived payoff functions of the first and the second player, respectively. Define $T_1^2 \coloneqq \mathrm{N_C}(u_1, c_1, c_2)$ where $\mathrm{N_C}(\cdot)$ is the narrow concretization set defined in Definition 4.6.

A pair $(\boldsymbol{x}^*, \boldsymbol{Y})$ is a Nash equilibrium of this instance if and only if the following conditions hold:

- $\boldsymbol{x}^* \in \Delta_{\mathcal{S}_1}$ is a mixed strategy for the first player, and $\boldsymbol{Y} : T_1^2 \mapsto \Delta_{\mathcal{S}_2}$ is the response function of the second player.
- The second player has no incentive to deviate from $\boldsymbol{Y}(\cdot)$ given $\boldsymbol{x}^*$:

$$\forall u_2' \in T_1^2 : \ \boldsymbol{Y}(u_2') \in \operatorname*{argmax}_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^-(-u_2', c_2, \boldsymbol{x}^*, \boldsymbol{y}) = \operatorname*{argmin}_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^+(u_2', c_2, \boldsymbol{x}^*, \boldsymbol{y}) \tag{4.42}$$

- Let $f_{\boldsymbol{Y}}(\boldsymbol{x})$ be the first player's worst payoff when they play $\boldsymbol{x}$ against $\boldsymbol{Y}$, defined as

$$f_{\boldsymbol{Y}}(\boldsymbol{x}) \coloneqq \inf_{u_2' \in T_1^2} \mathcal{P}^-(u_2', c_2, \boldsymbol{x}, \boldsymbol{Y}(u_2')) \tag{4.43}$$

  Then $\boldsymbol{x}^*$ is a maximizer of $f_{\boldsymbol{Y}}(\boldsymbol{x})$:

$$\boldsymbol{x}^* \in \operatorname*{argmax}_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} f_{\boldsymbol{Y}}(\boldsymbol{x}) \tag{4.44}$$

Given a Nash equilibrium $(\boldsymbol{x}^*, \boldsymbol{Y})$, the first player samples an action according to $\boldsymbol{x}^*$, and the second player samples an action according to $\boldsymbol{Y}(u_2)$. The values $f_{\boldsymbol{Y}}(\boldsymbol{x}^*)$ and $\mathcal{P}^+(u_2, c_2, \boldsymbol{x}^*, \boldsymbol{Y}(u_2))$ are the *Nash values* of the first and second player at the equilibrium, respectively. Of note, the second player tries to minimize their Nash value. ◁

Remark 4.8 In (4.43), c_2 is used in $\mathcal{P}^-(\cdot)$ because given an assumed perception of the second player as u_2', the first player narrows down the set of possible true payoff functions to those compatible with u_2' at c_2.

Definition 4.13 defines the Nash equilibrium for $c_1 \leq c_2$ for easier presentation. In some cases, it is helpful to define a Nash equilibrium in the form of two response functions $(\boldsymbol{X}, \boldsymbol{Y})$ so that the equilibrium representation does not depend on which player has the higher capability. The following two definitions formalize the symmetric Nash equilibrium.

Definition 4.14 (Perception selection function)
Given strategy spaces $\mathcal{S}_1, \ldots, \mathcal{S}_n$, let $U \coloneqq \mathbb{R}^{\mathcal{S}_1 \times \cdots \times \mathcal{S}_n}$ be the set of all possible payoff functions. Given a payoff perception function $\alpha : U \times \overline{\mathbb{Z}}_+ \to U$ (Definition 2.2), two capability levels $c_1, c_2 \in \overline{\mathbb{Z}}_+$, and two compatible perceptions $u_1, u_2 \in U$ such that $\alpha(u_1, \min\{c_1, c_2\}) = \alpha(u_2, \min\{c_1, c_2\})$, the *perception selection function* is defined as

$$\widehat{\mathcal{P}}\left(\begin{smallmatrix} u_1 & c_1 \\ u_2 & c_2 \end{smallmatrix}\right) \coloneqq \begin{cases} u_1 & \text{if } c_1 \geq c_2 \\ u_2 & \text{otherwise} \end{cases}$$ ◁

Definition 4.15 (Symmetric Nash equilibrium of TFZOLP-Game)
In an instance of a two-player fully-capability-aware zero-sum game as defined in Definition 4.12 with the capability profile (c_1, c_2), let $(u_1, -u_1)$ and $(u_2, -u_2)$ be the perceived payoff functions of the first and the second player, respectively. Define $T_1^2 \coloneqq \mathrm{N_C}(u_1, c_1, c_2)$ and $T_2^1 \coloneqq \mathrm{N_C}(u_2, c_2, c_1)$ where $\mathrm{N_C}(\cdot)$ is the narrow concretization set defined in Definition 4.6. Let $c_h \coloneqq \max\{c_1, c_2\}$.

Let $\boldsymbol{X} : T_2^1 \mapsto \Delta_{\mathcal{S}_1}$ and $\boldsymbol{Y} : T_1^2 \mapsto \Delta_{\mathcal{S}_2}$ be two response functions. Then $(\boldsymbol{X}, \boldsymbol{Y})$ is a Nash equilibrium of this instance if and only if:

$$
\begin{aligned}
&\forall u_1' \in T_2^1 : \ \boldsymbol{X}(u_1') \in \operatorname*{argmax}_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} f_1(u_1', \boldsymbol{x}) \\
&\forall u_2' \in T_1^2 : \ \boldsymbol{Y}(u_2') \in \operatorname*{argmin}_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} f_2(u_2', \boldsymbol{y}) \\
&\qquad \text{where } f_1(u_1', \boldsymbol{x}) \coloneqq \inf_{u_2' \in \mathrm{N_C}\left(u_1', c_1, c_2\right)} \mathcal{P}^-\left(\widehat{\mathcal{P}}\left(\begin{smallmatrix} u_1' & c_1 \\ u_2' & c_2 \end{smallmatrix}\right), c_h, \boldsymbol{x}, \boldsymbol{Y}(u_2')\right) \\
&\qquad\qquad f_2(u_2', \boldsymbol{y}) \coloneqq \sup_{u_1' \in \mathrm{N_C}\left(u_2', c_2, c_1\right)} \mathcal{P}^+\left(\widehat{\mathcal{P}}\left(\begin{smallmatrix} u_1' & c_1 \\ u_2' & c_2 \end{smallmatrix}\right), c_h, \boldsymbol{X}(u_1'), \boldsymbol{y}\right)
\end{aligned}
\tag{4.45}
$$

The values $f_1(u_1, \boldsymbol{X}(u_1))$ and $f_2(u_2, \boldsymbol{Y}(u_2))$ are the *Nash values* of the first and second player at the equilibrium, respectively. ◁

The existence of the Nash equilibrium can be proved similarly to Theorem 4.5.

Next, we define the Stackelberg equilibrium for the zero-sum game.

Definition 4.16 (Stackelberg equilibrium of TFZOLP-Game)
In an instance of a two-player fully-capability-aware zero-sum Stackelberg game as defined in Definition 4.12, assume without loss of generality that the first player is the leader. Define $T_1^2 \coloneqq \mathrm{N_C}(u_1, c_1, c_2)$ where $\mathrm{N_C}(\cdot)$ is the narrow concretization set defined in Definition 4.6.

The pair $(\boldsymbol{x}^*, Y)$ is a Stackelberg equilibrium of this instance if and only if the following conditions hold:

- $\boldsymbol{x}^* \in \Delta_{\mathcal{S}_1}$ is the mixed strategies of the leader.
- $Y : T_1^2 \times \Delta_{\mathcal{S}_1} \rightrightarrows \Delta_{\mathcal{S}_2}$ is the response function of the follower given an assumed perceived game and the leader's mixed strategy, which satisfies

$$
\forall (u_2', \boldsymbol{x}) \in T_1^2 \times \Delta_{\mathcal{S}_1} : \ \emptyset \neq Y(u_2', \boldsymbol{x}) \subseteq \operatorname*{argmin}_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^+(u_2', c_2, \boldsymbol{x}, \boldsymbol{y})
\tag{4.46}
$$

- The leader obtains maximum worst payoff using $\boldsymbol{x}^*$:

$$
\begin{aligned}
&g_1(\boldsymbol{x}^*) = \sup_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} g_1(\boldsymbol{x}) \\
&\text{where } g_1(\boldsymbol{x}) \coloneqq \inf_{u_2' \in \mathrm{N_C}\left(u_1, c_1, c_2\right)} \inf_{\boldsymbol{y} \in Y\left(u_2', \boldsymbol{x}\right)} \mathcal{P}^-\left(\widehat{\mathcal{P}}\left(\begin{smallmatrix} u_1 & c_1 \\ u_2' & c_2 \end{smallmatrix}\right), \max\{c_1, c_2\}, \boldsymbol{x}, \boldsymbol{y}\right)
\end{aligned}
\tag{4.47}
$$

The function $g_1(\cdot)$ is called the *Stackelberg objective function* of the leader; the value of $\sup_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} g_1(\boldsymbol{x})$ is called the *Stackelberg value* of the leader. At the equilibrium, the leader commits to a mixed strategy $\boldsymbol{x}^*$, and the follower responds with a mixed strategy $\boldsymbol{y}^*$ such that $\boldsymbol{y}^* \in Y(v_2, \boldsymbol{x}^*)$. ◁

Remark 4.9 Definition 4.16 assumes that both players believe that the follower does not use the leader's strategy to reduce their uncertainty. If this is not the case, and if $c_1 \geq c_2$ and thus $T_1^2 = \{u_2\}$, the follower should use the fact that the leader plays $\boldsymbol{x}^*$ to remove values in $\mathrm{N_C}(u_2, c_2, c_1)$ that are incompatible with $\boldsymbol{x}^*$. Let $R(\boldsymbol{x}) \subseteq U$ be the follower's reasoning of possible leader's perceptions given $\boldsymbol{x}$ as the leader's strategy. The condition in (4.46) should then be replaced with the following:

$$\forall \boldsymbol{x} \in \Delta_{\mathcal{S}_1} : \emptyset \neq Y(u_2, \boldsymbol{x}) \subseteq \operatorname*{argmin}_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \sup_{u_1' \in R(\boldsymbol{x})} \mathcal{P}^+(u_1', c_1, \boldsymbol{x}, \boldsymbol{y})$$

$$\text{(4.48)} \qquad \text{where} \quad R(\boldsymbol{x}) \coloneqq \begin{cases} R_0(\boldsymbol{x}) & \text{if } R_0(\boldsymbol{x}) \neq \emptyset \\ \mathrm{N_C}(u_2, c_2, c_1) & \text{otherwise} \end{cases}$$

$$R_0(\boldsymbol{x}) \coloneqq \left\{ u_1' \in \mathrm{N_C}(u_2, c_2, c_1) \;\middle|\; \boldsymbol{x} \in \operatorname*{argmax}_{\boldsymbol{x}' \in \Delta_{\mathcal{S}_1}} \inf_{\boldsymbol{y} \in Y(u_2, \boldsymbol{x}')} \mathcal{P}^-(u_1', c_1, \boldsymbol{x}', \boldsymbol{y}) \right\}$$

Of note, (4.48) is a recursive definition since $R_0(\cdot)$ depends on $Y(u_2, \cdot)$. The case of $R_0(\boldsymbol{x}) = \emptyset$ corresponds to noncredible leader strategies (i.e., strategies that the leader would never play since there is no function u_1' that could make it a best strategy); the follower considers all possible leader's perceptions as a fallback. Whether/when an equilibrium defined by (4.47) and (4.48) exists is left as an open problem. Specially, when $c_1 = c_2$, or more generally when $|\mathrm{N_C}(u_2, c_2, c_1)| = 1$ (i.e., when the follower has no uncertainty), we have $R(\boldsymbol{x}) = \mathrm{N_C}(u_2, c_2, c_1)$, and the Stackelberg equilibrium defined by Definition 4.16 satisfies (4.48).

Remark 4.10 Remark 4.6 argues that in the general-sum case, the Stackelberg equilibrium may not exist due to the objective function being noncontinuous. By contrast, in the zero-sum case, if the follower chooses the best response that is also favorable to the leader, then an equilibrium exists, as shown below.

Theorem 4.9 (Existence of Stackelberg equilibrium in TFZOLP-Game) *In a TFZOLP-Game instance as defined in Definition 4.12, assume without loss of generality that the first player is the leader. Let* $T_1^2 \coloneqq \mathrm{N_C}(u_1, c_1, c_2)$. *For* $u \in T_1^2$, *define the correspondences* $A_u : \Delta_{\mathcal{S}_1} \rightrightarrows \Delta_{\mathcal{S}_2}$ *and* $B_u : \Delta_{\mathcal{S}_1} \rightrightarrows \Delta_{\mathcal{S}_2}$ *as*

$$\text{(4.49)} \qquad \begin{aligned} A_u(\boldsymbol{x}) &\coloneqq \operatorname*{argmin}_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^+(u, c_2, \boldsymbol{x}, \boldsymbol{y}) \\ B_u(\boldsymbol{x}) &\coloneqq \operatorname*{argmax}_{\boldsymbol{y} \in A_u(\boldsymbol{x})} \mathcal{P}^- \left(\widehat{\mathcal{P}}\left(\begin{smallmatrix} u_1 & c_1 \\ u & c_2 \end{smallmatrix}\right), \max\{c_1, c_2\}, \boldsymbol{x}, \boldsymbol{y} \right) \end{aligned}$$

Let $Y : T_1^2 \times \Delta_{\mathcal{S}_1} \rightrightarrows \Delta_{\mathcal{S}_2}$ *be a response function such that that*

$$\text{(4.50)} \qquad \forall u \in T_1^2, \forall \boldsymbol{x} \in \Delta_{\mathcal{S}_1} : \emptyset \neq Y(u, \boldsymbol{x}) \subseteq B_u(\boldsymbol{x})$$

Then Y *satisfies* (4.46), *and there exists* $\boldsymbol{x}^*$ *satisfying* (4.47).

Proof. Substituting (4.49) and (4.50) into (4.47) yields:

$$g_1(\boldsymbol{x}) = \inf_{u \in T_1^2} \gamma_u(\boldsymbol{x})$$

$$\text{where } \gamma_u(\boldsymbol{x}) \coloneqq \max_{\boldsymbol{y} \in A_u(\boldsymbol{x})} \mathcal{P}^- \left(\widehat{\mathcal{P}}\left(\begin{smallmatrix} u_1 & c_1 \\ u & c_2 \end{smallmatrix}\right), \max\{c_1, c_2\}, \boldsymbol{x}, \boldsymbol{y} \right)$$

By Berge Maximum Theorem [AB13, Theorem 17.31], since $\mathcal{P}^-(u, c, \boldsymbol{x}, \boldsymbol{y})$ is continuous and concave in both $\boldsymbol{x}$ and $\boldsymbol{y}$ for any fixed u and c (Theorem 2.8), it follows that:

- $A_u(\boldsymbol{x})$ is nonempty, convex and compact for all $\boldsymbol{x} \in \Delta_{\mathcal{S}_1}$.
- $A_u(\cdot)$ is an upper hemicontinuous correspondence.
- $\gamma_u(\cdot)$ is upper semicontinuous.

Then $g_1(\cdot)$ is upper semicontinuous [AB13, Lemma 2.41]. Therefore, $g_1(\cdot)$ attains its maximum at some $\boldsymbol{x}^*$ in the compact set $\Delta_{\mathcal{S}_1}$ [AB13, Theorem 2.43]. □

In the Nash and Stackelberg equilibrium definitions, the first player considers each $u_2' \in \mathrm{N_C}(u_1, c_1, c_2)$ and then computes the payoff bounds corresponding to u_2'. Since $\mathcal{P}^-(\cdot)$ enumerates all true payoff functions, an alternative idea is to consider true payoff functions $u' \in \alpha^{-1}(u_1, c_1)$ and use the value $u'(\cdot, \cdot)$ directly, which may be easier to work with in some cases. The following lemma shows their equivalence.

Lemma 4.10
In a two-player fully-capability-aware game instance, let $Y : U \times \Delta_{\mathcal{S}_1} \mapsto 2^{\Delta_{\mathcal{S}_2}}$ be a nonempty-valued set-valued function. Let $c_h \coloneqq \max\{c_1, c_2\}$ and $T_1^2 \coloneqq \mathrm{N_C}(u_1, c_1, c_2)$. For any $\boldsymbol{x} \in \Delta_{\mathcal{S}_1}$, it holds that

$$\inf_{u' \in T_1^2} \inf_{\boldsymbol{y} \in Y(u', \boldsymbol{x})} \mathcal{P}^-\left(\widehat{\mathcal{P}}\left(\begin{smallmatrix} u_1 & c_1 \\ u' & c_2 \end{smallmatrix}\right), c_h, \boldsymbol{x}, \boldsymbol{y}\right) = \inf_{u' \in \alpha^{-1}(u_1, c_1)} \inf_{\boldsymbol{y} \in Y(\alpha(u', c_2), \boldsymbol{x})} u'(\boldsymbol{x}, \boldsymbol{y}) \tag{4.51}$$

Proof. Recall that $\mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{y}) = \inf_{u' \in \alpha^{-1}(u_1, c_1)} u'(\boldsymbol{x}, \boldsymbol{y})$ (Definition 2.7).

If $c_1 \geq c_2$, then $c_h = c_1$, and $T_1^2 = \{\alpha(u_1, c_2)\}$. Both sides of (4.51) are equal to $\inf_{\boldsymbol{y} \in Y(\alpha(u_1, c_2), \boldsymbol{x})} \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{y})$.

If $c_1 < c_2$, then $T_1^2 = \{u' \in U \mid \bar{c}(u') \leq c_2 \text{ and } \alpha(u', c_1) = u_1\}$. Define a set $S \coloneqq \{u'' \in U \mid \exists u' \in T_1^2 : u'' \in \alpha^{-1}(u', c_2)\}$. We have

$$\begin{aligned}
\inf_{u' \in T_1^2} \inf_{\boldsymbol{y} \in Y(u', \boldsymbol{x})} \mathcal{P}^-(u', c_2, \boldsymbol{x}, \boldsymbol{y}) &= \inf_{u' \in T_1^2} \inf_{\boldsymbol{y} \in Y(u', \boldsymbol{x})} \inf_{u'' \in \alpha^{-1}(u', c_2)} u''(\boldsymbol{x}, \boldsymbol{y}) \\
&= \inf_{u' \in T_1^2} \inf_{u'' \in \alpha^{-1}(u', c_2)} \inf_{\boldsymbol{y} \in Y(\alpha(u'', c_2), \boldsymbol{x})} u''(\boldsymbol{x}, \boldsymbol{y}) = \inf_{u'' \in S} \inf_{\boldsymbol{y} \in Y(\alpha(u'', c_2), \boldsymbol{x})} u''(\boldsymbol{x}, \boldsymbol{y})
\end{aligned}$$

To prove (4.51), it suffices to show that $S = \alpha^{-1}(u_1, c_1)$. By Definition 2.2, $\alpha^{-1}(u_1, c_1) = \{u \in U \mid \alpha(u, c_1) = u_1\}$. For $u' \in \alpha^{-1}(u_1, c_1)$, if $\bar{c}(u') \leq c_2$, then $u' \in T_1^2$ and $u' \in \alpha^{-1}(u', c_2)$; otherwise, let $u'' \coloneqq \alpha(u', c_2)$; then $u'' \in T_1^2$ and $u' \in \alpha^{-1}(u'', c_2)$. Thus $u' \in S$ in both cases. Conversely, for $u' \in S$, there exists u'' such that $u_1 = \alpha(u'', c_1)$ and $u'' = \alpha(u', c_2)$, which implies $u_1 = \alpha(u', c_1)$ and thus $u' \in \alpha^{-1}(u_1, c_1)$. Therefore, $S = \alpha^{-1}(u_1, c_1)$. □

4.3.2 Hardness of equilibrium solving

The minimax theorem implies that in a conventional zero-sum game, the value of all Nash equilibria equals the value of the mixed Stackelberg equilibrium. Consequently, a Nash equilibrium can be found by solving the leader's Stackelberg strategies for both players via linear programming [Nis+07, Section 1.4.2].

By contrast, for two-player fully-capability-aware games, the zero-sum case is still hard. The following theorem presents a reduction from a general-sum instance to a TFZOLP-Game instance. The key idea is that in a TFZOLP-Game instance, players are optimizing weakly coupled objectives (one maximizing the lower bound, the other minimizing the upper bound of a different perception). Since any game's equilibrium is invariant with respect to affine transformations of the payoff functions, given a two-player general-sum instance, we can separate the ranges of the players' payoffs by applying affine transformation so that they are optimizing equivalent objectives in a corresponding TFZOLP-Game instance.

Theorem 4.11 (Reducing a general-sum game to TFZOLP-Game)
Let (u_1, v_1) and (u_2, v_2) be the perceived payoff functions of the players in a two-player OLP-Game instance with capability levels $c_1 \le c_2$ and $\alpha(\cdot)$ as the payoff perception function. Let $m \coloneqq |\mathcal{S}_1|$ and $n \coloneqq |\mathcal{S}_2|$ be the numbers of pure actions available to each player. Let $T_1^2 \coloneqq \mathrm{N_C}(v_1, c_1, c_2)$.

Then there exist constants $k_n, b_n \in \mathbb{R}$ with $k_n \neq 0$, and a TFZOLP-Game instance where $\bar{\mathcal{S}}_1 = [\![1, 2m]\!]$, $\bar{\mathcal{S}}_2 = \mathcal{S}_2$, $\bar{\alpha}(\cdot)$ is the payoff perception function, and players have capability levels $\bar{c}_1 = 1$ and $\bar{c}_2 = 2$ with perceptions as $(\bar{u}_1, -\bar{u}_1)$ and $(\bar{u}_2, -\bar{u}_2)$, such that for any Nash equilibrium $\left(\overline{\boldsymbol{x}}^*, \overline{\boldsymbol{Y}}\right)$ of the TFZOLP-Game instance, $(\boldsymbol{x}^*, \boldsymbol{Y})$ is a Nash equilibrium of the original game where

$$\boldsymbol{x}^* \coloneqq \overline{\boldsymbol{x}}^*_{:m}, \quad \boldsymbol{Y}(v) \coloneqq \overline{\boldsymbol{Y}}\left(\begin{bmatrix}\mathbf{1}\\ k_n \boldsymbol{v} + b_n\end{bmatrix}\right) \text{ for } v \in T_1^2 \tag{4.52}$$

Note that we treat a payoff function and a payoff matrix interchangeably, with the first player being the row player. For a (row) vector or a matrix $\boldsymbol{A}$, $\boldsymbol{A}_{:m}$ denotes the first m rows of $\boldsymbol{A}$, and $\boldsymbol{A}_{m+1:}$ denotes the remaining rows.

Proof. Let $U \coloneqq \mathbb{R}^{m\times n}$ and $\bar{U} \coloneqq \mathbb{R}^{2m\times n}$ be the sets of all possible payoff functions in the general-sum and zero-sum games, respectively. For a set $C \subsetneq U$ of payoff matrices, use $\min C$ to denote the minimum payoff among the items in C, defined as $\min C \coloneqq \inf_{\boldsymbol{A}\in C} \min_{i,j} A_{ij}$. Define $\max C$ analogously. For $k, b \in \mathbb{R}$, use $kC + b$ to denote the set $\{k\boldsymbol{A} + b \mid \boldsymbol{A} \in S\}$.

Let $C_u \coloneqq \alpha^{-1}(u_1, c_1)$ and $C_v \coloneqq \alpha^{-1}(v_1, c_1)$. Note that $T_1^2 \subseteq C_v$. Due to the bounded concretization property, there are constants $k_n < 0 < k_p$ (where "n" stands for negative and "p" for positive) and $b_n, b_p \in \mathbb{R}$ such that

$$0 < \min(k_p C_u + b_p) \le \max(k_p C_u + b_p) < 1$$
$$1 < \min(k_n C_v + b_n) \le \max(k_n C_v + b_n) < 2$$

Define the function $\bar{u}_1(\cdot)$ as the all-one function, i.e., $\bar{u}_1(i, j) = 1$ for $i \in \bar{\mathcal{S}}_1$ and $j \in \bar{\mathcal{S}}_2$. Define a function $\bar{\alpha}_0 : \bar{U} \times \overline{\mathbb{Z}}_+ \to \bar{U}$ as $\bar{\alpha}_0(\bar{u}_1, c) = \bar{u}_1$, and $\bar{\alpha}_0(u, c) = u$ except for the following cases:

$$\forall \boldsymbol{V} \in T_1^2: \quad \bar{\alpha}_0\left(\begin{bmatrix}k_n \boldsymbol{V}' + b_n\\ \mathbf{0}\end{bmatrix}, 2\right) = \begin{bmatrix}\mathbf{1}\\ k_n \boldsymbol{V} + b_n\end{bmatrix} \quad \text{for } \boldsymbol{V}' \in \alpha^{-1}(\boldsymbol{V}, c_2)$$
$$\forall \boldsymbol{V} \in T_1^2: \bar{\alpha}_0\left(\begin{bmatrix}k_p \boldsymbol{U} + b_p\\ -(k_n \boldsymbol{V} + b_n)\end{bmatrix}, 2\right) = \begin{bmatrix}\mathbf{1}\\ k_n \boldsymbol{V} + b_n\end{bmatrix} \quad \text{for } \boldsymbol{U} \in C_u$$
$$\forall \boldsymbol{V} \in T_1^2: \quad \bar{\alpha}_0\left(\begin{bmatrix}\mathbf{1}\\ k_n \boldsymbol{V} + b_n\end{bmatrix}, 2\right) = \begin{bmatrix}\mathbf{1}\\ k_n \boldsymbol{V} + b_n\end{bmatrix}, \ \bar{\alpha}_0\left(\begin{bmatrix}\mathbf{1}\\ k_n \boldsymbol{V} + b_n\end{bmatrix}, 1\right) = \bar{u}_1$$

Define $\bar{\alpha} : \bar{U} \times \overline{\mathbb{Z}}_+ \to \bar{U}$ as $\bar{\alpha}(-u, c) \coloneqq -\bar{\alpha}_0(u, c)$ if u corresponds to any special case above; otherwise, $\bar{\alpha}(u, c) \coloneqq \bar{\alpha}_0(u, c)$. Then $\bar{\alpha}(\cdot)$ is an odd payoff perception function. Let $\bar{T}_1^2 \coloneqq \mathrm{N}_{\mathrm{C}}(\bar{u}_1, 1, 2)$ be the narrow concretization set with respect to $\bar{\alpha}(\cdot)$. We have $\bar{T}_1^2 = \{\bar{u}_1\} \cup \left\{ \begin{bmatrix} \mathbf{1} \\ k_n \boldsymbol{V} + b_n \end{bmatrix} \middle| \, \boldsymbol{V} \in T_1^2 \right\}$. One can verify that for any $u' \in \bar{T}_1^2 \setminus \{\bar{u}_1\}$, $\boldsymbol{x} \in \Delta_{2m}$, and $\boldsymbol{y} \in \Delta_n$, the payoff bounds satisfy:

$$\mathcal{P}_{\bar{\alpha}}^-(u', 2, \boldsymbol{x}, \boldsymbol{y}) = \inf_{\boldsymbol{U} \in C_u} \boldsymbol{x}^\intercal \begin{bmatrix} k_p \boldsymbol{U} + b_p \\ -(k_n \boldsymbol{V}_0 + b_n) \end{bmatrix} \boldsymbol{y} \overset{?}{=} k_p \mathcal{P}_\alpha^-(u_1, c_1, \boldsymbol{x}_{:m}, \boldsymbol{y}) + b_p$$

$$\mathcal{P}_{\bar{\alpha}}^+(u', 2, \boldsymbol{x}, \boldsymbol{y}) \overset{?}{=} \sup_{\boldsymbol{V}' \in \alpha^{-1}(\boldsymbol{V}_0, c_2)} \boldsymbol{x}^\intercal \begin{bmatrix} k_n \boldsymbol{V}' + b_n \\ \mathbf{0} \end{bmatrix} \boldsymbol{y} = k_n \mathcal{P}_\alpha^-(\boldsymbol{V}_0, c_2, \boldsymbol{x}_{:m}, \boldsymbol{y}) + b_n$$

$$\text{where } u' = \begin{bmatrix} \mathbf{1} \\ k_n \boldsymbol{V}_0 + b_n \end{bmatrix} \text{ for some } \boldsymbol{V}_0 \in T_1^2, \text{ and } \overset{?}{=} \text{ holds if } \boldsymbol{x}_{m+1:} = 0$$

We also have $\mathcal{P}_{\bar{\alpha}}^-(\bar{u}_1, 2, \boldsymbol{x}, \boldsymbol{y}) = \mathcal{P}_{\bar{\alpha}}^+(\bar{u}_1, 2, \boldsymbol{x}, \boldsymbol{y}) = 1$ for any $\boldsymbol{x} \in \Delta_{2m}$ and $\boldsymbol{y} \in \Delta_n$. Since $-(k_n \boldsymbol{V}_0 + b_n) < 0 < k_p \boldsymbol{U} + b_p < 1$ for $\boldsymbol{V}_0 \in T_1^2$ and $\boldsymbol{U} \in C_u$, and $\overline{\boldsymbol{x}}^*$ maximizes $\boldsymbol{x} \mapsto \inf_{u' \in \bar{T}_1^2} \mathcal{P}_{\bar{\alpha}}^-\big(u', 2, \boldsymbol{x}, \overline{\boldsymbol{Y}}(u')\big)$, we have $\overline{\boldsymbol{x}}^*_{m+1:} = \mathbf{0}$. Given (4.42) and (4.43), the equilibrium condition for $\big(\overline{\boldsymbol{x}}^*, \overline{\boldsymbol{Y}}\big)$ is:

$$\overline{\boldsymbol{x}}^* \in \operatorname*{argmax}_{\boldsymbol{x} \in \Delta_{2m}} \inf_{u' \in \bar{T}_1^2} \mathcal{P}_{\bar{\alpha}}^-\big(u', 2, \boldsymbol{x}, \overline{\boldsymbol{Y}}(u')\big) = \operatorname*{argmax}_{\substack{\boldsymbol{x} \in \Delta_{2m} \\ \boldsymbol{x}_{m+1:} = \mathbf{0}}} \inf_{\substack{u' \in \bar{T}_1^2 \\ u' \neq \bar{u}_1}} \mathcal{P}_\alpha^-\big(u_1, c_1, \boldsymbol{x}_{:m}, \overline{\boldsymbol{Y}}(u')\big)$$

$$\forall u' \in \bar{T}_1^2 \setminus \{\bar{u}_1\} : \overline{\boldsymbol{Y}}(u') \in \operatorname*{argmin}_{\boldsymbol{y} \in \Delta_n} \mathcal{P}_{\bar{\alpha}}^+\big(u', 2, \overline{\boldsymbol{x}}^*, \boldsymbol{y}\big) = \operatorname*{argmax}_{\boldsymbol{y} \in \Delta_n} \mathcal{P}_\alpha^-(\boldsymbol{V}_{u'}, c_2, \overline{\boldsymbol{x}}^*_{:m}, \boldsymbol{y})$$

$$\text{where } \boldsymbol{V}_{u'} \coloneqq \frac{\boldsymbol{u}'_{m+1:} - b_n}{k_n} \text{ so that } \begin{bmatrix} \mathbf{1} \\ k_n \boldsymbol{V}_{u'} + b_n \end{bmatrix} = u' \text{ always holds}$$

Define $(\boldsymbol{x}^*, \boldsymbol{Y})$ as in (4.52). Then the above conditions are equivalent to (4.7) and (4.8) for $(\boldsymbol{x}^*, \boldsymbol{Y})$ in the original game. □

4.3.3 Efficiently solvable games

This section identifies classes of two-player fully-capability-aware zero-sum games with properties similar to the minimax theorem, which allows efficient computation of approximate Nash equilibria. In particular, I show that if one player's maximin value is equal to their Stackelberg value in an instance (called a *maximin-attainable* instance), then the instance is efficiently solvable. In the following subsections, I first define (uniformly) maximin-attainable instances and present properties of equilibria in such instances. Then I present three classes of uniformly maximin-attainable instances: constant-gap games, strategy-separable linear component games, and narrowly reversible games.

4.3.3.1 The maximin-attainable TFZolp-Game instances

Definition 4.17 (Maximin strategies)
In an instance of a two-player fully-capability-aware zero-sum game as defined in Definition 4.12, define the *maximin strategy* of the first player as $\boldsymbol{x}^* \in \Delta_{\mathcal{S}_1}$ that satisfies

$$\begin{aligned} h_1(\boldsymbol{x}^*) &= \sup_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} h_1(\boldsymbol{x}) \\ \text{where } h_1(\boldsymbol{x}) &\coloneqq \inf_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{y}) \end{aligned} \tag{4.53}$$

The function $h_1(\cdot)$ is called the *maximin objective function* of the first player; the value $h_1(\boldsymbol{x}^*)$ is called the *maximin value* of the first player. ◁

Theorem 4.12 (Existence of the maximin strategy)
In an instance of a two-player fully-capability-aware zero-sum game, the maximin strategy as defined in Definition 4.17 always exists.

Proof. Since $\mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{y})$ is continuous and bounded in both $\boldsymbol{x}$ and $\boldsymbol{y}$ (Theorem 2.8), $h_1(\cdot)$ defined in (4.53) is continuous and bounded. Since $\Delta_{\mathcal{S}_1}$ is compact, the supremum of $h_1(\cdot)$ is achieved at some $\boldsymbol{x}^* \in \Delta_{\mathcal{S}_1}$. □

Lemma 4.13 (Maximin objective function bounded by Stackelberg objective function)
In an instance of a two-player fully-capability-aware zero-sum game, it holds that

$$\forall \boldsymbol{x} \in \Delta_{\mathcal{S}_1} : \; h_1(\boldsymbol{x}) \leq g_1(\boldsymbol{x}), \tag{4.54}$$

where $h_1(\cdot)$ is the maximin objective function defined in (4.53), and $g_1(\cdot)$ is the Stackelberg objective function when the first player is the leader as defined in (4.47). The response function $Y(\cdot)$ associated with $g_1(\cdot)$ is an arbitrary function that satisfies (4.46).

Proof. Plugging Lemma 4.10 into the definition of $g_1(\cdot)$ yields

$$g_1(\boldsymbol{x}) = \inf_{u' \in \alpha^{-1}(u_1, c_1)} \inf_{\boldsymbol{y} \in Y(\alpha(u', c_2), \boldsymbol{x})} u'(\boldsymbol{x}, \boldsymbol{y})$$

Plugging Definition 2.7 into the definition of $h_1(\cdot)$ yields

$$h_1(\boldsymbol{x}) = \inf_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \inf_{u' \in \alpha^{-1}(u_1, c_1)} u'(\boldsymbol{x}, \boldsymbol{y}) = \inf_{u' \in \alpha^{-1}(u_1, c_1)} \inf_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} u'(\boldsymbol{x}, \boldsymbol{y})$$

Since $Y(u, \boldsymbol{x}) \subseteq \Delta_{\mathcal{S}_2}$ for all $u \in U$ and $\boldsymbol{x} \in \Delta_{\mathcal{S}_1}$, it follows that $h_1(\boldsymbol{x}) \leq g_1(\boldsymbol{x})$. □

Lemma 4.14 (Order of TFZolp-Game equilibrium values)
In an instance of a two-player fully-capability-aware zero-sum game where $c_1 \leq c_2$, let $\boldsymbol{x}^*_m$ be the maximin strategy of the first player and $V_h \coloneqq h_1(\boldsymbol{x}^*_m)$ be the corresponding maximin value (Definition 4.17). Let $(\boldsymbol{x}^*_n, \boldsymbol{Y}_n)$ be a Nash equilibrium, and V_n be the corresponding Nash value of the first player (Definition 4.13).

Given $\boldsymbol{Y}_n(\cdot)$*, define an extended response function* $\boldsymbol{Y}^* : T_1^2 \times \Delta_{\mathcal{S}_1} \mapsto \Delta_{\mathcal{S}_2}$ *where* $T_1^2 \coloneqq \mathrm{N_C}(u_1, c_1, c_2)$ *as*

$$\boldsymbol{Y}^*(u, \boldsymbol{x}) \coloneqq \begin{cases} \boldsymbol{Y}_n(u) & \text{if } \boldsymbol{x} = \boldsymbol{x}_n^* \\ \operatorname{argmin}^* \limits_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^+(u, c_2, \boldsymbol{x}, \boldsymbol{y}) & \text{otherwise} \end{cases}$$

Define a set-valued response function as $Y(u, \boldsymbol{x}) = \{\boldsymbol{Y}^*(u, \boldsymbol{x})\}$*. Let* $V_s \coloneqq \sup_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} g_1(\boldsymbol{x})$ *be the Stackelberg value (Definition 4.16) with the first player being the leader and the second player's response function being* $Y(\cdot)$*. Of note,* $Y(\cdot)$ *satisfies the best response condition of* (4.46) *since* $\boldsymbol{Y}_n(\cdot)$ *satisfies* (4.42)*.*

Then the following inequality holds:

$$V_h \leq V_n \leq V_s \tag{4.55}$$

If $V_h = V_s$*, then* $(\boldsymbol{x}_n^*, \boldsymbol{Y}_n(u_2))$ *is a Stackelberg equilibrium.*

Proof. It's easy to verify that the Nash value of the first player defined in Definition 4.13 can be defined using the extended response function $\boldsymbol{Y}^*(\cdot)$ as:

$$V_n = \sup_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} f_1(\boldsymbol{x}, \boldsymbol{x}_n^*) = f_1(\boldsymbol{x}_n^*, \boldsymbol{x}_n^*)$$
$$\text{where } f_1(\boldsymbol{x}, \boldsymbol{x}') \coloneqq \inf_{u' \in T_1^2} \mathcal{P}^-(u', c_2, \boldsymbol{x}, \boldsymbol{Y}^*(u', \boldsymbol{x}')) = \inf_{u' \in T_1^2} \inf_{\boldsymbol{y} \in Y(u', \boldsymbol{x}')} \mathcal{P}^-(u', c_2, \boldsymbol{x}, \boldsymbol{y})$$

Plugging Definition 2.7 and Lemma 4.10 into $h_1(\cdot)$ yields

$$h_1(\boldsymbol{x}) = \inf_{u' \in T_1^2} \inf_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^-(u', c_2, \boldsymbol{x}, \boldsymbol{y}) \leq f_1(\boldsymbol{x}, \boldsymbol{x}_n^*)$$

Combining the above with $V_n = \sup_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} f_1(\boldsymbol{x}, \boldsymbol{x}_n^*)$ and $V_h = \sup_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} h_1(\boldsymbol{x})$ implies $V_h \leq V_n$.

Expanding $g_1(\cdot)$ in the definition of V_s yields

$$V_s = \sup_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} \inf_{u' \in T_1^2} \inf_{\boldsymbol{y} \in Y(u', \boldsymbol{x})} \mathcal{P}^-(u', c_2, \boldsymbol{x}, \boldsymbol{y}) = \sup_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} f_1(\boldsymbol{x}, \boldsymbol{x}) \geq f_1(\boldsymbol{x}_n^*, \boldsymbol{x}_n^*) = V_n$$

Therefore, (4.55) holds. The above inequality also implies that $(\boldsymbol{x}_n^*, \boldsymbol{Y}_n(u_2))$ is a Stackelberg equilibrium when $V_h = V_s$. □

Given Lemma 4.14, it is natural to consider games with $V_h = V_s$, so that the lower-capability player's Nash value equals their maximin value. The following definition captures this concept.

Definition 4.18 (Maximin-attainable TFZOLP-Game instances)
Given an instance of a two-player fully-capability-aware zero-sum game as defined in Definition 4.12, assume without loss of generality that $c_1 \leq c_2$. This instance is called *maximin-attainable* for the first player if for any singleton-valued second player's response function, the first player's Stackelberg value is the same as their maximin value. The instance is

uniformly maximin-attainable if the maximin objective function agrees with the Stackelberg objective function for any first player's strategy and any singleton-valued second player's response function.

Formally, let $T_1^2 \coloneqq \mathrm{N_C}\,(u_1,\, c_1,\, c_2)$. Given any best response function $\boldsymbol{Y} : T_1^2 \times \Delta_{\mathcal{S}_1} \mapsto \Delta_{\mathcal{S}_2}$ such that

$$\forall (u,\, \boldsymbol{x}) \in T_1^2 \times \Delta_{\mathcal{S}_1} :\ \boldsymbol{Y}(u,\, \boldsymbol{x}) \in \operatorname*{argmin}_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^+(u,\, c_2,\, \boldsymbol{x},\, \boldsymbol{y}), \tag{4.56}$$

define $g_{\boldsymbol{Y}} : \Delta_{\mathcal{S}_1} \mapsto \mathbb{R}$ as

$$g_{\boldsymbol{Y}}(\boldsymbol{x}) \coloneqq \inf_{u_2' \in T_1^2} \mathcal{P}^-(u_2',\, c_2,\, \boldsymbol{x},\, \boldsymbol{Y}(u_2',\, \boldsymbol{x})) \tag{4.57}$$

The function $g_{\boldsymbol{Y}}(\cdot)$ corresponds to the Stackelberg objective function $g_{1\cdot}$ as defined in Definition 4.16 with the follower's response function being $Y(u,\, \boldsymbol{x}) \coloneqq \{\boldsymbol{Y}(u,\, \boldsymbol{x})\}$. The function $g_{\boldsymbol{Y}}(\cdot)$ is called a *singleton-valued Stackelberg objective function*.

Let $\boldsymbol{x}^*$ and $h_1(\cdot)$ be the maximin strategy and the maximin objective function of the first player as defined in Definition 4.17. Then the instance is *maximin-attainable* if and only if $h_1(\boldsymbol{x}^*) = \sup_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} g_{\boldsymbol{Y}}(\boldsymbol{x})$ for any $\boldsymbol{Y}(\cdot)$ satisfying (4.56). Furthermore, the instance is *uniformly maximin-attainable* if and only if $h_1(\boldsymbol{x}) = g_{\boldsymbol{Y}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \Delta_{\mathcal{S}_1}$ and any $\boldsymbol{Y}(\cdot)$ satisfying (4.56). ◁

4.3.3.2 Equilibrium of uniformly maximin-attainable TFZolp-Game instances

This section argues that the lower-capability player in a uniformly maximin-attainable instance should use their maximin strategy as their Nash/Stackelberg strategy. I present a few theorems to support this argument.

First, I show that for maximin-attainable instances, the maximin strategy is likely tractable and the maximin value is the same as the Nash and Stackelberg values:

Theorem 4.15 (Equilibrium of maximin-attainable TFZolp-Game instances) *In an instance of a two-player fully-capability-aware zero-sum game as defined in Definition 4.12, assume that the instance is a maximin-attainable TFZolp-Game for the first player as defined in Definition 4.18. Then there exists a strategy profile $(\boldsymbol{x}^*,\, \boldsymbol{y}^*)$ that satisfies:*

$$\begin{gathered} \mathfrak{f}_1(\boldsymbol{x}^*) = \sup_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} \mathfrak{f}_1(\boldsymbol{x}), \quad \mathfrak{f}_2(\boldsymbol{y}^*) = \inf_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathfrak{f}_2(\boldsymbol{y}) \\ \textit{where } \mathfrak{f}_1(\boldsymbol{x}) \coloneqq \min_{1 \le k \le |\mathcal{S}_2|} \mathcal{P}^-(u_1,\, c_1,\, \boldsymbol{x},\, \boldsymbol{e}_k), \quad \mathfrak{f}_2(\boldsymbol{y}) \coloneqq \mathcal{P}^+(u_2,\, c_2,\, \boldsymbol{x}^*,\, \boldsymbol{y}) \end{gathered} \tag{4.58}$$

Any strategy profile $(\boldsymbol{x}^*,\, \boldsymbol{y}^*)$ satisfying (4.58) is a Stackelberg equilibrium when the first player is the leader. For the first player, all Nash values are the same as the Stackelberg value.

Proof. The existence of $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ is due to compactness of $\Delta_{\mathcal{S}_1}$ and $\Delta_{\mathcal{S}_2}$, and the continuity of $\mathfrak{f}_1(\cdot)$ and $\mathfrak{f}_2(\cdot)$, which follows from the continuity of payoff bounds (Theorem 2.8).

Let $h_1(\cdot)$ be the maximin objective function of the first player as defined in Definition 4.17. Since $\boldsymbol{y} \mapsto \mathcal{P}^-(u_2, c_2, \boldsymbol{x}, \boldsymbol{y})$ is concave and continuous (Theorem 2.8), and the extreme points of the simplex Δ_k are the basis vectors $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_k\}$, Lemma 4.1 implies that

$$\mathfrak{f}_1(\boldsymbol{x}) = \min_{1 \le k \le |\mathcal{S}_2|} \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{e}_k) = \inf_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{y}) = h_1(\boldsymbol{x})$$

To prove the last paragraph of the theorem, note that the definition of the maximin-attainable TFZOLP-Game requires that $h_1(\boldsymbol{x}^*) = \sup_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} g_{\boldsymbol{Y}}(\boldsymbol{x})$ for any best response function $\boldsymbol{Y}(\cdot)$. For any $\boldsymbol{x}' \in \Delta_{\mathcal{S}_1}$, we have $g_{\boldsymbol{Y}}(\boldsymbol{x}') \overset{①}{\le} h_1(\boldsymbol{x}^*) \overset{②}{\le} g_{\boldsymbol{Y}}(\boldsymbol{x}^*)$, where $\overset{①}{\le}$ is due to that $h_1(\boldsymbol{x}^*)$ is the supremum of $g_{\boldsymbol{Y}}(\cdot)$ per the maximin-attainable definition, and $\overset{②}{\le}$ is due Lemma 4.13. Therefore, $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is a Stackelberg equilibrium. Lemma 4.14 implies that $V_h = V_n = V_s$ for any Nash value V_n. □

Next, I present topological properties of the sets of equilibrium strategies of the lower-capability player in maximin-attainable instances.

Definition 4.19 (TFZOLP-Game equilibrium sets)
In a TFZOLP-Game instance that is maximin-attainable for the first player as defined in Definition 4.18, denote the sets of the first player's maximin, arbitrary Nash, some Nash, and Stackelberg strategies as $\mathcal{X}_{\mathrm{m}}$, $\mathcal{X}_{\mathrm{N}}^{\mathrm{a}}$, $\mathcal{X}_{\mathrm{N}}^{\mathrm{s}}$, and $\mathcal{X}_{\mathrm{S}}$, respectively.

Formally, let $T_1^2 \coloneqq \mathrm{N_C}\,(u_1, c_1, c_2)$. Define

- $\mathcal{X}_{\mathrm{m}} \coloneqq \operatorname{argmax}_{\boldsymbol{x} \in \mathcal{S}_1} h_1(\boldsymbol{x})$ where $h_1(\cdot)$ is the maximin objective function of the first player as defined in Definition 4.17.
- $\mathcal{X}_{\mathrm{N}}^{\mathrm{a}} \subseteq \mathcal{S}_1$ such that $\boldsymbol{x} \in \mathcal{X}_{\mathrm{N}}^{\mathrm{a}}$ if and only if for any response function $\boldsymbol{Y} : T_1^2 \mapsto \Delta_{\mathcal{S}_2}$ that is a best response to $\boldsymbol{x}$ (i.e., satisfying (4.42)), $(\boldsymbol{x}, \boldsymbol{Y})$ is a Nash equilibrium as defined in Definition 4.13.
- $\mathcal{X}_{\mathrm{N}}^{\mathrm{s}} \subseteq \mathcal{S}_1$ such that $\boldsymbol{x} \in \mathcal{X}_{\mathrm{N}}^{\mathrm{s}}$ if and only if there exists $\boldsymbol{Y}$ such that $(\boldsymbol{x}, \boldsymbol{Y})$ is a Nash equilibrium as defined in Definition 4.13.
- $\mathcal{X}_{\mathrm{S}} \coloneqq \operatorname{argmax}_{\boldsymbol{x} \in \mathcal{S}_1} g_{\boldsymbol{Y}}(\boldsymbol{x})$ where $g_{\boldsymbol{Y}}(\cdot)$ is any best response function as defined in Definition 4.18. ◁

Theorem 4.16 (Topological properties of equilibrium sets)
With the notation defined in Definition 4.19, the following properties hold in a maximin-attainable TFZOLP-Game instance:

1. $\mathcal{X}_{\mathrm{m}} \subseteq \mathcal{X}_{\mathrm{S}}$. *Moreover,* $\mathcal{X}_{\mathrm{m}}$ *is convex and compact.*
2. $\mathcal{X}_{\mathrm{N}}^{\mathrm{a}} \subseteq \mathcal{X}_{\mathrm{N}}^{\mathrm{s}} \subseteq \mathcal{X}_{\mathrm{S}}$.
3. $(\operatorname{int} \mathcal{X}_{\mathrm{m}}) \subseteq \mathcal{X}_{\mathrm{N}}^{\mathrm{a}}$.
4. $\mathcal{X}_{\mathrm{N}}^{\mathrm{s}}$ *is nonempty. If the instance is uniformly maximin-attainable, then* $\mathcal{X}_{\mathrm{m}} = \mathcal{X}_{\mathrm{S}}$, *and* $\mathcal{X}_{\mathrm{N}}^{\mathrm{s}}$ *is compact.*

Of note, the interior of $\mathcal{X}_{\mathrm{m}}$ is defined with respect to the simplex space $\Delta_{\mathcal{S}_1}$, not the Euclidean space $\mathbb{R}^{|\mathcal{S}_1|}$.

Proof. Let $V \coloneqq \max_{\boldsymbol{x}\in\mathcal{S}_1} h_1(\boldsymbol{x})$ be the first player's maximin value. Then V is also the first player's Nash value and Stackelberg value according to Theorem 4.15.

Properties 1 and 2 Theorem 4.15 directly implies $\mathcal{X}_{\mathrm{m}} \subseteq \mathcal{X}_{\mathrm{S}}$. Since $h_1(\cdot)$ is concave and continuous, $\mathcal{X}_{\mathrm{m}}$ is convex and compact. We have $\mathcal{X}_{\mathrm{N}}^{\mathrm{a}} \subseteq \mathcal{X}_{\mathrm{N}}^{\mathrm{s}}$ by definition. For any $\boldsymbol{x} \in \mathcal{X}_{\mathrm{N}}^{\mathrm{s}}$, there exits $\boldsymbol{Y}(\cdot)$ such that $(\boldsymbol{x}, \boldsymbol{Y})$ is a Nash equilibrium; since $g_{\boldsymbol{Y}}(\boldsymbol{x})$ equals the Nash value, Theorem 4.15 then implies that $g_{\boldsymbol{Y}}(\boldsymbol{x}) = V \geq g_{\boldsymbol{Y}}(\boldsymbol{x}')$ for any $\boldsymbol{x}' \in \mathcal{S}_1$, meaning that $\boldsymbol{x} \in \mathcal{X}_{\mathrm{S}}$.

Property 3 Let $\boldsymbol{Y}^{\mathrm{a}} : T_1^2 \times \Delta_{\mathcal{S}_1} \mapsto \mathcal{S}_2$ be an arbitrary best response function of the second player satisfying (4.56). Define a function $F : \Delta_{\mathcal{S}_1}^2 \mapsto \mathbb{R}$ as

$$F(\boldsymbol{x}, \boldsymbol{x}') \coloneqq \inf_{u\in T_1^2} \mathcal{P}^-(u, c_2, \boldsymbol{x}, \boldsymbol{Y}^{\mathrm{a}}(u, \boldsymbol{x}'))$$

Intuitively, $F(\boldsymbol{x}, \boldsymbol{x}')$ is the first player's Nash objective value when the first player plays $\boldsymbol{x}$ with the second player responding to $\boldsymbol{x}'$ according to $\boldsymbol{Y}^{\mathrm{a}}(\cdot)$. One can easily verify that

$$\forall \boldsymbol{x} \in \Delta_{\mathcal{S}_1} : \ F(\boldsymbol{x}, \boldsymbol{x}) = g_{\boldsymbol{Y}^{\mathrm{a}}}(\boldsymbol{x}) \tag{4.59}$$

Since $F(\boldsymbol{x}, \boldsymbol{x}')$ considers the specific response $\boldsymbol{Y}^{\mathrm{a}}(\cdot)$ while $h_1(\boldsymbol{x})$ considers the worst response, we also have

$$\forall \boldsymbol{x}, \boldsymbol{x}' \in \Delta_{\mathcal{S}_1} : \ F(\boldsymbol{x}, \boldsymbol{x}') \geq h_1(\boldsymbol{x}) \tag{4.60}$$

Take an arbitrary $\boldsymbol{x}_1 \in \operatorname{int} \mathcal{X}_{\mathrm{m}}$. Then $\boldsymbol{x}_1 \in \mathcal{X}_{\mathrm{N}}^{\mathrm{a}}$ if and only if $F(\boldsymbol{x}_1, \boldsymbol{x}_1) \geq F(\boldsymbol{x}, \boldsymbol{x}_1)$ for all $\boldsymbol{x} \in \Delta_{\mathcal{S}_1}$. Take an arbitrary $\bar{\boldsymbol{x}} \in \Delta_{\mathcal{S}_1} \setminus \{\boldsymbol{x}_1\}$. Let $\boldsymbol{d} \coloneqq \bar{\boldsymbol{x}} - \boldsymbol{x}_1$. Since $\boldsymbol{x}_1 \in \operatorname{int} \mathcal{X}_{\mathrm{m}}$, there exists $\epsilon > 0$ such that $\boldsymbol{x}_1 - \epsilon\boldsymbol{d} \in \mathcal{X}_{\mathrm{m}}$. Let $\boldsymbol{x}_2 \coloneqq \boldsymbol{x}_1 - \epsilon\boldsymbol{d}$. Eq. (4.60) implies that $F(\boldsymbol{x}_2, \boldsymbol{x}_1) \geq h_1(\boldsymbol{x}_2) = V$ since $\boldsymbol{x}_2 \in \mathcal{X}_{\mathrm{m}}$. Eq. (4.59) implies that $F(\boldsymbol{x}_1, \boldsymbol{x}_1) = g_{\boldsymbol{Y}^{\mathrm{a}}}(\boldsymbol{x}_1) = V$ since $\boldsymbol{x}_1 \in \mathcal{X}_{\mathrm{m}} \subseteq \mathcal{X}_{\mathrm{S}}$. One can also verify that $F(\boldsymbol{x}, \boldsymbol{x}')$ is continuous and concave in $\boldsymbol{x}$ for any fixed $\boldsymbol{x}'$. Since $\boldsymbol{x}_2 = \boldsymbol{x}_1 - \epsilon(\bar{\boldsymbol{x}} - \boldsymbol{x}_1)$, we have $\boldsymbol{x}_1 = \frac{\epsilon}{1+\epsilon}\bar{\boldsymbol{x}} + \frac{1}{1+\epsilon}\boldsymbol{x}_2$, which implies that:

$$F(\bar{\boldsymbol{x}}, \boldsymbol{x}_1) \leq \left(1 + \frac{1}{\epsilon}\right) F(\boldsymbol{x}_1, \boldsymbol{x}_1) - \frac{1}{\epsilon} F(\boldsymbol{x}_2, \boldsymbol{x}_1) \leq \left(1 + \frac{1}{\epsilon}\right) V - \frac{1}{\epsilon} V = V$$

Since $\bar{\boldsymbol{x}}$ is arbitrary, it follows that $\max_{\boldsymbol{x}\in\Delta_{\mathcal{S}_1}} F(\boldsymbol{x}, \boldsymbol{x}_1) \leq V = F(\boldsymbol{x}_1, \boldsymbol{x}_1)$, which implies that $\boldsymbol{x}_1 \in \mathcal{X}_{\mathrm{N}}^{\mathrm{a}}$.

Property 4 $\mathcal{X}_{\mathrm{N}}^{\mathrm{s}}$ is nonempty due to the existence of a Nash equilibrium. If the instance is uniformly maximin-attainable, then $\mathcal{X}_{\mathrm{m}} = \mathcal{X}_{\mathrm{S}}$ follows trivially. Now we prove that $\mathcal{X}_{\mathrm{N}}^{\mathrm{s}}$ is compact.

For $u \in T_1^2$, define a correspondence $A_u : \Delta_{\mathcal{S}_1} \rightrightarrows \Delta_{\mathcal{S}_2}$ as the best response set of the second player:

$$A_u(\boldsymbol{x}) \coloneqq \operatorname*{argmax}_{\boldsymbol{y}\in\Delta_{\mathcal{S}_2}} \mathcal{P}^-(u, c_2, \boldsymbol{x}, \boldsymbol{y})$$

Since $\mathcal{P}^-(u,\, c_2,\, \boldsymbol{x},\, \boldsymbol{y})$ is continuous and concave in both $\boldsymbol{x}$ and $\boldsymbol{y}$ (Theorem 2.8), it follows that $A_u(\cdot)$ is an upper hemicontinuous correspondence (Berge Maximum Theorem [AB13, Theorem 17.31]), and that $A_u(\boldsymbol{x})$ is nonempty, convex and compact for all $\boldsymbol{x} \in \Delta_{\mathcal{S}_1}$.

Define the (possibly infinite dimensional) vector space $\overline{\mathcal{Y}} \coloneqq \mathcal{B}(T_1^2 \times \mathcal{S}_1)$ where $\mathcal{B}(V)$ denotes the set of bounded real-valued functions on V. Endow $\overline{\mathcal{Y}}$ with the topology induced by the (modified) supremum norm defined as

$$\forall p \in \overline{\mathcal{Y}} :\ \|p\| \coloneqq \sup_{u \in T_1^2} \sqrt{\sum_{x \in \mathcal{S}_1} \left(p(u,\, x)\right)^2}$$

Then $\|p\|$ for any $p \in \overline{\mathcal{Y}}$ is finite due to the boundedness.

Define the set $\overline{\mathcal{Y}}_\Delta \coloneqq \Delta_{\mathcal{S}_2}^{T_1^2}$. Interpret $\overline{\mathcal{Y}}_\Delta$ as a subset of $\overline{\mathcal{Y}}$ by identifying a point $\boldsymbol{y} \in \Delta_{\mathcal{S}_2}$ as a function in $\mathcal{B}(\mathcal{S}_2)$. Then $\overline{\mathcal{Y}}_\Delta$ is a compact and convex set in $\overline{\mathcal{Y}}$. A point $\boldsymbol{Y} \in \overline{\mathcal{Y}}_\Delta$ can be interpreted as a response function $\boldsymbol{Y} : T_1^2 \mapsto \Delta_{\mathcal{S}_2}$. Define the correspondence $A_* : \Delta_{\mathcal{S}_1} \rightrightarrows \overline{\mathcal{Y}}_\Delta$ as

$$A_*(\boldsymbol{x}) \coloneqq \prod_{u \in T_1^2} A_u(\boldsymbol{x})$$

Then $A_*(\cdot)$ is an upper hemicontinuous correspondence, and $A_*(\boldsymbol{x})$ is nonempty, convex and compact for all $\boldsymbol{x} \in \Delta_{\mathcal{S}_1}$.

Due to the bounded concretization property (Definition 2.2), there is a constant $L > 0$ such that for any $u \in \alpha^{-1}(u_1,\, c_1)$, the function $(\boldsymbol{x},\, \boldsymbol{y}) \mapsto u(\boldsymbol{x},\, \boldsymbol{y})$ is L-Lipschitz; since the Lipschitz constant can be carried over point-wise infimum or supremum [Wea18, Proposition 1.32], it follows that:

$$\begin{gathered}\forall u \in T_1^2 :\ \forall \boldsymbol{x}, \boldsymbol{x}' \in \Delta_{\mathcal{S}_1} :\ \forall \boldsymbol{y}, \boldsymbol{y}' \in \Delta_{\mathcal{S}_2} : \\ \left|\mathcal{P}^-(u,\, c_2,\, \boldsymbol{x},\, \boldsymbol{y}) - \mathcal{P}^-(u,\, c_2,\, \boldsymbol{x}',\, \boldsymbol{y}')\right| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\|_2 + L\|\boldsymbol{y} - \boldsymbol{y}'\|_2\end{gathered}$$

Define the Nash objective function $N : \Delta_{\mathcal{S}_1} \times \overline{\mathcal{Y}}_\Delta \mapsto \mathbb{R}$ as

$$N(\boldsymbol{x},\, \boldsymbol{Y}) \coloneqq \inf_{u \in T_1^2} \mathcal{P}^-(u,\, c_2,\, \boldsymbol{x},\, \boldsymbol{Y}(u))$$

If $\|\boldsymbol{Y} - \boldsymbol{Y}'\| \leq \epsilon$, then $\|\boldsymbol{Y}(u) - \boldsymbol{Y}'(u)\|_2 \leq \epsilon$ for any $u \in T_1^2$, which implies that

$$\begin{gathered}\forall \boldsymbol{x}, \boldsymbol{x}' \in \Delta_{\mathcal{S}_1} :\ \forall \boldsymbol{y}, \boldsymbol{y}' \in \Delta_{\mathcal{S}_2} : \\ |N(\boldsymbol{x},\, \boldsymbol{Y}) - N(\boldsymbol{x}',\, \boldsymbol{Y}')| \leq L(\|\boldsymbol{x} - \boldsymbol{x}'\|_2 + \|\boldsymbol{Y} - \boldsymbol{Y}'\|)\end{gathered}$$

Therefore, $N(\cdot)$ is continuous.

Define the function $G : \Delta_{\mathcal{S}_1} \mapsto \mathbb{R}$ as

$$\begin{aligned} G(\boldsymbol{x}) &\coloneqq \min_{\boldsymbol{Y} \in A_*(\boldsymbol{x})} \gamma(\boldsymbol{Y}) \\ \text{where } \gamma(\boldsymbol{Y}) &\coloneqq \max_{\boldsymbol{x}' \in \Delta_{\mathcal{S}_1}} N(\boldsymbol{x}',\, \boldsymbol{Y}) \end{aligned} \tag{4.61}$$

Since $\gamma(\cdot)$ takes the supremum of a family of continuous functions, $\gamma(\cdot)$ is continuous [BD84, Theorem 7.4.9]. Since $A_*(\cdot)$ is an upper hemicontinuous correspondence, $G(\cdot)$ is lower semicontinuous [AB13, Lemma 17.30].

For any $\boldsymbol{x} \in \mathcal{X}_{\mathrm{S}}$ and $\boldsymbol{Y} \in A_*(\boldsymbol{x})$, we have $G(\boldsymbol{x}) \geq N(\boldsymbol{x}, \boldsymbol{Y}) = g_{\boldsymbol{Y}}(\boldsymbol{x}) = V$. One can easily verify that if $\boldsymbol{x} \in \mathcal{X}_{\mathrm{S}}$, then $\boldsymbol{x} \in \mathcal{X}_{\mathrm{N}}^{\mathrm{s}}$ if and only if $G(\boldsymbol{x}) = V$.

Now suppose the instance is uniformly maximin-attainable. Then $\mathcal{X}_{\mathrm{N}}^{\mathrm{s}} \subseteq \mathcal{X}_{\mathrm{S}} = \mathcal{X}_{\mathrm{m}}$. Since $\mathcal{X}_{\mathrm{m}}$ is compact, $\mathcal{X}_{\mathrm{S}}$ is also compact. Let $\{\boldsymbol{x}_n\}_{n\geq 1}$ be a sequence in $\mathcal{X}_{\mathrm{N}}^{\mathrm{s}}$ that converges to $\boldsymbol{x}_\circ$. Since $\boldsymbol{x}_i \in \mathcal{X}_{\mathrm{N}}^{\mathrm{s}} \subseteq \mathcal{X}_{\mathrm{S}}$ for any $i \in \mathbb{Z}_+$, we have $\boldsymbol{x}_\circ \in \mathcal{X}_{\mathrm{S}}$. Since $G(\boldsymbol{x}_i) = V$ for any $i \in \mathbb{Z}_+$, the lower semicontinuity of $G(\cdot)$ implies that $G(\boldsymbol{x}_\circ) \leq V$. Since $\boldsymbol{x}_\circ \in \mathcal{X}_{\mathrm{S}}$, we have $G(\boldsymbol{x}_\circ) \geq V$. It follows that $G(\boldsymbol{x}_\circ) = V$, implying that $\boldsymbol{x}_\circ \in \mathcal{X}_{\mathrm{N}}^{\mathrm{s}}$. Therefore, $\mathcal{X}_{\mathrm{N}}^{\mathrm{s}}$ is compact. □

Remark 4.11 Characterizing when $\mathcal{X}_{\mathrm{m}} \subseteq \mathcal{X}_{\mathrm{N}}^{\mathrm{s}}$ is left as an interesting open problem. A sufficient condition is that given $\boldsymbol{x}^s \in \mathcal{X}_{\mathrm{m}}$, if for any open set U containing $\boldsymbol{x}^s$, there exists $\boldsymbol{x} \in U \cap \mathcal{X}_{\mathrm{m}}$ such that $A_*(\cdot)$ is continuous at $\boldsymbol{x}$, then one can prove that $\boldsymbol{x}^s \in \mathcal{X}_{\mathrm{N}}^{\mathrm{s}}$ using techniques similar to the proof of properties 3 and 4.

I conjecture that $\mathcal{X}_{\mathrm{N}}^{\mathrm{s}} = \mathcal{X}_{\mathrm{m}}$ holds for uniformly maximin-attainable instances.

Corollary 4.17 (Equilibrium play in uniformly maximin-attainable TFZolp-Game) *In a TFZolp-Game instance that is maximin-attainable for the first player as defined in Definition 4.18, with the notation defined in Definition 4.19, no player has an incentive to deviate from the following protocol:*

1. *If* $(\operatorname{int} \mathcal{X}_{\mathrm{m}}) \neq \emptyset$*, the first player chooses an arbitrary value* $\boldsymbol{x}^* \in \operatorname{int} \mathcal{X}_{\mathrm{m}}$ *deterministically. The second player solves* $\boldsymbol{y}^* \coloneqq \operatorname{argmin}^*_{\boldsymbol{y}\in\Delta_{\mathcal{S}_2}} \mathcal{P}^+(u_2, c_2, \boldsymbol{x}^*, \boldsymbol{y})$.

2. *If the instance is uniformly maximin-attainable and* $\mathcal{X}_{\mathrm{m}} = \{\boldsymbol{x}^*\}$ *contains only one element* $\boldsymbol{x}^*$*, the second player solves* $\boldsymbol{Y}^*$ *as:*

$$\begin{aligned} &\underset{\boldsymbol{Y}\in\Delta_{\mathcal{S}_2}^{T_1^2},\, v\in\mathbb{R}}{\text{minimize}} && v \\ &\text{subject to} && \boldsymbol{Y}(u) \in \operatorname*{argmin}_{\boldsymbol{y}\in\Delta_{\mathcal{S}_2}} \mathcal{P}^+(u, c_2, \boldsymbol{x}^*, \boldsymbol{y}) \quad \forall u \in T_1^2 \\ &&& \max_{\boldsymbol{x}\in\Delta_{\mathcal{S}_1}} \inf_{u\in T_1^2} \mathcal{P}^-(u, c_2, \boldsymbol{x}, \boldsymbol{Y}(u)) \leq v \end{aligned} \tag{4.62}$$

   Then the second player sets $\boldsymbol{y}^* \coloneqq \boldsymbol{Y}(u_2)$.

In both cases, the first player samples an action according to $\boldsymbol{x}^*$*, and the second player samples an action according to* $\boldsymbol{y}^*$.

Proof. In the first case, property 3 of Theorem 4.16 implies that there exists $\{\boldsymbol{x}^*, \boldsymbol{Y}\}$ as a Nash equilibrium such that $\boldsymbol{Y}(u) = \boldsymbol{y}^*$.

In the second case, property 4 of Theorem 4.16 implies that $\mathcal{X}_{\mathrm{N}}^{\mathrm{s}} = \{\boldsymbol{x}^*\}$. Then $\boldsymbol{Y}^*$ define by (4.62) is a Nash equilibrium. □

For a randomly generated or slightly disturbed TFZolp-Game instance, $\mathcal{X}_{\mathrm{m}}$ is likely (with probability 1 in many cases) to contain a single element because:

- The set of convex programs that have nonunique solutions is a negligible subset of the set of all convex programs [PW16].
- When $\mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\intercal \boldsymbol{A}\boldsymbol{y}$ is a bilinear function, the set U is open and everywhere dense with U defined as $\boldsymbol{A} \in U$ if and only if $h_1(\boldsymbol{x}) = \min_{\boldsymbol{y}} \boldsymbol{x}^\intercal \boldsymbol{A}\boldsymbol{y}$ has a unique maximizer [BKS50, Theorem 3].

Therefore, for a uniformly maximin-attainable TFZOLP-Game instance, the second case in Corollary 4.17 is likely to hold. Of note, checking the uniqueness of solutions to a linear program is in P [Man79]. However, (4.62) is a nonconvex program that is computationally challenging to solve. I argue that computation-bounded players should be satisfied with an arbitrary best strategy when the game is sufficiently hard:

Corollary 4.18
In a TFZOLP-Game instance that is maximin-attainable for the first player as defined in Definition 4.18, assume computing $\gamma(\boldsymbol{Y})$ (defined in (4.61)) for any $\boldsymbol{Y} \in A_*(\boldsymbol{x})$ and any $\boldsymbol{x} \in \Delta_{\mathcal{S}_1}$ is intractable for both players. Then both players have no incentive to deviate from the following protocol:

The first player solves an arbitrary $\boldsymbol{x}^* \in \mathcal{X}_{\mathrm{m}}$ deterministically and samples an action according to $\boldsymbol{x}^*$. The second player solves an arbitrary $\boldsymbol{y}^* = \operatorname{argmin}^*_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^+(u_2, c_2, \boldsymbol{x}^*, \boldsymbol{y})$ and samples an action according to $\boldsymbol{y}^*$.

Proof. With this protocol, player one can guarantee the same expected payoff as they would have in any Nash equilibrium due to Theorem 4.15. With the response function $\boldsymbol{Y}(u) \coloneqq \operatorname{argmin}^*_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^+(u, c_2, \boldsymbol{x}^*, \boldsymbol{y})$ for $u \in T_1^2$, since player one can not compute $\gamma(\boldsymbol{Y})$, player one is not incentivized to deviate from $\boldsymbol{x}^*$. Player two is not incentivized to deviate from $\boldsymbol{y}^*$ since it is already a best response to $\boldsymbol{x}^*$. □

An interesting case is when the first player has unlimited computational resources, while the second player has limited computational resources. Then the protocol described in Corollary 4.18 is not necessarily an equilibrium since $\gamma(\boldsymbol{Y}) > h_1(\boldsymbol{x}^*)$ may hold for the $\boldsymbol{Y}$ arbitrarily chosen by the second player. However, if the second player can hide their strategy of choosing $\boldsymbol{y}^*$ from the first player, and if $\boldsymbol{x}^* \in \mathcal{X}^{\mathrm{s}}_{\mathrm{N}}$, then the first player still has no incentive to deviate from $\boldsymbol{x}^*$.

Finally, I remark that if computing $\mathcal{P}^-(u, c, \boldsymbol{x}, \boldsymbol{y})$ and its subgradient is in polynomial time, then the objective function $\mathfrak{f}_1(\cdot)$ defined in Theorem 4.15 and its subgradient is computable in polynomial time. Therefore, an approximate maximin strategy can be efficiently computed using nonsmooth convex optimization methods, as formalized below:

Theorem 4.19 (Efficient approximations to maximin strategies)
Let $h_1(\cdot)$ be the maximin objective function of the first player in an instance of a two-player fully-capability-aware zero-sum game as defined in Definition 4.17. Let $m \coloneqq |\mathcal{S}_1|$ and $n \coloneqq |\mathcal{S}_2|$. Assume that $\mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{y})$ and an arbitrary subgradient of $\boldsymbol{x} \mapsto \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{y})$ are both computable in $\mathsf{poly}(m, n)$ time. Let L be the maximum Lipschitz constant of $\boldsymbol{x} \mapsto \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{e}_k)$ over $\Delta_{\mathcal{S}_1}$ for all $k \in [\![1, n]\!]$. Then computing an ϵ-approximate $\boldsymbol{x}_{\mathrm{approx}}$ such that $h_1(\boldsymbol{x}_{\mathrm{approx}}) \geq h_1(\boldsymbol{x}^*) - \epsilon$ can be done in $\mathsf{poly}(m, n, L, \epsilon^{-1})$ time.

Proof. Since $h_1(\boldsymbol{x}) = \min_{k \in [\![1, n]\!]} \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{e}_k)$ is concave and continuous, its subgradient (Definition 7.1) exists and can be computed from subgradients of $\boldsymbol{x} \mapsto \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{e}_k)$. The projected subgradient method computes $\boldsymbol{x}_{\text{approx}}$ in polynomial time using subgradients of $h_1(\cdot)$ [Bub+15]. □

4.3.3.3 Constant-gap games

This subsection presents a class of uniformly maximin-attainable TFZOLP-Game instances called constant games that generalize a generalization of the quantized payoff game introduced in Example 2.2. In such games, the gap between the upper and lower bounds of the payoffs is a constant depending only on the capability levels. All instances of such games are maximin-attainable.

Definition 4.20 (Constant-gap games)
A OLP-Game as defined in Definition 2.3 is a *constant-gap game* if and only if there exists a function $\delta : \mathbb{Z}_+ \to \mathbb{R}$ such that

$$\begin{gathered} \forall u \in U,\ c \in \mathbb{Z}_+,\ \boldsymbol{\sigma}_k \in \Delta_{\mathcal{S}_k} \text{ for } k \in \mathcal{N} : \\ \mathcal{P}^+\big(u, c, (\boldsymbol{\sigma}_i)_{i \in \mathcal{N}}\big) = \mathcal{P}^-\big(u, c, (\boldsymbol{\sigma}_i)_{i \in \mathcal{N}}\big) + \delta(c) \end{gathered}$$ ◁

Example 4.3 (Uniformly quantized payoff game as a constant-gap game)
The uniformly quantized payoff game is a generalization of the quantized payoff game (Example 2.2). In the uniformly quantized game, there is a constant $\gamma \in (0, 1)$ defining the quantization precision and a constant $\beta > 0$ defining the base quantization level. Define a quantization function $q : \mathbb{R} \times \overline{\mathbb{Z}}_+ \to \mathbb{R}$ as:

$$\begin{gathered} q(x, c) \coloneqq \begin{cases} \lceil \beta \gamma^{-c} x \rfloor \, \beta^{-1} \gamma^c & \text{if } c < \infty \\ x & \text{otherwise} \end{cases} \\ \text{where } \lceil x \rfloor \coloneqq \operatorname{sign}(x) \max \{ v \in \mathbb{N} \mid |v| \leq |x| \} \end{gathered}$$

For any $u \in U$ and $c \in \overline{\mathbb{Z}}_+$, define $\alpha(u, c) = u'$ such that $u'(s) = q(u(s), c)$ for all $s \in \prod_{i \in \mathcal{N}} \mathcal{S}_i$. It is easy to verify that $\alpha(\cdot, \cdot)$ is an odd payoff perception function. One can also verify that for all $u \in U$, $c \in \mathbb{Z}_+$:

$$\begin{aligned} \mathcal{P}^-\big(u, c, (\boldsymbol{\sigma}_i)_{i \in \mathcal{N}}\big) &= u_c^-(\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n) \\ \mathcal{P}^+\big(u, c, (\boldsymbol{\sigma}_i)_{i \in \mathcal{N}}\big) &= u_c^-(\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n) + \beta^{-1}\gamma^c \\ \text{where } u_c^-(s_1, \ldots, s_n) &\coloneqq \begin{cases} u(s_1, \ldots, s_n) & \text{if } u(s_1, \ldots, s_n) \geq 0 \\ u(s_1, \ldots, s_n) - \beta^{-1}\gamma^c & \text{otherwise} \end{cases} \end{aligned} \tag{4.63}$$

Therefore, the uniformly quantized payoff game is a constant-gap game with $\delta(c) = \beta^{-1}\gamma^c$. ◁

Theorem 4.20
An instance of a constant-gap TFZOLP-Game game is uniformly maximin-attainable.

Proof. Assume without loss of generality that $c_1 \leq c_2$. Let $T_1^2 \coloneqq \mathrm{N_C}(u_1, c_1, c_2)$. For any response function $Y : U \times \Delta_{\mathcal{S}_1} \to 2^{\Delta_{\mathcal{S}_2}}$ satisfying (4.46), we have

$$\begin{aligned} &\forall \boldsymbol{x} \in \Delta_{\mathcal{S}_1}, u \in T_1^2, \forall \boldsymbol{y} \in Y(u, \boldsymbol{x}) : \\ \mathcal{P}^-(u, c_2, \boldsymbol{x}, \boldsymbol{y}) &= \mathcal{P}^+(u, c_2, \boldsymbol{x}, \boldsymbol{y}) - \delta(c_2) \\ &= \min_{\boldsymbol{y}' \in \Delta_{\mathcal{S}_2}} \mathcal{P}^+(u, c_2, \boldsymbol{x}, \boldsymbol{y}') - \delta(c_2) = \min_{\boldsymbol{y}' \in \Delta_{\mathcal{S}_2}} \mathcal{P}^-(u, c_2, \boldsymbol{x}, \boldsymbol{y}') \end{aligned}$$

Consequently, for any $\boldsymbol{x} \in \Delta_{\mathcal{S}_1}$,

$$h_1(\boldsymbol{x}) = \inf_{u \in T_1^2} \min_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^-(u, c_2, \boldsymbol{x}, \boldsymbol{y}) = \inf_{u \in T_1^2} \inf_{\boldsymbol{y} \in Y(u, \boldsymbol{x})} \mathcal{P}^-(u, c_2, \boldsymbol{x}, \boldsymbol{y}) = g_1(\boldsymbol{x}) \qquad \square$$

4.3.3.4 Strategy-separable linear component games

This subsection presents linear component games, which generalize the masked payoff game introduced in Example 2.1. In such games, players have access to a set of orthogonal basis payoff functions. Given a true payoff function, it is decomposed as a linear combination of the basis functions of which only the components with the largest coefficients are kept. A player's capability level determines the number of nonzero coefficients kept in the decomposition. The idea of representing an observation with only a few linear components has been widely used in signal processing and machine learning. For example, the Principal Component Analysis (PCA) method finds the linear components that capture the most variance in the data; the PCA can facilitate the understanding of high-dimensional data by keeping only the most significant components [Bis06, Section 12.1].

Definition 4.21 (Linear component games)
A *linear component game* is an OLP-Game (Definition 2.3) with a set of basis payoff functions $\{\mathfrak{b}_i\}_{1 \leq i \leq B}$ where $\mathfrak{b}_i \in U$ and $B \coloneqq \prod_{i \in \mathcal{N}} |\mathcal{S}_i|$ is the dimensionality of the payoff space.

The basis functions are mutually orthogonal and have unit norms, i.e., for $1 \leq i < j \leq B$, $\langle \mathfrak{b}_i, \mathfrak{b}_j \rangle = 0$, and for $1 \leq i \leq B$, $\langle \mathfrak{b}_i, \mathfrak{b}_i \rangle = 1$. Of note, for any payoff functions u and v, $\langle u, v \rangle \coloneqq \sum_{s \in \prod_{i \in \mathcal{N}} \mathcal{S}_i} u(s)v(s)$ denotes their inner product.

The payoff perception function $\alpha : U \times \mathbb{Z}_+ \to U$ is defined as follows. Given $u \in U$, decompose u as a linear combination of the basis functions by computing the coefficients $\beta_i \coloneqq \langle u, \mathfrak{b}_i \rangle$ for $1 \leq i \leq B$. Let $(\pi_i)_{1 \leq i \leq B}$ be a permutation of $\{1, \ldots, B\}$ such that $|\beta_{\pi_i}| > |\beta_{\pi_{i+1}}|$ or $\left(|\beta_{\pi_i}| = |\beta_{\pi_{i+1}}| \text{ and } \pi_i < \pi_{i+1}\right)$. Then for any $c \in [\![1, B]\!]$, define $\alpha(u, c) \coloneqq \sum_{i=1}^{c} \beta_{\pi_i} \mathfrak{b}_{\pi_i}$. For any $c > B$, define $\alpha(u, c) \coloneqq u$. $\triangleleft$

One can verify that $\alpha(\cdot, \cdot)$ defined in Definition 4.21 is an odd payoff perception function. The proof is omitted here.

The masked payoff game in Example 2.1 is a special case of the linear component game where the basis functions are the standard basis vectors.

Consider the two-player case. Given a perceived payoff function $u \in U$ and a capability level $c \in \mathbb{Z}_+$, let $u = \sum_{i=1}^{c} \beta_{\pi_i} \mathfrak{b}_{\pi_i}$ be the decomposition of u as defined in Definition 4.21. By

setting $\beta_{\pi_i} = \pm\beta_{\pi_c}$ for $i > c$, the following payoff bounds are obtained:

$$\begin{aligned} \mathcal{P}^-(u,\, c,\, \boldsymbol{x},\, \boldsymbol{y}) &= u(\boldsymbol{x},\, \boldsymbol{y}) - D(u,\, c,\, \boldsymbol{x},\, \boldsymbol{y}) \\ \mathcal{P}^+(u,\, c,\, \boldsymbol{x},\, \boldsymbol{y}) &= u(\boldsymbol{x},\, \boldsymbol{y}) + D(u,\, c,\, \boldsymbol{x},\, \boldsymbol{y}) \\ \text{where } D(u,\, c,\, \boldsymbol{x},\, \boldsymbol{y}) &\coloneqq |\beta_{\pi_c}| \sum_{i=c+1}^{B} |\mathfrak{b}_{\pi_i}(\boldsymbol{x},\, \boldsymbol{y})|, \qquad u = \sum_{i=1}^{c} \beta_{\pi_i} \mathfrak{b}_{\pi_i} \end{aligned} \tag{4.64}$$

Even though the payoff bounds in (4.64) are nonlinear, the maximin strategy can still be solved via a linear program. Recall that in a linear component TFZOLP-Game instance, the maximin objective function is

$$h_1(\boldsymbol{x}) \coloneqq \inf_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^-(u_1,\, c_1,\, \boldsymbol{x},\, \boldsymbol{y}) = \min_{i \in [\![1,\, |\mathcal{S}_2|]\!]} u(\boldsymbol{x},\, \boldsymbol{e}_i) - D(u,\, c,\, \boldsymbol{x},\, \boldsymbol{e}_i) \tag{4.65}$$

Given $\boldsymbol{x}$, introduce auxiliary variables $t_{ij} = \left|\beta_{\pi_c} \mathfrak{b}_{\pi_j}(\boldsymbol{x},\, \boldsymbol{e}_i)\right|$ and $s = h_1(\boldsymbol{x})$. Then the maximin strategy can be solved by the following linear program:

$$\begin{aligned} \text{maximize} \quad & s \\ \text{subject to} \quad & s \le u_1(\boldsymbol{x},\, \boldsymbol{e}_i) - \sum_{j=c+1}^{B} t_{ij} && \forall i \in [\![1,\, |\mathcal{S}_2|]\!] \\ & t_{ij} \ge \beta_{\pi_c} \mathfrak{b}_{\pi_j}(\boldsymbol{x},\, \boldsymbol{e}_i) && \forall (i,\, j) \in [\![1,\, |\mathcal{S}_2|]\!] \times [\![c+1,\, B]\!] \\ & t_{ij} \ge -\beta_{\pi_c} \mathfrak{b}_{\pi_j}(\boldsymbol{x},\, \boldsymbol{e}_i) && \forall (i,\, j) \in [\![1,\, |\mathcal{S}_2|]\!] \times [\![c+1,\, B]\!] \\ & \boldsymbol{x} \in \Delta_{\mathcal{S}_1} \end{aligned} \tag{4.66}$$

However, solving the Stackelberg equilibrium is more challenging. The difficulty arises from the lack of a simple relationship between the values of $\boldsymbol{y} \mapsto \mathcal{P}^+(u',\, c_2,\, \boldsymbol{x},\, \boldsymbol{y})$ and the values of $\boldsymbol{y} \mapsto \mathcal{P}^-(u',\, c_2,\, \boldsymbol{x},\, \boldsymbol{y})$ when $\boldsymbol{y} \in Y(u',\, c_2)$. The problem can be formulated as a bilevel linear program where certain constraints require some variables to be the minimizers of functions involving other variables. The bilevel optimization is known as a hard problem [LPR96]. Even linear maximin problems, a special class of bilevel linear programs, are strongly NP-hard [HJS92, Theorem 3.1].

There are some instances of linear component games that are maximin-attainable and thus efficiently solvable. Let $\boldsymbol{x}^* = \operatorname{argmax}^*_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} h_1(\boldsymbol{x})$ be the maximin strategy. Let i^* be the index such that $h_1(\boldsymbol{x}^*) = u(\boldsymbol{x}^*,\, \boldsymbol{e}_{i^*}) - D(u_1,\, c_1,\, \boldsymbol{x}^*,\, \boldsymbol{e}_{i^*})$ in (4.65). If for any value of $\boldsymbol{x}^*$ and i^*, there exists $u' \in \mathrm{N_C}\,(u_1,\, c_1,\, c_2)$ such that $\mathcal{P}^+(u',\, c_2,\, \boldsymbol{x}^*,\, \boldsymbol{e}_{i^*}) = \mathcal{P}^-(u_1,\, c_1,\, \boldsymbol{x}^*,\, \boldsymbol{e}_{i^*})$, then the maximin strategy is also a Stackelberg equilibrium. The strategy-separable property defined below is a sufficient condition for the existence of such u'.

Definition 4.22 (Strategy-separable linear component TFZOLP-Game instances) In an instance of a linear component TFZOLP-Game as defined in Definitions 4.12 and 4.21, assume $c_1 < c_2$. Define $\mathfrak{n}_i$ as the number of unused basis payoff functions that can be affected by the i-th strategy of the second player. Formally, given $u_1 \in U$, let π_i be the permutation of basis functions for u_1 as defined in Definition 4.21. Define

$$\begin{aligned} \mathfrak{n}_i &\coloneqq |\mathfrak{B}_i| \\ \text{where } \mathfrak{B}_i &\coloneqq \{\mathfrak{b} \in \mathfrak{B}_* \mid \exists k \in [\![1,\, |\mathcal{S}_1|]\!] :\ \mathfrak{b}(\boldsymbol{e}_k,\, \boldsymbol{e}_i) \ne 0\} \\ \mathfrak{B}_* &\coloneqq \left\{\mathfrak{b}_{\pi_j} \;\middle|\; j \in [\![c_1+1,\, B]\!]\right\} \end{aligned}$$

An instance is *strategy-separable* for the first player if and only if for all $i \in [\![1, |\mathcal{S}_2|]\!]$, $\mathfrak{n}_i \leq c_2 - c_1$. ◁

Example 4.4
In an instance of a masked payoff game (Example 2.1), if $c_1 < c_2$ and there is at most one zero payoff in each column of u_1, then the instance is strategy-separable for the first player.◁

Theorem 4.21
If an instance of a linear component TFZolp-Game game is strategy-separable for the first player, then it is uniformly maximin-attainable for the first player.

Proof. Take an arbitrary $\boldsymbol{x} \in \Delta_{\mathcal{S}_1}$. Let $i \coloneqq \operatorname{argmin}^*_{j \in [\![1, |\mathcal{S}_2|]\!]} \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{e}_j)$ so that $h_1(\boldsymbol{x}) = u(\boldsymbol{x}, \boldsymbol{e}_i) - D(u_1, c_1, \boldsymbol{x}, \boldsymbol{e}_i)$ given (4.64). Since $|\mathfrak{B}_i| \leq c_2 - c_1$, we can choose $u' \in \mathrm{N_C}(u_1, c_1, c_2)$ by setting appropriate coefficients for basis functions in $\mathfrak{B}_i$ such that $u'(\boldsymbol{x}, \boldsymbol{e}_i) = h_1(\boldsymbol{x})$. It holds that

$$\forall \boldsymbol{y} \in \Delta_{\mathcal{S}_2} : \mathcal{P}^+(u', c_2, \boldsymbol{x}, \boldsymbol{y}) \geq \mathcal{P}^-(u', c_2, \boldsymbol{x}, \boldsymbol{y}) \overset{①}{\geq} \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{y}) \overset{②}{\geq} h_1(\boldsymbol{x}), \tag{4.67}$$

where $\overset{①}{\geq}$ is due to $u' \in \alpha^{-1}(u_1, c_1)$ and Theorem 2.6, and $\overset{②}{\geq}$ is due to that $h_1(\boldsymbol{x})$ minimizes $\boldsymbol{y} \mapsto \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{y})$. Since the basis functions in $D(u', c_2, \boldsymbol{x}, \boldsymbol{y})$ are a subset of $\mathfrak{B}_* \setminus \mathfrak{B}_i$, we have $D(u', c_2, \boldsymbol{x}, \boldsymbol{e}_i) = 0$ and thus $\mathcal{P}^+(u', c_2, \boldsymbol{x}, \boldsymbol{e}_i) = u'(\boldsymbol{x}, \boldsymbol{e}_i) = h_1(\boldsymbol{x})$. Combining this with (4.67) yields

$$\min_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^+(u', c_2, \boldsymbol{x}, \boldsymbol{y}) = \mathcal{P}^+(u', c_2, \boldsymbol{x}, \boldsymbol{e}_i) = u'(\boldsymbol{x}, \boldsymbol{e}_i) \tag{4.68}$$

Let $Y(u', \boldsymbol{x})$ be a singleton-valued response function as defined in Definition 4.18. Eq. (4.68) then implies that for $\boldsymbol{y}'$ such that $Y(u', \boldsymbol{x}) = \{\boldsymbol{y}'\}$, it holds that $D(u', c_2, \boldsymbol{x}, \boldsymbol{y}') = 0$. Therefore, we have $g_1(\boldsymbol{x}) \leq \mathcal{P}^-(u', c_2, \boldsymbol{x}, \boldsymbol{y}') = u'(\boldsymbol{x}, \boldsymbol{y}') - D(u', c_2, \boldsymbol{x}, \boldsymbol{y}') = \mathcal{P}^+(u', c_2, \boldsymbol{x}, \boldsymbol{y}') = h_1(\boldsymbol{x})$. Since $h_1(\boldsymbol{x}) \leq g_1(\boldsymbol{x})$ by Lemma 4.13, it follows that $h_1(\boldsymbol{x}) = g_1(\boldsymbol{x})$. □

4.3.3.5 Narrowly reversible games

The previous subsection shows that the strategy-separable instances of linear component games are maximin-attainable. The key property that enables the maximin-attainability is the existence of $u' \in \mathrm{N_C}(u_1, c_1, c_2)$ such that $\mathcal{P}^+(u', c_2, \boldsymbol{x}, \boldsymbol{e}_i) = \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{e}_i)$. This subsection generalizes the property to a broader class of games called *narrowly reversible* games.

Definition 4.23 (Narrowly reversible payoff games)
In a TFZolp-Game instance as defined in Definition 4.12 with $c_1 \leq c_2$, let $T_1^2 \coloneqq \mathrm{N_C}(u_1, c_1, c_2)$. The game is *narrowly reversible* for the first player at capability levels c_1 and c_2 if and only if the following holds:

$$\forall \boldsymbol{x} \in \Delta_{\mathcal{S}_1} : \forall k \in [\![1, |\mathcal{S}_2|]\!] : \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{e}_k) = \inf_{u' \in T_1^2} \mathcal{P}^+(u', c_2, \boldsymbol{x}, \boldsymbol{e}_k) \tag{4.69}$$

If (4.69) holds for any $c_1 \leq c_2$, then the game is *universally narrowly reversible* for the first player. ◁

Remark 4.12 The following holds for any n-player game due to Theorem 2.6. Narrow reversibility is defined by taking equality in the following inequalities.

$$\begin{aligned} \inf_{u \in \mathrm{N_C}(v, c_1, c_2)} \mathcal{P}^+(u, c_2) &\geq \inf_{u \in \mathrm{N_C}(v, c_1, c_2)} \mathcal{P}^-(u, c_2) \geq \mathcal{P}^-(v, c_1) \\ \sup_{u \in \mathrm{N_C}(v, c_1, c_2)} \mathcal{P}^-(u, c_2) &\leq \sup_{u \in \mathrm{N_C}(v, c_1, c_2)} \mathcal{P}^+(u, c_2) \leq \mathcal{P}^+(v, c_1) \end{aligned} \tag{4.70}$$

where $(\boldsymbol{\sigma}_i)_{i \in \mathcal{N}}$ is an arbitrary mixed profile and is omitted

The proof of Theorem 4.21 shows that the strategy-separable linear component games are narrowly reversible. However, it is not necessarily universally narrowly reversible.

For a constant-gap game, if $\delta(c_2) > 0$, then the first inequality in (4.70) becomes strict and therefore it can not be narrowly reversible. In particular, the quantized payoff game in Example 2.2 is not narrowly reversible.

Universally narrowly reversible games exist. As shown in Theorem 3.3, limited-rank games are universally narrowly reversible. Below is another example constructed from a nonuniform rounding function:

Example 4.5 (A universally narrowly reversible quantized payoff game) With a nonuniform payoff rounding function, the quantized payoff game can become universally narrowly reversible. Recursively define two nonuniform rounding functions $r_1, r_2 : \mathbb{R}_{\geq 0} \times \overline{\mathbb{Z}}_+ \to \mathbb{R}_{\geq 0}$ as follows:

- Set $r_1(x, 1) = \lfloor x \rfloor$ and $r_2(x, 1) = \lfloor x \rfloor + 1$ for $x \in \mathbb{R}_{\geq 0}$.
- For $x \in \mathbb{R}_{\geq 0}$ and $2 \leq c < \infty$, find $x' \in \mathbb{R}_{\geq 0}$ such that $l \leq x < u$ or $l = x = u$ where $l \coloneqq r_1(x', c-1)$ and $u \coloneqq r_2(x', c-1)$. Let $m \coloneqq \frac{l+u}{2}$. Consider three cases:
  - If $x = l$, then set $r_1(x, c) = l$ and $r_2(x, c) = l$.
  - If $l < x < m$, then find $k \in \mathbb{N}$ such that $2^{-k-1} \leq \frac{x-l}{m-l} < 2^{-k}$. Set $r_1(x, c) = l + 2^{-k-1}(m-l)$ and $r_2(x, c) = l + 2^{-k}(m-l)$.
  - If $m \leq x < u$, then find $k \in \mathbb{N}$ such that $2^{-k-1} < \frac{u-x}{u-m} \leq 2^{-k}$. Set $r_1(x, c) = u - 2^{-k}(u-m)$ and $r_2(x, c) = u - 2^{-k-1}(u-m)$.
- Set $r_1(x, \infty) = r_2(x, \infty) = x$ for $x \in \mathbb{R}_{\geq 0}$.

Define $[a, a) = (a, a] = \{a\}$ instead of the empty set for any $a \in \mathbb{R}$. For any $x \in \mathbb{R}_{\geq 0}$ and $c \in \mathbb{Z}_+$, define $l \coloneqq r_1(x, c)$ and $u \coloneqq r_2(x, c)$. It is easy to verify the following properties:

- If $x' \in [l, u)$, then $r_1(x', c) = l$ and $r_2(x', c) = u$.
- If $r_1(x', c) = l$ or $r_2(x', c) = u$, then $x' \in [l, u)$.
- The intervals $\{[r_1(x', c), r_2(x', c)) \mid x' \in \mathbb{R}_{\geq 0}\}$ form a partition of $\mathbb{R}_{\geq 0}$.
- The intervals $\{[r_1(x', c+1), r_2(x', c+1)) \mid x' \in [l, u)\}$ form a partition of $[l, u)$.

- Since $r_2(x', c+1) - r_1(x', c+1) \to 0$ when $x' \to l$ or $x' \to u$, it follows that

$$\inf_{x' \in C} r_2(x', c+1) = l, \quad \sup_{x' \in C} r_1(x', c+1) = u$$
$$\text{where } C \coloneqq \{x' \in \mathbb{R}_{\geq 0} \mid r_1(x', c) = l \text{ and } r_1(x', c+1) = x'\}$$

Define the pair (H_c, r) where

$$H_c \coloneqq \{[r_1(x', c), r_2(x', c)) \mid x' \in \mathbb{R}_{\geq 0}\} \cup \{(-r_2(x', c), r_1(x', c)] \mid x' \in \mathbb{R}_{\geq 0}\}$$
$$r(S) \coloneqq \begin{cases} \min S & \text{if } \exists x \in S : \ x \geq 0 \\ \max S & \text{otherwise} \end{cases}$$

Then (H_c, r) satisfies the properties of a hierarchical partition of $\mathbb{R}$ as defined in Definition 2.5. With Theorem 2.4, for a bimatrix game, $\alpha_{\mathrm{n}}(\cdot, \cdot)$ is a payoff perception function with $\alpha_{\mathrm{n}}(\boldsymbol{M}, c) = \boldsymbol{M}'$ where

$$M'_{ij} = \operatorname{sign}(M_{ij}) r_1(|M_{ij}|, c)$$

It is easy to verify that $\alpha_{\mathrm{n}}(\cdot, \cdot)$ is both odd and universally reversible. ◁

It is worth noting that in the general-sum case, even if a game is universally narrowly reversible, it is not necessarily easier to solve. In the proof of Theorem 4.6, one can make the constructed game universally narrowly reversible while still keeping the problem NP-hard by setting $\alpha(2v_1, 1) = \alpha\left(\frac{1}{2}v_1, 1\right) = v_1$. This result is summarized below:

Proposition 4.22 (Hardness of solving general-sum narrowly reversible games) *The decision problem given in Theorem 4.6 is* NP*-hard with the additional constraint that the payoff perception function is universally narrowly reversible.*

By contrast, in the zero-sum case, if an instance is narrowly reversible, then it is maximin-attainable, which makes it efficiently solvable.

Theorem 4.23
In an instance of a TFZOLP-Game game with capability profile (c_1, c_2), if the payoff perception function is narrowly reversible for the first player at capability levels c_1 and c_2, then the instance is uniformly maximin-attainable for the first player.

Proof. Fix $\boldsymbol{x} \in \Delta_{\mathcal{S}_1}$. Let i be the index such that $h_1(\boldsymbol{x}) = \min_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{y}) = \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{e}_i)$, which exists since $\boldsymbol{y} \mapsto \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{y})$ is continuous and concave and Lemma 4.1. Take an arbitrary value $\epsilon \in \mathbb{R}_+$. There exists $u' \in \mathrm{N_C}(u_1, c_1, c_2)$ such that $\mathcal{P}^+(u', c_2, \boldsymbol{x}, \boldsymbol{e}_i) \leq \mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{e}_i) + \epsilon = h_1(\boldsymbol{x}) + \epsilon$ due to (4.69). Let $Y(\cdot, \cdot)$ be an arbitrary singleton-valued response function as defined in Definition 4.18. Let $Y(u', \boldsymbol{x}) = \{\boldsymbol{y}'\}$. It holds that

$$g_1(\boldsymbol{x}) \overset{①}{\leq} \mathcal{P}^-(u', c_2, \boldsymbol{x}, \boldsymbol{y}') \leq \mathcal{P}^+(u', c_2, \boldsymbol{x}, \boldsymbol{y}') \overset{②}{\leq} \mathcal{P}^+(u', c_2, \boldsymbol{x}, \boldsymbol{e}_i) \leq h_1(\boldsymbol{x}) + \epsilon,$$

where $\overset{①}{\leq}$ is follows from taking u' in the infimum in the definition of $g_1(\cdot)$ (see (4.47)), and $\overset{②}{\leq}$ is due to the definition of $Y(u', \boldsymbol{x})$ containing a minimizer of $\boldsymbol{y} \mapsto \mathcal{P}^+(u', c_2, \boldsymbol{x}, \boldsymbol{y})$. Taking $\epsilon \to 0$ yields $g_1(\boldsymbol{x}) \leq h_1(\boldsymbol{x})$. Since $h_1(\boldsymbol{x}) \leq g_1(\boldsymbol{x})$ by Lemma 4.13, it follows that $h_1(\boldsymbol{x}) = g_1(\boldsymbol{x})$. □

4.4 Capability transfer properties

One interesting question in the context of fully-capability-aware games is that given the chance to increase a player's capability at a cost so that the player can perceive a new payoff function with the increased capability and the opponents make new decisions based on the new capability profile, should the player choose to do so? The answer depends on the capability transfer properties of the game. *Capability transfer properties* refer to any properties that characterize how the game outcome changes when a player's capability level changes.

Generally speaking, in a *capability-positive* (resp. capability-negative) game, a player obtains better (resp. worse) payoffs when they increase their capability level. To formalize this concept so that players can make decisions based on the capability transfer properties, this section introduces two more refined capability transfer properties: pessimistic and pragmatic capability-positivity. In a *pessimistically capability-positive* game, a player obtains a better lower bound of expected true payoffs when they increase their capability level and perceive a new game. In a *pragmatically capability-positive* game, a player obtains a better true payoff for all possible true payoff functions when they increase their capability level and perceive a new game. I consider fully capability-aware games in this section where increasing one player's capability level may change other players' strategies.

This section formally defines the above capability transfer properties for both Nash equilibrium and Stackelberg equilibrium in general-sum games. One can extend the definitions to zero-sum games by adding the constraint that the opponent's perception considered by any player also restricts their own possible true payoffs. I will show that all two-player games are pessimistically capability-positive with Stackelberg equilibrium (Theorem 4.25), which also implies that maximin-attainable games are pessimistically capability-positive with Nash equilibrium (Corollary 4.26). This section also presents a sufficient condition for a game to be both pessimistically and pragmatically capability-positive with Nash or Stackelberg equilibrium (Theorem 4.27). Finally, Example 4.6 provides an example of a two-player zero-sum game where the higher-capability player gets worse Nash value if they further increase their capability level. In this example, the increased capability level of the higher-capability player triggers the lower-capability player to defend against more possibilities, which causes the lower-capability player to change their strategy and results in a better true payoff for themselves but a worse true payoff for the higher-capability opponent.

I first formalize the definitions for Nash equilibrium. Let v_i^i be player i's perception of their own payoff function. From this player's perspective, since they do not know what their perceived payoff function will be after the capability level is increased, they need to consider all possibilities in $\mathrm{N_C}\left(v_i^i,\, c_i,\, c_i+1\right)$. They also need to reason about their new perceptions of other players' payoff functions in $\mathrm{N_C}\left(v_j^i,\, c_i,\, \min\left\{c_i+1,\, c_j\right\}\right)$ for $j \neq i$. One caveat is that since only the players with the two highest capability levels can solve Nash equilibrium using their private information (Remark 4.5), the definition is restricted to those players.

Definition 4.24 (Pessimistically capability-positive Nash games)
In an instance of an OLP-Game with capability profile $(c_i)_{i\in\mathcal{N}}$ as defined in Definition 2.3, let $\mathfrak{N}(v_1,\, \ldots,\, v_n,\, c_1,\, \ldots,\, c_n) \subseteq \prod_{i\in\mathcal{N}} \Delta_{\mathcal{S}_i}^U$ be the set of Nash equilibrium response functions (Definition 4.7) in this instance when a top-two-capability player perceives the payoff func-

tions as $v_1, \ldots, v_n$. Let $i \in \mathcal{N}$ be a top-two-capability player, i.e., $|\{c_j \mid j \in \mathcal{N} \text{ and } c_j \geq c_i\}| \leq 2$. The instance is *pessimistically capability-positive* with Nash equilibrium for the player i if and only if

$$
\begin{aligned}
&\forall v_i^{i\prime} \in \mathrm{N_C}\left(v_i^i,\, c_i,\, c_i+1\right) :\ \forall v_j^{i\prime} \in \mathrm{N_C}\left(v_j^i,\, c_i,\, \min\left\{c_i+1,\, c_j\right\}\right) \text{ for } j \neq i : \\
&\inf_{(\boldsymbol{R}_1', \ldots, \boldsymbol{R}_n') \in S'} f_i'\left(v_i^{i\prime},\, \boldsymbol{R}_i'(v_i^{i\prime}) \;\middle|\; \boldsymbol{R}_{-i}'\right) \geq \sup_{(\boldsymbol{R}_1, \ldots, \boldsymbol{R}_n) \in S} f_i\left(v_i^i,\, \boldsymbol{R}_i(v_i^i) \;\middle|\; \boldsymbol{R}_{-i}\right) \\
&\qquad \text{where } f_i \text{ is defined as in (4.11)} \\
&\qquad\quad S \coloneqq \mathfrak{N}\Big(\left(v_k^i\right)_{k\in\mathcal{N}},\, (c_k)_{k\in\mathcal{N}}\Big) \\
&\qquad\quad S' \coloneqq \mathfrak{N}\Big(\left(v_k^{i\prime}\right)_{k\in\mathcal{N}},\, (c_i+1,\, c_{-i})\Big) \\
&\qquad\quad f_i' \text{ is defined analogously to } f_i \text{ corresponding to } S'
\end{aligned}
\tag{4.71}
$$

◁

By comparison, the pragmatically capability-positive property considers all possible true payoff functions instead of the lower bound:

Definition 4.25 (Pragmatically capability-positive Nash games)
With the setup and notation in Definition 4.24, the instance with capability profile $(c_i)_{i\in\mathcal{N}}$ is *pragmatically capability-positive* with Nash equilibrium for the player i if and only if

$$
\begin{aligned}
&\forall v_i^{i\prime} \in \mathrm{N_C}\left(v_i^i,\, c_i,\, c_i+1\right) :\ \forall v_j^{i\prime} \in \mathrm{N_C}\left(v_j^i,\, c_i,\, \min\left\{c_i+1,\, c_j\right\}\right) \text{ for } j \neq i : \\
&\qquad\qquad \forall u_i' \in \alpha^{-1}\big(v_i^{i\prime},\, c_i+1\big) : \\
&\inf_{(\boldsymbol{R}_1', \ldots, \boldsymbol{R}_n') \in S'} u_i'\big(\boldsymbol{R}_j'(v_j^{j\prime})_{j\in\mathcal{N}}\big) \geq \sup_{(\boldsymbol{R}_1, \ldots, \boldsymbol{R}_n) \in S} u_i'\big(\boldsymbol{R}_j(v_j^j)_{j\in\mathcal{N}}\big),
\end{aligned}
\tag{4.72}
$$

where $v_j^{j\prime} = v_j^j$ for $j \neq i$. ◁

Remark 4.13 Definition 4.25 reuses the setup in Definition 4.24, which assumes that the player i is among the players with the two highest capability levels. When player i has the second highest capability level, they are unable to calculate $\boldsymbol{R}_j(v_j^j)$ if $c_j > c_i$, but under certain circumstances they can still ensure that the game is pragmatically capability-positive for them as will be shown in Theorem 4.27.

When Nash equilibria are unique in the original instance and the higher-capability instance (i.e., $|S| = |S'| = 1$), and if player i has the highest capability level (i.e., no uncertainty about v_j^j for $j \neq i$), the pragmatic capability-positivity implies the pessimistic capability-positivity since the infimum and supremum in (4.71) and (4.72) can be removed, while f_i' and f_i implicitly take the infimum over $\alpha^{-1}(v_i^{i\prime}, c_i+1)$ and $\alpha^{-1}(v_i^i, c_i)$, respectively. However, in the general case, one of the capability-positive properties does not necessarily imply the other.

Analogously, the capability-positive properties can be defined for the Stackelberg equilibrium. Since Stackelberg value is unique, the pessimistic capability-positive games are easier to define.

Definition 4.26 (Pessimistically capability-positive Stackelberg games)
In a two-player fully-capability-aware game, assume without loss of generality that the first player is the leader. Assume the same response function $Y(\cdot)$ is used by two instances with

capability profiles (c_1, c_2) and $(c_1 + 1, c_2)$. The instance with capability profile (c_1, c_2) is *pessimistically capability-positive* with Stackelberg equilibrium for the first player if and only if

$$\begin{aligned} &\forall u_1' \in \mathrm{N_C}\,(u_1, c_1, c_1 + 1) : \forall v_1' \in \mathrm{N_C}\,(v_1, c_1, \min\{c_1 + 1, c_2\}) : \\ &\qquad V(u_1', v_1', c_1 + 1) \geq V(u_1, v_1, c_1) \\ &\qquad \text{where } V(u, v, c) \coloneqq \sup_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} \inf_{v_2' \in \mathrm{N_C}\,(v, c, c_2)} \inf_{\boldsymbol{y} \in Y(v_2', \boldsymbol{x})} \mathcal{P}^-(u, c, \boldsymbol{x}, \boldsymbol{y}) \end{aligned} \tag{4.73}$$

Of note, $V(u_1, v_1, c_1)$ is Stackelberg value of the first player in the lower-capability instance as defined in Definition 4.9. ◁

All games are pessimistically capability-positive with Stackelberg equilibrium. The proof uses the following lemma:

Lemma 4.24
Given a payoff perception function, for any integers c_1, c_2 and a payoff function $v_1 \in U$ such that $\bar{c}(v_1) \leq c_1$, for any $v_1' \in \mathrm{N_C}\,(v_1, c_1, \min\{c_1 + 1, c_2\})$, it holds that

$$\mathrm{N_C}\,(v_1', c_1 + 1, c_2) \subseteq \mathrm{N_C}\,(v_1, c_1, c_2)$$

Proof. Take any $v_2' \in \mathrm{N_C}\,(v_1', c_1 + 1, c_2)$. If $c_1 + 1 \geq c_2$, then $v_2' = v_1'$ due to Theorem 4.3. Since $v_1' \in \mathrm{N_C}\,(v_1, c_1, \min\{c_1 + 1, c_2\}) = \mathrm{N_C}\,(v_1, c_1, c_2)$, it holds that $v_2' \in \mathrm{N_C}\,(v_1, c_1, c_2)$. If $c_1 + 1 < c_2$, then with Definition 4.6 we have $\alpha(v_2', c_1 + 1) = \alpha(v_1', c_1 + 1) = v_1'$, $\bar{c}(v_2') \leq c_2$, and $\alpha(v_1', c_1) = v_1$, which together imply $\alpha(v_2', c_1) = v_1$ and thus $v_2' \in \mathrm{N_C}\,(v_1, c_1, c_2)$. Therefore, $\mathrm{N_C}\,(v_1', c_1 + 1, c_2) \subseteq \mathrm{N_C}\,(v_1, c_1, c_2)$. □

Theorem 4.25
All instances of any two-player fully-capability-aware game (Definition 4.4) are pessimistically capability-positive with Stackelberg equilibrium for any player.

Proof. Since $u_1 = \alpha(u_1', c_1)$, Theorem 2.6 states that $\mathcal{P}^-(u_1, c_1, \boldsymbol{x}, \boldsymbol{y}) \leq \mathcal{P}^-(u_1', c_1 + 1, \boldsymbol{x}, \boldsymbol{y})$ for any $\boldsymbol{x} \in \Delta_{\mathcal{S}_1}$ and $\boldsymbol{y} \in \Delta_{\mathcal{S}_2}$. Combining this with Lemma 4.24 implies that (4.73) holds universally. □

If an instance is maximin-attainable for the first player, Theorem 4.15 states that all Nash values are the same as the player's maximin value and Stackelberg value, which yields the following result when combined with Theorem 4.25.

Corollary 4.26
If two TFZOLP-Game instances (Definition 4.12) with capability profiles (c_1, c_2) and $(c_1 + 1, c_2)$ are both maximin-attainable for the first player (Definition 4.18), then the first instance is pessimistically capability-positive with Nash equilibrium for the first player.

Next comes the definition of the pragmatic capability-positivity for the Stackelberg equilibrium. Since the leader can choose their strategy first, the following definition differs from the definition for Nash equilibrium (Definition 4.25) in that it needs only one strategy profile that leads to nondecreasing true payoffs. The definition assumes for simplicity that Stackelberg value is attainable at some strategy profile (cf. Remark 4.6); if Stackelberg value is not attainable, the definition can be extended with sequences of strategies that converge to the Stackelberg value.

Definition 4.27 (Pragmatically capability-positive Stackelberg games) With the setup and notation in Definition 4.26, assume that Stackelberg values are attainable by some strategies. Let $\mathfrak{S}(u,\, v,\, c) \subseteq \Delta_{\mathcal{S}_1}$ be the set of Stackelberg strategies that attain $V(u,\, v,\, c)$ defined in (4.73). The instance with the capability profile $(c_1,\, c_2)$ is *pragmatically capability-positive* with the Stackelberg equilibrium for the first player if and only if

$$\begin{aligned} &\forall u_1' \in \mathrm{N_C}\left(u_1,\, c_1,\, c_1+1\right) :\, \forall v_1' \in \mathrm{N_C}\left(v_1,\, c_1,\, \min\left\{c_1+1,\, c_2\right\}\right) : \\ &\exists \boldsymbol{x}' \in \mathfrak{S}(u_1',\, v_1',\, c_1+1) :\, \forall u' \in \alpha^{-1}(u_1',\, c_1+1) : \\ &\inf_{v_2' \in \mathrm{N_C}\left(v_1',\, c_1+1,\, c_2\right)} \inf_{\boldsymbol{y} \in Y\left(v_2',\, \boldsymbol{x}'\right)} u'(\boldsymbol{x}',\, \boldsymbol{y}) \geq \sup_{\boldsymbol{x} \in \mathfrak{S}(u_1,\, v_1,\, c_1)} \inf_{v_2' \in \mathrm{N_C}\,(v_1,\, c_1,\, c_2)} \inf_{\boldsymbol{y} \in Y\left(v_2',\, \boldsymbol{x}'\right)} u'(\boldsymbol{x},\, \boldsymbol{y}) \end{aligned} \tag{4.74}$$

◁

Assuming $Y(v,\, \boldsymbol{x}') = Y(v,\, \boldsymbol{x})$, one sufficient condition for (4.74) to hold is that all payoff functions in $\alpha^{-1}(u_1',\, c_1+1)$ share one pure dominant strategy for the first player so that the optimal strategy of any true payoff function can be found after the capability level is increased. The following theorem formalizes this idea.

Theorem 4.27 (A sufficient condition for capability-positive games) *In a two-player fully-capability-aware game, assume the second player has a dominant strategy $\boldsymbol{y}_v$ for any $v \in \mathrm{N_C}\left(v_1,\, c_1,\, c_2\right)$; i.e.,*

$$\forall v \in \mathrm{N_C}\left(v_1,\, c_1,\, c_2\right) :\, \boldsymbol{y}_v \in \bigcap_{\boldsymbol{x} \in \Delta_{\mathcal{S}_1}} \operatorname*{argmax}_{\boldsymbol{y} \in \Delta_{\mathcal{S}_2}} \mathcal{P}^-(v,\, c_2,\, \boldsymbol{x},\, \boldsymbol{y})$$

Assume the second player uses $\boldsymbol{y}_v$ in Nash or Stackelberg equilibrium. If the following condition holds, then the instance with capability profile $(c_1,\, c_2)$ is both pessimistically and pragmatically capability-positive with Nash or Stackelberg equilibrium for the first player:

$$\begin{aligned} &\forall u_1' \in \mathrm{N_C}\left(u_1,\, c_1,\, c_1+1\right) :\, \exists i \in \mathcal{S}_1 : \\ &\forall u' \in \alpha^{-1}(u_1',\, c_1+1) :\, \forall v_2' \in \mathrm{N_C}\left(v_1,\, c_1,\, c_2\right) :\, \forall j \in \mathcal{S}_1 : \\ &u'\left(\boldsymbol{e}_i,\, \boldsymbol{y}_{v_2'}\right) \geq u'\left(\boldsymbol{e}_j,\, \boldsymbol{y}_{v_2'}\right) \end{aligned} \tag{4.75}$$

Proof. Since the second player has a dominant strategy, the first player's Nash strategy/-value and Stackelberg strategy/value are the same. Therefore, the following proof only needs to work with Stackelberg equilibrium.

Take any $u_1' \in \mathrm{N_C}\left(u_1,\, c_1,\, c_1+1\right)$. Let $i \in \mathcal{S}_1$ be the index satisfying (4.75). Take any $v_1' \in \mathrm{N_C}\left(v_1,\, c_1,\, \min\left\{c_1+1,\, c_2\right\}\right)$.

The Stackelberg objective function in the higher-capability instance is

$$g_1'(\boldsymbol{x}) = \inf_{v_2' \in \mathrm{N_C}\left(v_1',\, c_1+1,\, c_2\right)} \mathcal{P}^-\left(u_1',\, c_1+1,\, \boldsymbol{x},\, \boldsymbol{y}_{v_2'}\right) = \inf_{v_2' \in \mathrm{N_C}\left(v_1',\, c_1+1,\, c_2\right)} \inf_{u' \in \alpha^{-1}\left(u_1',\, c_1+1\right)} u'\left(\boldsymbol{x},\, \boldsymbol{y}_{v_2'}\right)$$

Since $\mathrm{N_C}\left(v_1',\, c_1+1,\, c_2\right) \subseteq \mathrm{N_C}\left(v_1,\, c_1,\, c_2\right)$ by Lemma 4.24, it follows that $u'\left(\boldsymbol{x},\, \boldsymbol{y}_{v_2'}\right) \leq u'\left(\boldsymbol{e}_i,\, \boldsymbol{y}_{v_2'}\right)$ due to (4.75), which implies that $g_1'(\boldsymbol{x}) \leq g_1'(\boldsymbol{e}_i)$ for any $\boldsymbol{x} \in \Delta_{\mathcal{S}_1}$. Therefore, $\boldsymbol{e}_i \in \mathfrak{S}(u_1',\, v_1',\, c_1+1)$. Setting $\boldsymbol{x}' = \boldsymbol{e}_i$ and plugging in (4.75) proves (4.74), i.e., the game is pragmatically capability-positive.

To prove that the game is pessimistically capability-positive, consider the Stackelberg objective function in the lower-capability instance:

$$g_1(\boldsymbol{x}) = \inf_{v_2' \in \mathrm{N_C}\,(v_1,\, c_1,\, c_2)} \mathcal{P}^-\left(u_1,\, c_1,\, \boldsymbol{x},\, \boldsymbol{y}_{v_2'}\right) = \inf_{v_2' \in \mathrm{N_C}\,(v_1,\, c_1,\, c_2)} \inf_{u' \in \alpha^{-1}(u_1,\, c_1)} u'\left(\boldsymbol{x},\, \boldsymbol{y}_{v_2'}\right)$$

Since $\mathrm{N_C}(v_1', c_1+1, c_2) \subseteq \mathrm{N_C}(v_1, c_1, c_2)$ (Lemma 4.24), $\alpha^{-1}(u_1', c_1+1) \subseteq \alpha^{-1}(u_1, c_1)$ (Theorem 2.2), and $u'(\boldsymbol{e}_i, \boldsymbol{y}_{v_2'}) \geq u'(\boldsymbol{x}, \boldsymbol{y}_{v_2'})$ for any $\boldsymbol{x} \in \Delta_{\mathcal{S}_1}$ (the assumption (4.75)), it follows that $g_1'(\boldsymbol{e}_i) \geq g_1(\boldsymbol{x})$. Therefore, the game is pessimistically capability-positive. □

One may conjecture that in a totally competitive setting (i.e., a zero-sum game), a player who increases their capability level gets better payoffs. However, this is not always the case. The following example presents a two-player zero-sum game that is neither pessimistically nor pragmatically capability-positive with Nash equilibrium for the higher-capability player.

Example 4.6 (Lower Nash value with increased capability in a zero-sum game) This example demonstrates a two-player fully-capability-aware zero-sum game where increasing the capability level of the higher-capability player results in both lower payoff bound and lower true payoff for this player. Furthermore, the lower-capability player's Nash strategy coincides with their Stackelberg strategy in each instance. The change of payoff is caused by that the lower-capability player needs to defend against more possibilities when the opponent has a higher capability level, which causes the lower-capability player to change their strategy.

In this game, each player has two strategies. The payoff function is represented by a matrix. The first player is the row player and the second player is the column player. Define the following matrices:

$$\boldsymbol{P}_1 = \begin{bmatrix} 3 & 1 \\ 2 & 3 \end{bmatrix}, \quad \boldsymbol{P}_2 = \begin{bmatrix} 3 & 4 \\ 4 & 5 \end{bmatrix}, \quad \boldsymbol{Q} = \begin{bmatrix} 3 & 1 \\ 4 & 3 \end{bmatrix}$$

Define the function $\alpha : \mathbb{R}^{2\times 2} \times \overline{\mathbb{Z}}_+ \to \mathbb{R}^{2\times 2}$ as:

$$\alpha(\boldsymbol{M}, c) \coloneqq \begin{cases} -\alpha_0(-\boldsymbol{M}, c) & \text{if } M_{ij} < 0 \text{ for all } i \in \{1, 2\}, j \in \{1, 2\} \\ \alpha_0(\boldsymbol{M}, c) & \text{otherwise} \end{cases}$$

$$\alpha_0(\boldsymbol{M}, c) \coloneqq \begin{cases} \boldsymbol{P}_1 & \text{if } \boldsymbol{M} \in \{\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{Q}\} \text{ and } c \leq 2 \\ \boldsymbol{M} & \text{otherwise} \end{cases}$$

It is easy to verify that $\alpha(\cdot, \cdot)$ is an odd payoff perception function. The true payoff function is $u^* = \boldsymbol{P}_2$. In the following calculation, the row player's goal is to maximize the game value, while the column player's goal is to minimize the game value.

In the first game instance, the capability profile is $c_1 = 1$ and $c_2 = 2$. The perceived payoff functions are $(\boldsymbol{P}_1, -\boldsymbol{P}_1)$ for both players, and $T_1^2 = \{\boldsymbol{P}_1\}$. Since $\boldsymbol{P}_1 \leq \boldsymbol{Q} \leq \boldsymbol{P}_2$ holds element-wise, $\mathcal{P}^-(\boldsymbol{P}_1, 1, \boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\intercal \boldsymbol{P}_1 \boldsymbol{y}$ and $\mathcal{P}^+(\boldsymbol{P}_1, 2, \boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\intercal \boldsymbol{P}_2 \boldsymbol{y}$ for any $(\boldsymbol{x}, \boldsymbol{y}) \in \Delta_2^2$. Therefore, $\boldsymbol{Y}(\boldsymbol{P}_1, \boldsymbol{x}) = \boldsymbol{e}_1$ for any $\boldsymbol{x} \in \Delta_2$ since the first column is the dominant strategy of $\boldsymbol{P}_2$ for the second player (whose goal is to minimize). There is only one Nash equilibrium $(\boldsymbol{e}_1, \boldsymbol{Y})$. It is easy to verify that the row player's Stackelberg strategy is also $\boldsymbol{e}_1$. The Nash value and the Stackelberg value of the row player are both $\mathcal{P}^-(\boldsymbol{P}_1, 1, \boldsymbol{e}_1, \boldsymbol{e}_1) = 3$. The Nash value and Stackelberg value of the column player are both $\mathcal{P}^+(\boldsymbol{P}_1, 2, \boldsymbol{e}_1, \boldsymbol{e}_1) = 3$. The true payoff in this instance is 3.

In the second game instance, the column player increases their capability level to $c_2' = 3$. The perceived payoff functions are $(\boldsymbol{P}_1, -\boldsymbol{P}_1)$ for the row player and $(\boldsymbol{P}_2, -\boldsymbol{P}_2)$ for the

column player. The row player now has $T_1^{2\prime} = \{\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{Q}\}$. Let the row player's mixed strategy be $\boldsymbol{x} = \begin{bmatrix} a & 1-a \end{bmatrix}^\intercal$. For any choice of the response function $\boldsymbol{Y}'(\cdot, \cdot)$ satisfying (4.42), it holds that

$$\begin{aligned}
\mathcal{P}^-(\boldsymbol{P}_1, 3, \boldsymbol{x}, \boldsymbol{Y}'(\boldsymbol{P}_1, \boldsymbol{x})) &= \min\{a+2, -2a+3\} \\
\mathcal{P}^-(\boldsymbol{P}_2, 3, \boldsymbol{x}, \boldsymbol{Y}'(\boldsymbol{P}_2, \boldsymbol{x})) &= -a+4 \\
\mathcal{P}^-(\boldsymbol{Q}, 3, \boldsymbol{x}, \boldsymbol{Y}'(\boldsymbol{Q}, \boldsymbol{x})) &= -2a+3
\end{aligned}$$

The row player's goal is to solve

$$\boldsymbol{x}^* \in \operatorname*{argmax}_{\boldsymbol{x} \in \Delta_2} \min_{\boldsymbol{M} \in \{\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{Q}\}} \mathcal{P}^-(\boldsymbol{M}, 3, \boldsymbol{x}, \boldsymbol{Y}'(\boldsymbol{M}, \boldsymbol{x}))$$

The row player's Nash strategy and Stackelberg strategy are both $\boldsymbol{x}^* = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \end{bmatrix}^\intercal$ with the response function $\boldsymbol{Y}'(\boldsymbol{P}_1, \boldsymbol{x}^*) = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \end{bmatrix}^\intercal$ and $\boldsymbol{Y}'(\boldsymbol{P}_2, \boldsymbol{x}^*) = \boldsymbol{e}_1$. The row player's Nash value and Stackelberg value are both $\frac{7}{3}$. The column player's Nash value and Stackelberg value are both $\boldsymbol{x}^{*\intercal} \boldsymbol{P}_2 \boldsymbol{Y}'(\boldsymbol{x}^*) = \frac{11}{3}$, which is also the true payoff.

In this example, both players get worse Nash and Stackelberg values (when the row player is the leader) from their own perspectives when the column player increases their capability by 1. The row player gets better true payoff and the column player gets worse true payoff.◁

Chapter 5

Capability transfer properties in limited-rank games via numerical simulation

Consider a world where agents compete with each other in a series of one-shot simultaneous-move games. All agents are rational, but each agent may have rank capability that limits their perception of the games. It's either that all agents are capability-oblivious or all agents are fully-capability-aware. An interesting question is whether having a higher capability level gives an agent a better expected payoff in the long run. Of note, a capability-oblivious agent may tend to have a lower capability since for a low-rank general-sum game, a Nash equilibrium can be computed in polynomial time with support enumeration [LMM03].

Motivated by the above question, this chapter investigates the following questions via numerical experiments on large limited-rank games:

1. How is a player's payoff related to their capability level? This chapter considers the expected payoff of the row player for all possible capability profiles in four settings: capability-oblivious zero-sum games (denoted as oz), capability-oblivious general-sum games (denoted as og), fully-capability-aware zero-sum games (denoted as az), and fully-capability-aware general-sum games (denoted as ag).

2. Does a player get a better payoff in the fully-capability-aware setting compared to the capability-oblivious setting, in either zero-sum or general-sum games?

The fully-capability-aware general-sum setting needs special attention. Since there is no known practical algorithm to solve the Nash equilibrium in this setting defined by Definition 4.5, the lower-capability player is assumed to use the maximin strategy in the general-sum fully-capability-aware setting, while the higher-capability player uses the best response, similar to the solution to the zero-sum case given by Corollary 3.4.

The numerical experiment uses two sets of games:

- The first set is random bimatrix games $(\boldsymbol{A}^i, \boldsymbol{B}^i)$ for $i \in [\![1, N]\!]$ with $N = 120,000$. For each i, the game size is 10×10, and all entries of $\boldsymbol{A}^i$ and $\boldsymbol{B}^i$ are independently drawn from the standard normal distribution.
- The second set has only one game, the normal-form representation of Kuhn poker [Kuh50] that will be described in Section 5.2.

For each random game $(\boldsymbol{A}^i, \boldsymbol{B}^i)$, the following strategy profiles are computed:

- For $c \in [\![1, 10]\!]$, $(\boldsymbol{x}^{i,c,\mathrm{oz}}, \boldsymbol{y}^{i,c,\mathrm{oz}})$ is the capability-oblivious Nash equilibrium in the zero-sum game $\left(\widetilde{\boldsymbol{A}^i}^{(c)}, -\widetilde{\boldsymbol{A}^i}^{(c)}\right)$, which is the maximin strategy solved using linear programming. Of note, the set of zero-sum games with a unique Nash equilibrium is dense in the set of all zero-sum games [BKS50, Theorem 3].
- For $c \in [\![1, 10]\!]$, $(\boldsymbol{x}^{i,c,\mathrm{og}}, \boldsymbol{y}^{i,c,\mathrm{og}})$ is a capability-oblivious Nash equilibrium in the general-sum game $\left(\widetilde{\boldsymbol{A}^i}^{(c)}, \widetilde{\boldsymbol{B}^i}^{(c)}\right)$. It is the simplest equilibrium (Definition 3.4) among the set of Nash equilibria found by the Lemke-Howson algorithm [LH64].
- For $(c_1, c_2) \in [\![1, 10]\!] \times [\![1, 10]\!]$ and $c_1 \neq c_2$, $\left(\boldsymbol{x}^{i,(c_1,c_2),\mathrm{az}}, \boldsymbol{y}^{i,(c_1,c_2),\mathrm{az}}\right)$ is the strategy profile defined by Corollary 3.4 of the instance with the capability profile (c_1, c_2) in the fully-capability-aware zero-sum game $(\boldsymbol{A}^i, -\boldsymbol{A}^i)$. The case of $c_1 = c_2$ is excluded since it involves computing the Nash equilibrium of a nonlinear concave game, for which there are no practical general implementations to the best of the author's knowledge.
- For $(c_1, c_2) \in [\![1, 10]\!] \times [\![1, 10]\!]$ and $c_1 \neq c_2$, $\left(\boldsymbol{x}^{i,(c_1,c_2),\mathrm{ag}}, \boldsymbol{y}^{i,(c_1,c_2),\mathrm{ag}}\right)$ is the strategy profile defined as follows. If $c_1 < c_2$, then $\boldsymbol{x}^{i,(c_1,c_2),\mathrm{ag}} := \boldsymbol{x}^{i,(c_1,c_2),\mathrm{az}}$ and $\boldsymbol{y}^{i,(c_1,c_2),\mathrm{ag}} = \operatorname{argmax}^*_{\boldsymbol{y} \in \Delta_{10}} \mathcal{P}^-\left(\widetilde{\boldsymbol{B}^i}^{(c_2)}, c_2, \boldsymbol{x}^{i,(c_1,c_2),\mathrm{ag}}, \boldsymbol{y}\right)$. They are defined analogously if $c_1 > c_2$.

For Kuhn poker, only the strategy profiles related to the zero-sum case are computed.

Capability-oblivious general-sum games are solved via the Lemke-Howson algorithm implemented by the `Gambit 16.1.1` library [MMT24]. The Second-Order Cone Programs for capability-aware players are described using the `CVXPY 1.4.1` modeling language [DB16] and solved by the `Clarabel 0.6.0` solver [GC21]. The experiment is conducted on a machine with an AMD Ryzen Threadripper 2970WX 24-Core Processor and 125GiB of RAM running Arch Linux.

5.1 Random games

For the random games, if solving $(\boldsymbol{x}^{i,c,\mathrm{og}}, \boldsymbol{y}^{i,c,\mathrm{og}})$ takes more than 100 seconds for any i or c, the game $(\boldsymbol{A}^i, \boldsymbol{B}^i)$ is discarded. Let T be the set of indices of the games that are not discarded. Formally, $i \in T$ if and only if computing $(\boldsymbol{x}^{i,c,\mathrm{og}}, \boldsymbol{y}^{i,c,\mathrm{og}})$ for any c takes no more than 100 seconds. There are $|T| = 117,577$ games kept.

To validate the statistical significance of the observations, the standard deviations of the calculated mean values are estimated by bootstrapping. For each mean value μ reported

in the remainder of this section (i.e., $\mu = \overline{P}^t_{c_2,\,c_1}$ for $t \in \{\text{oz, og, az}\}$ or $\mu = \overline{D}^t_{c_2,\,c_1}$ for $t \in \{\text{az, ag}\}$, where $\overline{\boldsymbol{P}}^t$ and $\overline{\boldsymbol{D}}^t$ are defined below), the standard deviation $\text{sd}\,(\mu)$ is estimated by bootstrapping for 1,000 rounds over samples in T. The absolute and relative upper bounds are $\text{sd}\,(\mu) \leq 0.0034$ and $\frac{\text{sd}\,(\mu)}{|\mu|+10^{-3}} \leq 1.7222$, respectively. Therefore, observations made in this section are statistically significant most capability levels.

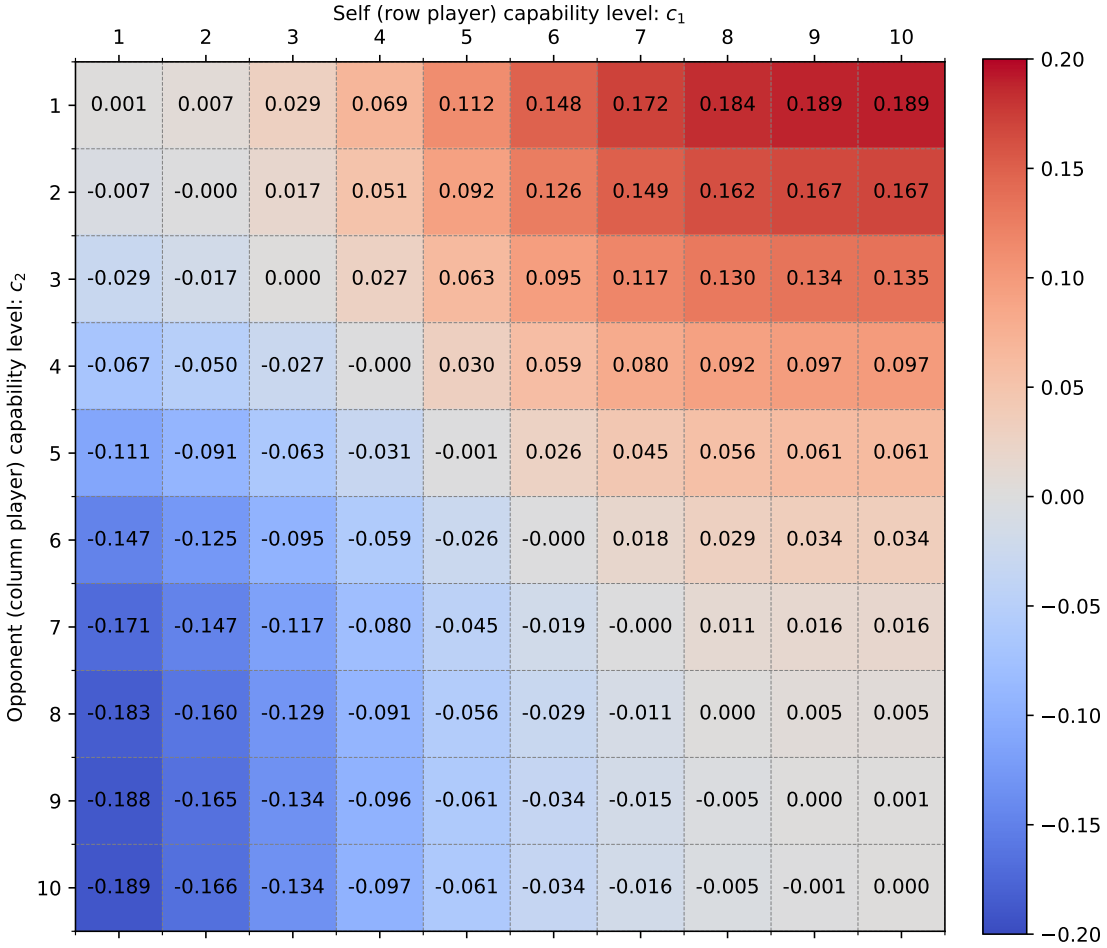


Figure 5.1: Heatmap of $\overline{\boldsymbol{P}}^{\text{oz}}$, the row player's mean payoff of different capability levels in the capability-oblivious zero-sum setting; defined in (5.1).

Figure 5.1 visualizes $\overline{\boldsymbol{P}}^{\text{oz}}$, the row player's mean payoff over the games in T for different capability levels in the capability-oblivious zero-sum setting, which is calculated as

$$\overline{\boldsymbol{P}}^{\text{oz}} := \frac{1}{|T|}\sum_{i\in T} \boldsymbol{P}^{i,\,\text{oz}}, \quad \text{where } P^{i,\,\text{oz}}_{c_2,\,c_1} := \boldsymbol{x}^{i,\,c_1,\,\text{oz}\intercal}\boldsymbol{A}^i\boldsymbol{y}^{i,\,c_2,\,\text{oz}} \tag{5.1}$$

A prominent pattern in Fig. 5.1 is that the row player's mean payoff increases as the capability level of the row player increases and decreases with increasing capability level of the column

player. This observation is in line with Theorem 3.10. Theorem 3.10 implies that

$$\begin{aligned} v_i - 2\sigma_{c_1}(\boldsymbol{A}^i) &\le \min_{j_2 \in [\![1,\,10]\!]} \boldsymbol{x}^{i,\,c_1,\,\mathrm{oz}\intercal} \boldsymbol{A}^i \boldsymbol{e}_{j_2} \le \boldsymbol{x}^{i,\,c_1,\,\mathrm{oz}\intercal} \boldsymbol{A}^i \boldsymbol{y}^{i,\,c_2,\,\mathrm{oz}} \\ &\le \max_{j_1 \in [\![1,\,10]\!]} \boldsymbol{e}_{j_1}^{\intercal} \boldsymbol{A}^i \boldsymbol{y}^{i,\,c_2,\,\mathrm{oz}} \le v_i + 2\sigma_{c_2}(\boldsymbol{A}^i), \end{aligned}$$

where v_i is the value of the zero-sum game $(\boldsymbol{A}^i, -\boldsymbol{A}^i)$. Since $\mathbb{E}[v_i] = 0$, if we assume the distribution of the true payoff among the bound $[v_i - 2\sigma_{c_1}(\boldsymbol{A}^i),\, v_i + 2\sigma_{c_2}(\boldsymbol{A}^i)]$ is symmetric, then $\mathbb{E}[\boldsymbol{x}^{i,\,c_1,\,\mathrm{oz}\intercal} \boldsymbol{A}^i \boldsymbol{y}^{i,\,c_2,\,\mathrm{oz}}] = \mathbb{E}[\sigma_{c_2}(\boldsymbol{A}^i)] - \mathbb{E}[\sigma_{c_1}(\boldsymbol{A}^i)]$. Since $\mathbb{E}[\sigma_c(\boldsymbol{A}^i)]$ is monotonically decreasing with c, one can expect the mean payoff to increase with increasing capability level of the row player or decreasing capability level of the column player.

The monotonicity of payoff discussed above does not hold in the individual games. Let $r_i \in \mathbb{N}$ be the number of capability profiles that violate the pattern in the i-th game; formally, r_i is the number of pairs $(c_1,\, c_2)$ for which $D^{i,\,\mathrm{oz}}_{c_2,\,c_1} < D^{i,\,\mathrm{oz}}_{c_2+1,\,c_1}$ or $D^{i,\,\mathrm{oz}}_{c_2,\,c_1} > D^{i,\,\mathrm{oz}}_{c_2,\,c_1+1}$. Then the median pattern-violation rate over all games, i.e., the median of $\{r_i/(10 \times 9 \times 2) \mid i \in T\}$, is 43.33%; the minimum and maximum pattern-violation rates are 17.78% and 66.11%, respectively.

Figure 5.2 shows $\overline{\boldsymbol{P}}^{\mathrm{og}}$, the row player's mean payoff over the games in T for different capability levels in the capability-oblivious general-sum setting:

$$\overline{\boldsymbol{P}}^{\mathrm{og}} \coloneqq \frac{1}{|T|} \sum_{i \in T} \boldsymbol{P}^{i,\,\mathrm{og}}, \quad \text{where } P^{i,\,\mathrm{og}}_{c_2,\,c_1} \coloneqq \boldsymbol{x}^{i,\,c_1,\,\mathrm{og}\intercal} \boldsymbol{A}^i \boldsymbol{y}^{i,\,c_2,\,\mathrm{og}} \tag{5.2}$$

An interesting observation from Fig. 5.2 is that the payoff is maximized when the row player's capability level matches column player's capability level. This observation is consistent with Theorem 3.9 since when both players have the same capability level, they perceive the same game, and therefore their strategy profile is an approximate Nash equilibrium of the true game. However, this pattern is again not universally true for individual games. Define the pattern violation rate of the game $(\boldsymbol{A}^i,\, \boldsymbol{B}^i)$ as the proportion of diagonal entries in $\boldsymbol{P}^{i,\,\mathrm{og}}$ that are not the maximum of their corresponding row or column. The median, minimum, and maximum pattern-violation rates over all games in T are 50.00%, 0.00%, and 100.00%, respectively.

Figure 5.3 visualizes $\overline{\boldsymbol{P}}^{\mathrm{az}}$, the row player's mean payoff over the games in T for different capability profiles in the fully-capability-aware zero-sum setting:

$$\overline{\boldsymbol{P}}^{\mathrm{az}} \coloneqq \frac{1}{|T|} \sum_{i \in T} \boldsymbol{P}^{i,\,\mathrm{az}}, \quad \text{where } P^{i,\,\mathrm{az}}_{c_2,\,c_1} \coloneqq \boldsymbol{x}^{i,\,(c_1,\,c_2),\,\mathrm{az}\intercal} \boldsymbol{A}^i \boldsymbol{y}^{i,\,(c_1,\,c_2),\,\mathrm{az}} \text{ for } c_1 \neq c_2 \tag{5.3}$$

Unlike the capability-oblivious setting where the column player's strategy is fixed for a fixed capability level, in Fig. 5.3, the column player's strategy is fixed when $c_1 > c_2$, and the row player's strategy is fixed when $c_1 < c_2$. The fully-capability-aware setting strengthens the importance of a higher capability level since a player only gets a positive mean payoff if they have a strictly higher capability level than their opponent. One interesting observation is that the payoff is not monotonically increasing with a player's capability level when this player has a lower capability level (i.e., each row in the lower triangle of the heatmap, or each column in the upper triangle of the heatmap).

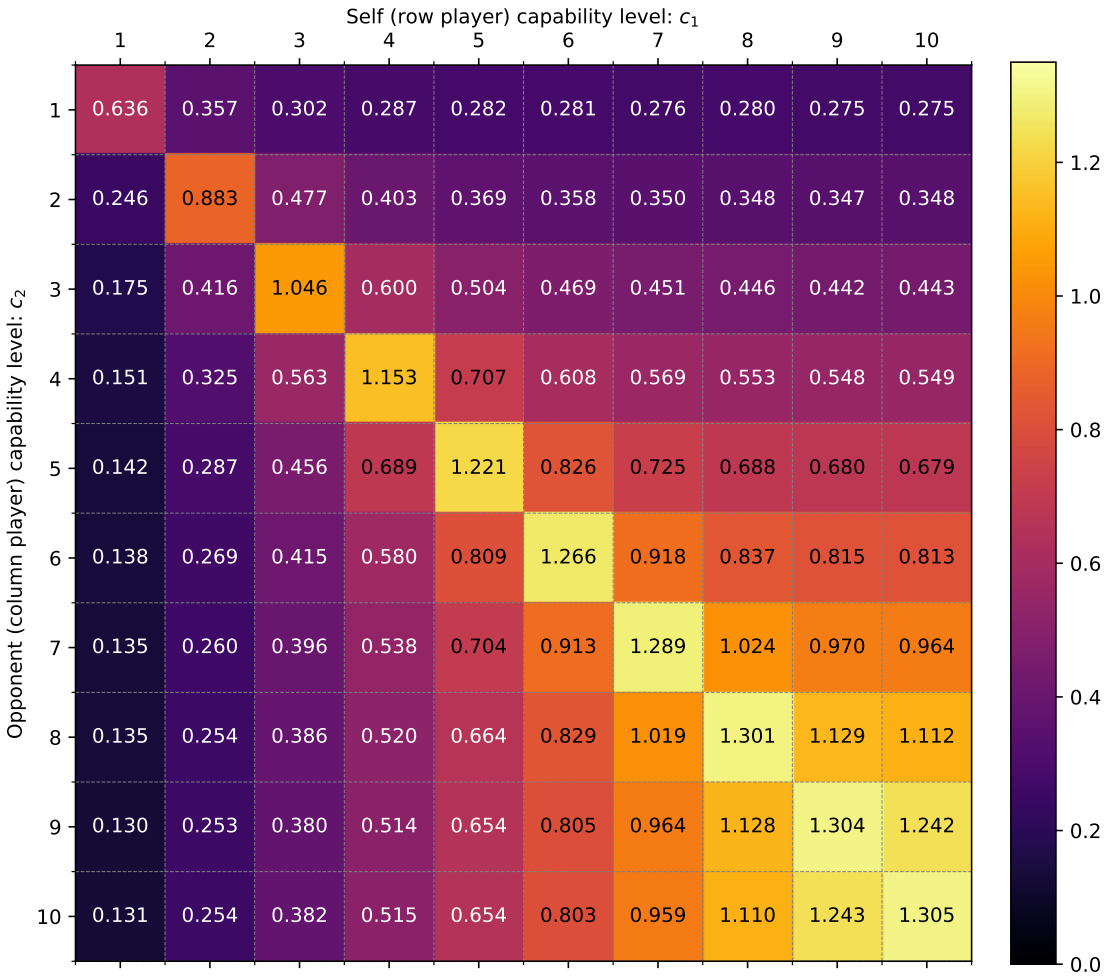


Figure 5.2: Heatmap of $\overline{\boldsymbol{P}}^{\text{og}}$, the row player's mean payoff of different capability levels in the capability-oblivious general-sum setting; defined in (5.2).

Since the lower-capability player uses the maximin strategy that is independent of their opponent's capability level, Fig. 5.3 suggests that the game is *pragmatically capability-positive by expectation* for the higher-capability player. Similar to Definition 4.25, a game is pragmatically capability-positive by expectation if the expected payoff of a player increases with their capability level while all other players stay at the same capability levels.

Again, individual games are not necessarily pragmatically capability-positive. Define the pattern violation rate of the game $(\boldsymbol{A}^i, \boldsymbol{B}^i)$ as the proportion of the pairs (c_1, c_2) such that $c_1 > c_2$ and $P^{i,\text{az}}_{c_2,c_1} > D^{i,\text{az}}_{c_2,c_1+1}$, or $c_1 < c_2$ and $P^{i,\text{az}}_{c_2,c_1} < D^{i,\text{az}}_{c_2+1,c_1}$. The median, minimum, and maximum pattern-violation rates over all games in T are 26.39%, 17.78%, and 54.17%, respectively.

Figure 5.4 visualizes $\overline{\boldsymbol{P}}^{\text{ag}}$, the row player's mean payoff over the games in T for different capability profiles in the fully-capability-aware general-sum setting:

$$\overline{\boldsymbol{P}}^{\text{ag}} := \frac{1}{|T|}\sum_{i\in T}\boldsymbol{P}^{i,\text{ag}}, \quad \text{where } P^{i,\text{ag}}_{c_2,c_1} := \boldsymbol{x}^{i,(c_1,c_2),\text{ag}\intercal}\boldsymbol{A}^i\boldsymbol{y}^{i,(c_1,c_2),\text{ag}} \text{ for } c_1 \neq c_2 \tag{5.4}$$

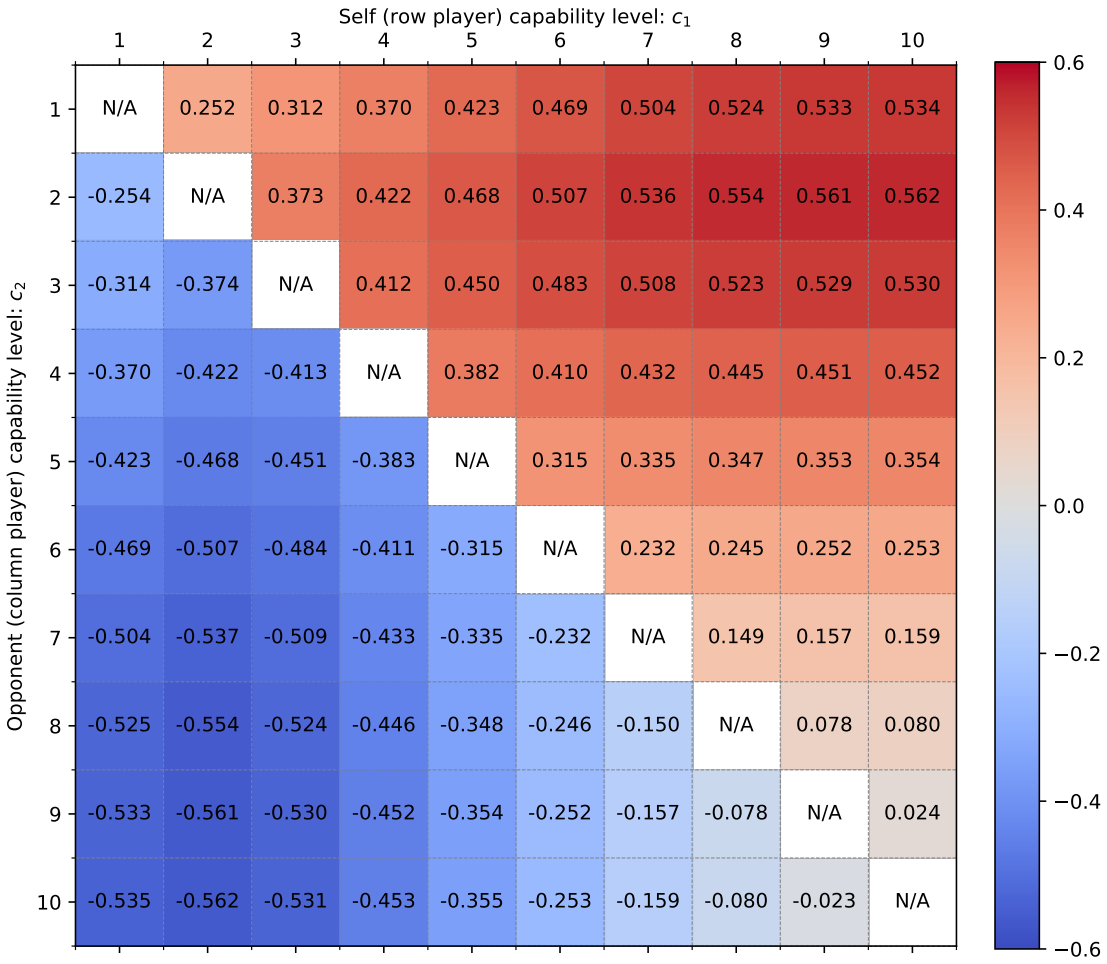


Figure 5.3: Heatmap of $\overline{\boldsymbol{P}}^{\mathrm{az}}$, the row player's mean payoff of different capability profiles in the fully-capability-aware zero-sum setting; defined in (5.3).

In contrast to the zero-sum case, the Fig. 5.4 suggests that the game is pragmatically capability-positive by expectation for any player, while the most significant payoff increase is observed when the player's capability level surpasses their opponent's capability level. In an individual game, let the pattern violation rate be the proportion of the pairs (c_1, c_2) such that $P^{i,\mathrm{ag}}_{c_2,c_1} > P^{i,\mathrm{ag}}_{c_2,c_1+1}$ for $c_1 + 1 < c_2$ or $c_1 > c_2$. The median, minimum, and maximum pattern-violation rates over all games in T are 37.50%, 1.39%, and 70.83%, respectively.

Figures 5.5 and 5.6 visualize $\overline{\boldsymbol{D}}^{\mathrm{az}}$ and $\overline{\boldsymbol{D}}^{\mathrm{ag}}$, the row player's mean payoff changes when the game transitions from the capability-oblivious setting to the fully-capability-aware setting in the zero-sum and general-sum cases, respectively. The matrices $\boldsymbol{D}^{i,\mathrm{az}}$ and $\boldsymbol{D}^{i,\mathrm{ag}}$ are defined as follows:

$$\overline{\boldsymbol{D}}^{\mathrm{az}} := \frac{1}{|T|}\sum_{i\in T}\boldsymbol{D}^{i,\mathrm{az}}, \quad \overline{\boldsymbol{D}}^{\mathrm{ag}} := \frac{1}{|T|}\sum_{i\in T}\boldsymbol{D}^{i,\mathrm{ag}} \tag{5.5}$$
$$\text{where } \boldsymbol{D}^{i,\mathrm{az}} := \boldsymbol{P}^{i,\mathrm{az}} - \boldsymbol{P}^{i,\mathrm{oz}}, \quad \boldsymbol{D}^{i,\mathrm{ag}} := \boldsymbol{P}^{i,\mathrm{ag}} - \boldsymbol{P}^{i,\mathrm{og}}$$

One can observe from Figs. 5.5 and 5.6 that the lower-capability player gets a worse mean

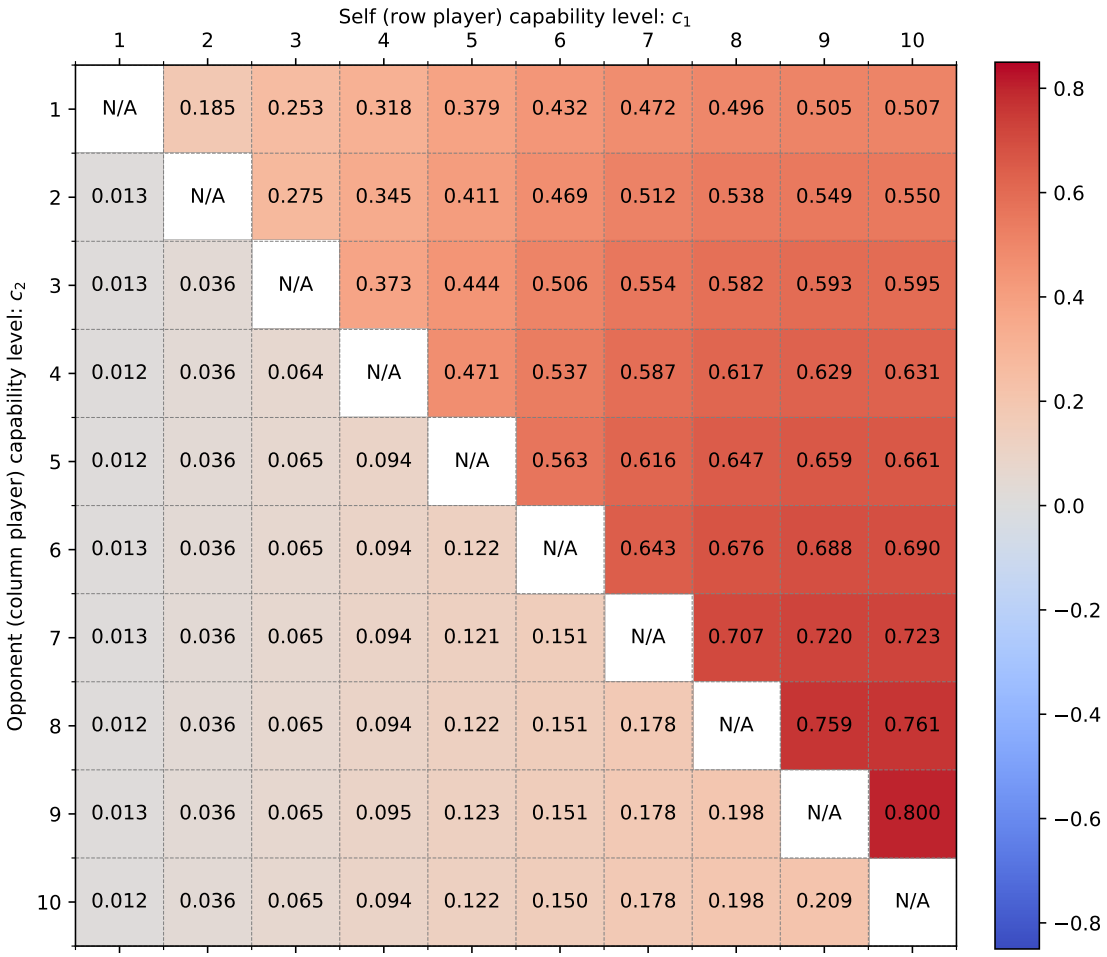


Figure 5.4: Heatmap of $\overline{\boldsymbol{P}}^{\text{ag}}$, the row player's mean payoff of different capability profiles in the fully-capability-aware general-sum setting; defined in (5.4).

payoff when both players become fully-capability-aware compared to the capability-oblivious setting. This is because the lower-capability player uses the maximin strategy in the fully-capability-aware setting, and their opponent anticipates their move. In the zero-sum case, the maximin strategy can be exploited by the best response of the higher-capability player; in the general-sum case, this maximin strategy is not as effective as the best response in the perceived game. As before, this pattern is not universally true for individual games. Define the pattern violation rate of the game $(\boldsymbol{A}^i, \boldsymbol{B}^i)$ as the proportion of the pairs (c_1, c_2) such that $D^{i,\text{az}}_{c_2,c_1} < 0$ or $D^{i,\text{ag}}_{c_2,c_1} < 0$ for the zero-sum and general-sum cases, respectively. The median, minimum, and maximum pattern-violation rates over all games in T are 4.44%, 0.00%, and 93.33% in the zero-sum case, and 33.33%, 0.00%, and 100.00% in the general-sum case, respectively.

It is worth noting that Fig. 5.6 exhibits another pattern: in general-sum games, when $c_2 \geq 5$, the row player always gets a worse average payoff in the fully-capability-aware setting compared to the capability-oblivious setting, no matter if the row player has a higher or lower capability level than the column player. Explaining this pattern is left as an open problem.

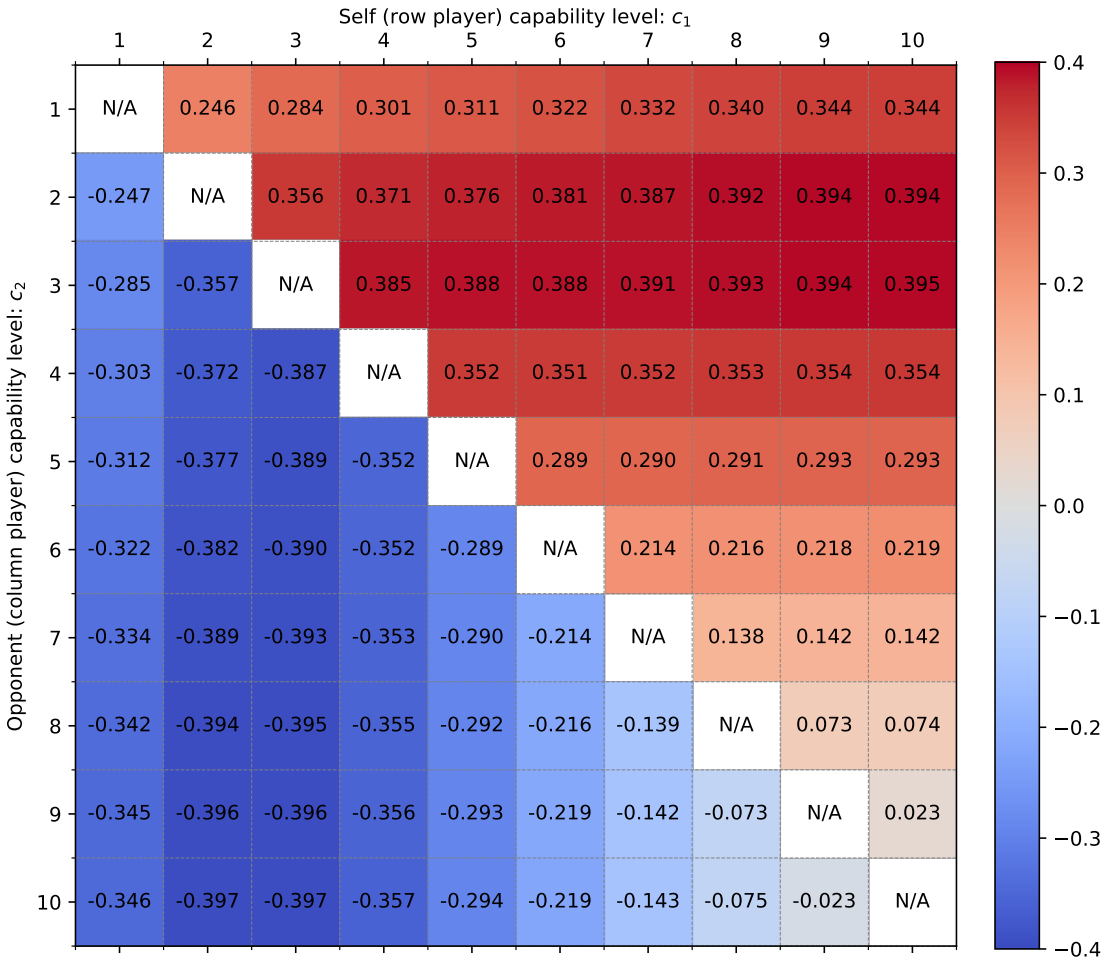


Figure 5.5: Heatmap of $\overline{\boldsymbol{D}}^{\mathrm{az}}$, the row player's mean payoff change when the game transitions from the capability-oblivious setting to the fully-capability-aware setting in the zero-sum case; defined in (5.5).

5.2 Kuhn poker

Kuhn poker is a simplified two-player poker with three distinct cards. At the beginning of the game, each player is privately dealt one card and antes one unit to the pot. Then a single round of betting occurs, where players take turns to check, bet, call, or fold. The first player starts with a check or a bet. The response to a check can be a check or a bet. The response to a bet can be a fold or a call. A player adds one unit to the pot when they bet or call. The game ends when both players check, one player folds, or one player calls. At the end of the game, if a player folds, the other player wins the pot; otherwise, a showdown occurs, and the player with the highest card wins the pot. The possible payoffs of a player are $[\![-2, 2]\!]$.

Despite its simplicity, Kuhn poker exhibits complex strategic interactions in the equilibrium solutions, including deceptive behaviors of bluffing (betting with a weak hand) and underbidding (passing with a strong hand) [Kuh50].

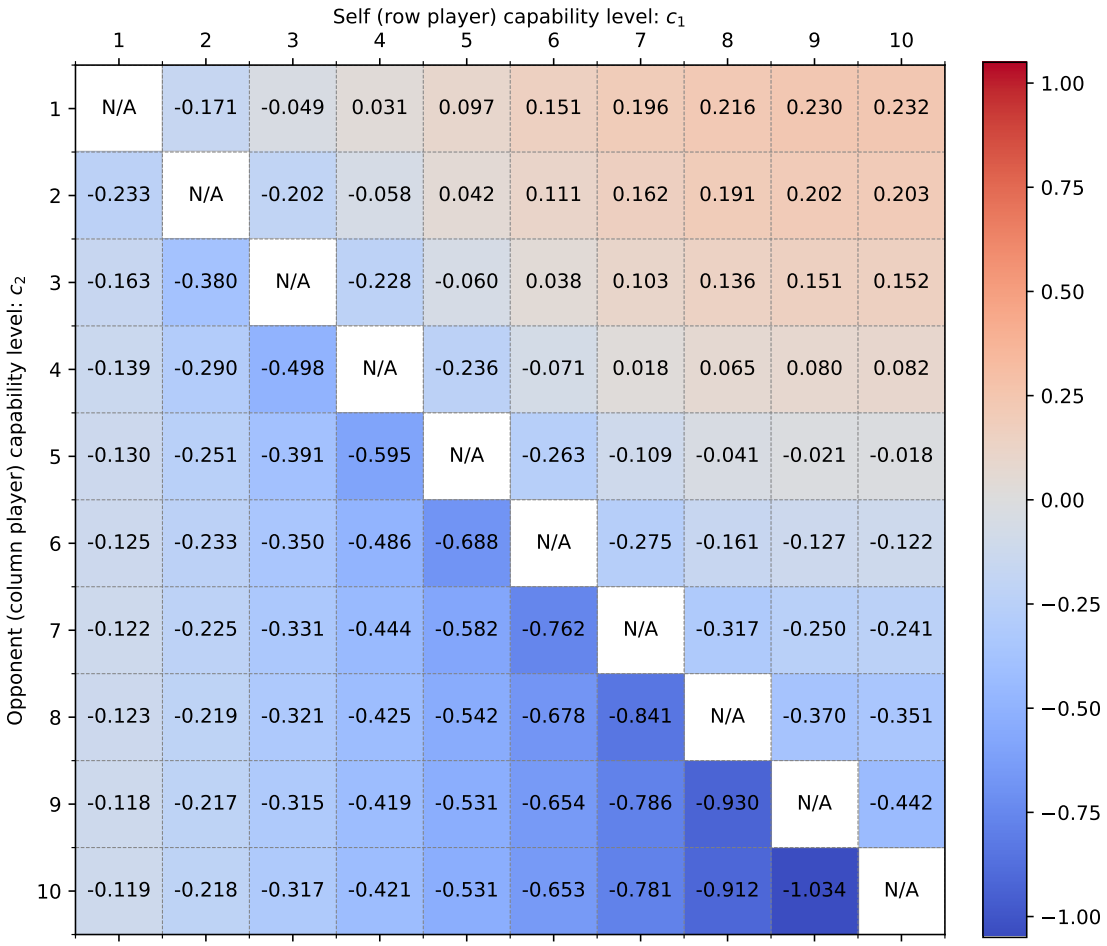


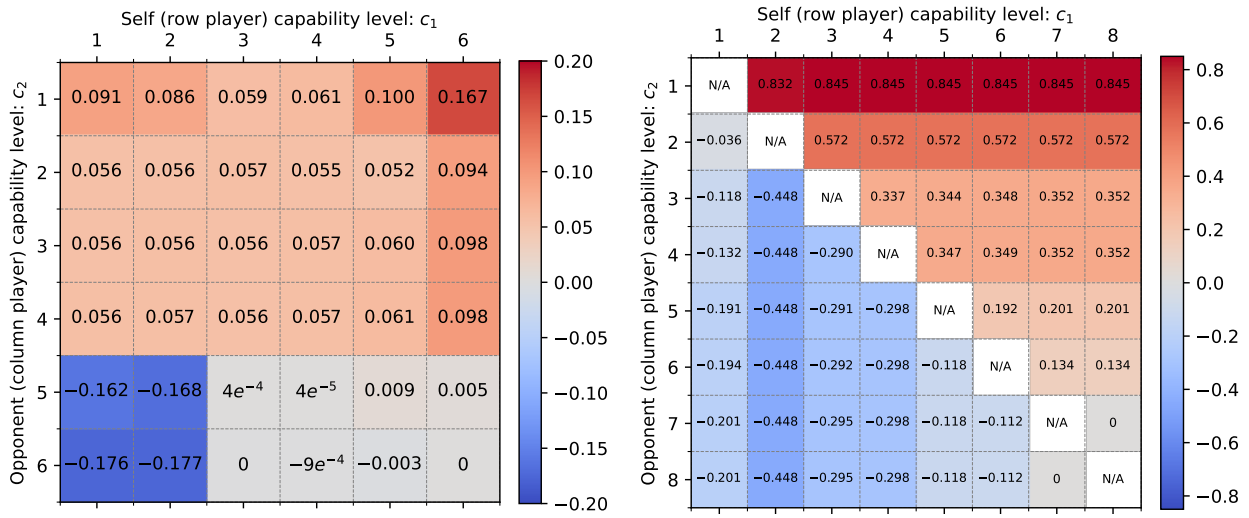
Figure 5.6: Heatmap of $\overline{\boldsymbol{D}}^{\text{ag}}$, the row player's mean payoff change when the game transitions from the capability-oblivious setting to the fully-capability-aware setting in the general-sum case; defined in (5.5).

Kuhn poker is an incomplete-information extensive-form zero-sum game. It can be converted into a normal-form game by considering all pure strategy profiles (s_1, s_2), where $s_i \in \mathcal{S}_i$ encodes the actions to be taken at each information set on the game tree for player i. The normal-form game has $|\mathcal{S}_1| = 27$ and $|\mathcal{S}_2| = 64$ pure strategies for each player. The payoff of (s_1, s_2) is the expected payoff of the game (i.e., payoffs averaged over $3! = 6$ possible hands) when player 1 acts according to s_1 and player 2 acts according to s_2. See Farina [Far23] for a more detailed description of Kuhn poker, its extensive-form representation, and its conversion to the normal-form game.

The payoff matrix of Kuhn poker is denoted as $\boldsymbol{A}^{\text{Kuhn}} \in \mathbb{R}^{27\times 64}$. Kuhn poker's value is $v_1 = -1/18$ for the first player. The rank of $\boldsymbol{A}^{\text{Kuhn}}$ is 6, with the following singular values:

$$\sigma_1 = 14.69 \quad \sigma_2 = 10.98 \quad \sigma_3 = 4.61 \quad \sigma_4 = 4.01 \quad \sigma_5 = 1.91 \quad \sigma_6 = 1.23 \quad \sigma_{\geq 7} = 0$$

The low-rank structure of Kuhn poker is unsurprising. Kuhn [Kuh50] shows that after heuristic strategy elimination in Kuhn poker, there are 8×4 strategy profiles left, and the



(a) Capability-oblivious setting

(b) Fully-capability-aware setting

Figure 5.7: Payoff changes of the first player in Kuhn poker.
Formally, the two heatmaps visualize $\boldsymbol{P}^{\text{Kuhn, oz}} - v_1$ and $\boldsymbol{P}^{\text{Kuhn, az}} - v_1$, respectively, with $\boldsymbol{P}^{\text{Kuhn, oz}}$ and $\boldsymbol{P}^{\text{Kuhn, az}}$ defined as in (5.1) and (5.3), and $v_1 = -1/18$ being the value of the first player in the full-rank game. A number in the form of ae^b denotes $a \times 10^b$.

remaining payoff matrix has rank 3. Farina and Sandholm [FS22] shows that many poker game variants have an inherent low-rank structure.

Since $\operatorname{rank}\left(\boldsymbol{A}^{\text{Kuhn}}\right) = 6$, having a capability level of 6 is sufficient to perceive the true game in the capability-oblivious setting. In the fully-capability-aware setting, a rank-6 player can not ensure that they perceive the true game since $\sigma_6\left(\boldsymbol{A}^{\text{Kuhn}}\right) > 0$; however, a rank-7 player reasons that their perceived game is the true game.

Figure 5.7 visualizes the change of true payoffs (compared to v_1) for the first player for all nontrivial capability profiles in both the capability-oblivious and fully-capability-aware settings. The game is capability-positive for the higher-capability player or the column player in the fully-capability-aware setting. In particular, having $c_1 = 2$ is worse than $c_1 = 1$ when $c_2 \geq 3$. Interestingly, the capability-oblivious setting exhibits a "rank-3 preference" for the first player, i.e., having a capability level of 3 gives the row player nonnegative payoff changes for all capability profiles. Additionally, when the column player has full capability, $\boldsymbol{x}^{\text{Kuhn}, 3, \text{oz}}$ is a best response to the column player's strategy, which is also strictly better than $\boldsymbol{x}^{\text{Kuhn}, 4, \text{oz}}$ or $\boldsymbol{x}^{\text{Kuhn}, 5, \text{oz}}$. To validate whether the rank-3 preference is robust, four games are sampled by perturbing the payoff matrix $\boldsymbol{A}^{\text{Kuhn}}$ with an additive Gaussian noise with standard deviation 10^{-3}; player's strategies are computed for the perceptions of the perturbed games. The resulting capability-oblivious payoff changes are visualized in Fig. 5.8. This rank-3 preference against a rank-6 opponent is still observed in the perturbed games.

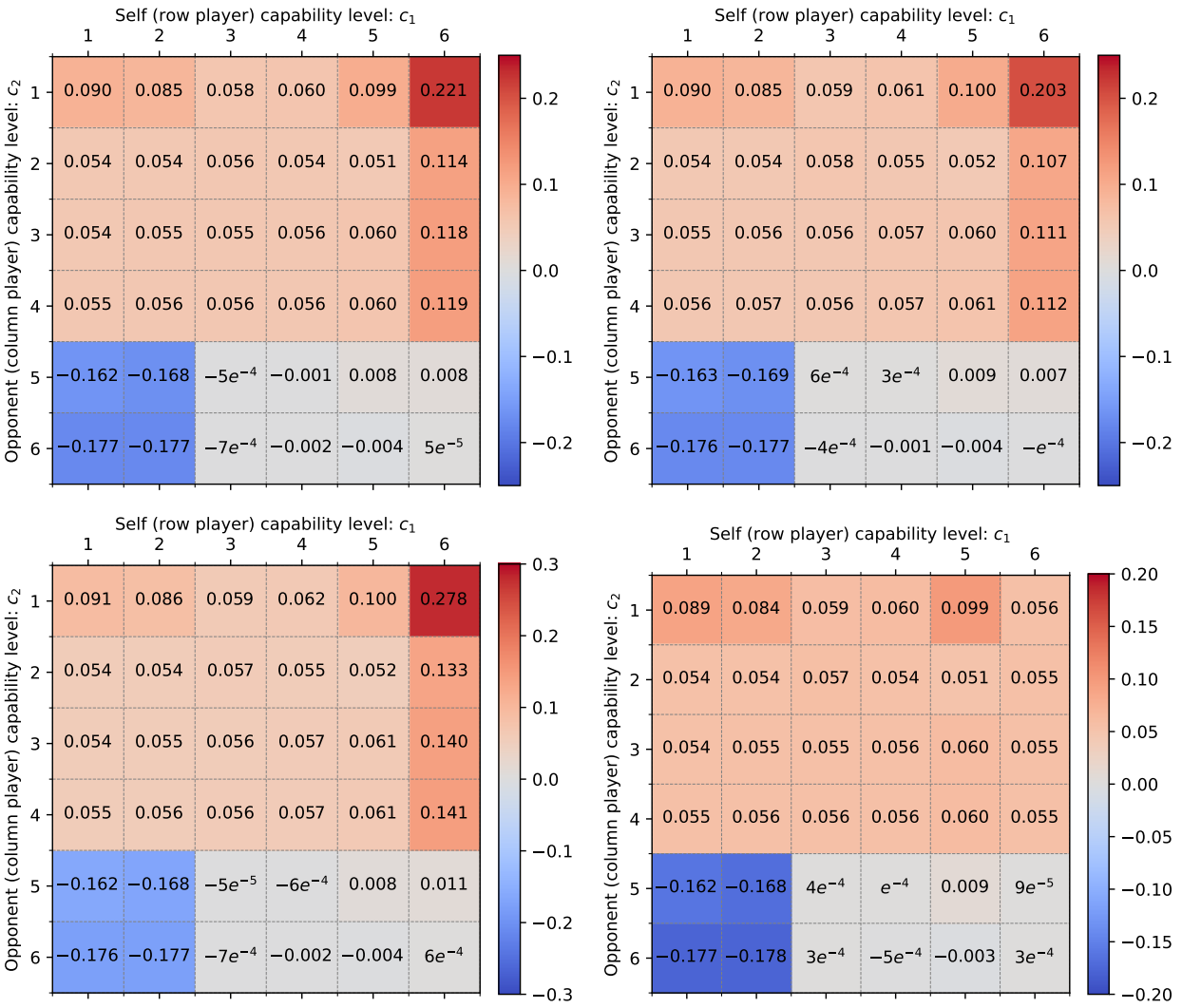


Figure 5.8: Payoff changes of the first player in perturbed Kuhn poker games in the capability-oblivious setting.

The matrices are generated similarly to Fig. 5.7a, except that the true payoff matrix $\boldsymbol{A}^{\text{Kuhn}}$ is perturbed by an additive Gaussian noise with standard deviation 10^{-3} (with new player strategies computed accordingly). Four independent samples are shown in this figure.

5.3 Summary and discussion

This chapter makes the following empirical observations for random games:

- In the capability-oblivious setting, if the game is totally competitive (i.e., zero-sum), then a player should increase their capability level as much as possible to get a better expected payoff. However, if the game is general-sum where players have independent payoff matrices, then a player should match the capability level of their opponent as close as possible to get a better expected payoff; players get better payoffs when they increase their matched capability levels simultaneously.

  As a comparison, for the general-sum 2×2 parameterized defense game analyzed in Section 3.4.2, Theorem 3.16 shows that for all game parameters, the attacker benefits from matching the defender's capability level, while the defender benefits from having a higher capability level.

- The zero-sum fully-capability-aware game is pragmatically capability-positive by expectation for the higher-capability player. The general-sum fully-capability-aware game is pragmatically capability-positive by expectation for any player.

- In both the zero-sum and general-sum cases, transitioning from the capability-oblivious setting to the fully-capability-aware setting reduces the expected payoff of the lower-capability player. In the zero-sum setting, this transition improves the expected payoff of the higher-capability player. In the general-sum setting, this transition can result in a better or worse expected payoff for the higher-capability player depending on the capability profile of the players, which means that both players could suffer from the transition.

- The above observations apply to the expected payoff over games sampled from normal distributions. For each of the observations, there are individual games that violate the pattern described in the observation.

For Kuhn poker, the following observations are made:

- In the fully-capability-aware setting, the game is capability-positive for the higher-capability player or the column player (i.e., the second player). If the row player has lower capability than the column player, then having a capability level of 2 for the row player gives them the worst payoff compared to other capability levels.

- In the capability-oblivious setting, the game is not capability-positive (nor capability-negative) for any player.

- In the capability-oblivious setting, if a player wants to have almost no worse payoff than the optimal strategy on the true game, the row player should have $c_1 = 3$ or $c_1 \geq 6$, and the column player should have $c_2 \geq 5$. This observation is robust to perturbations of the true payoff matrix.

Chapter 6

Related work for limited-perception games

Bayesian games Limited-perception games can be seen as an instance of incomplete information games. One classic model of incomplete information game is the Bayesian game [Har67], where each player has a type, and a player's payoff depends not only on all players' strategy profile but also on the player's type. In a typical Bayesian game, each player knows their type as private information, and the prior distribution of all players' types is common knowledge. At an equilibrium, each player maximizes their expected payoff. There are many extensions and applications of Bayesian games, such as mechanism design [HL10], network security [Che+21], and robust optimization [Gro+13]. By comparison, the limited-perception game model differs from the Bayesian game model in three key aspects:

- Limited-perception games take a distribution-free approach. Capability-oblivious players optimize their apparent payoff, while capability-aware players optimize their worst-case payoff.
- In a Bayesian game, a player knows their own type as private information, which completely determines their own payoff function. By contrast, in a limited-perception game, a player at most knows their own capability level, which only partially determines their payoff function.
- The limited-perception game models a hierarchy of players' perception capabilities, which is not well captured by assigning a single type to each player as in a Bayesian game.

Robust games Aghassi and Bertsimas [AB06] studies a distribution-free model of incomplete information games that they call robust games. The robust game model is closely related to the limited-perception game model. In a robust game, there is a set of possible payoff functions $\mathcal{U}$. Aghassi and Bertsimas [AB06] defines two types of equilibria:

- In an *equilibrium* of the robust game, the strategy profile $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ satisfies

$$\forall i \in \mathcal{N}: \quad \boldsymbol{x}_i \in \operatorname*{argmax}_{\boldsymbol{x}_i' \in \Delta_{\mathcal{S}_i}} \inf_{(u_1, \ldots, u_n) \in \mathcal{U}} u_i(\boldsymbol{x}_i', \boldsymbol{x}_{-i})$$

- In an *ex post equilibrium* of the robust game, the strategy profile $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ satisfies

$$\forall (u_1, \ldots, u_n) \in \mathcal{U} : \forall i \in \mathcal{N} : \quad \boldsymbol{x}_i \in \operatorname*{argmax}_{\boldsymbol{x}_i' \in \Delta_{\mathcal{S}_i}} u_i(\boldsymbol{x}_i', \boldsymbol{x}_{-i})$$

There are a few important differences between robust games and limited-perception games:

- A player's perception in the robust game can be viewed as a one-level abstraction of the true payoff functions. In contrast, limited-perception games model the perception as a multi-level hierarchy of abstractions.
- In a robust game, the abstraction is defined over the Cartesian product of the players' payoff functions. In contrast, in a limited-perception game, the abstraction is defined over each player's payoff function. All players share the abstraction process (not the abstraction result) in a limited-perception game.
- Aghassi and Bertsimas [AB06] proves the existence of an equilibrium in any robust game and provides examples where the ex post equilibrium does not exist. They do not specify how the players receive their final payoffs. In contrast, in a limited-perception game, there is a true payoff function that determines a player's final payoff. The Nash equilibrium in a limited-perception game is conceptually more similar to the ex post equilibrium in a robust game since the equilibrium strategy must be a best response to all possible true payoff functions. In the limited-perception game, the Nash equilibrium is guaranteed to exist (Theorem 4.5) since our formulation uses higher-order functions to encode all possible opponents' perceptions instead of the mixed strategies used in robust games.

Stackelberg games Another related research topic is Stackelberg games with uncertainty. One particular application is in security games, where a defender commits to a strategy before the attacker's move in the presence of the defender's uncertainty about the attacker's payoff function [BGA+09; Con12]. Prior research has considered settings where the uncertainty is modeled by a Bayesian prior distribution [Par+08; KTM10] and settings where the uncertainty is modeled by intervals of possible true payoffs [KIK13; Ngu+14; KFS18]. By comparison, the current research models a large class of distribution-free hierarchical uncertainties that include interval uncertainty as a special case; our model also considers uncertainties of the leader's payoff functions.

An interesting property of conventional Stackelberg games is that the equilibrium is guaranteed to exist only when the follower breaks ties in favor of the leader [CS06]. However, in the limited-perception game model, an exact equilibrium does not always exist in the general-sum case with an infinite narrow concretization set due to discontinuities in the leader's objective function even if we assume some tie-breaking rule (Remark 4.6). By comparison, a zero-sum limited-perception Stackelberg game has similar properties as conventional Stackelberg games, where the equilibrium exists if the follower breaks ties in favor of the leader (Theorem 4.9).

Bounded rationality The analysis of capability-oblivious players in limited-perception games is related to the study on nonrational or nonoptimal behavior in games, which can be broadly described as bounded rationality. The concept of bounded rationality was coined by Simon [Sim69] to characterize human decision-making processes under cognitive constraints. Prior efforts to formalize bounded rationality include bounding the complexity of the decision-making automaton in repeated games [Ney85; AR88], bounding the randomness available to the players in implementing mixed strategies [NO00; VG19; OR23], modeling the cost of Turing machines that implement strategies [HP15], modeling players who may (mistakenly) deviate from their strategies with negligible probability [SB88; Far+18], and analyzing games with cost functions associated with strategies [BKK06]. Unlike the current research, prior work on bounded rationality neither explicitly models a hierarchy of player capabilities nor studies the capability transfer properties based on a hierarchy of player capabilities.

Learning in games Another extensively studied setting with uncertain payoffs is learning in repeated games, where repeated interaction with the environment or other agents allows an agent to accumulate information about their payoff function. Popular frameworks for learning include no-regret learning [DFG21] and reinforcement learning [NVD12]. By comparison, the current research focuses on one-shot games where the uncertainty in the payoff is caused by perception limitations, without any further interaction with other agents in the game play. Incorporating learning dynamics into the limited-perception game model is an interesting direction for future research.

Special cases of limited-perception games Zwillinger and Clemente [ZC23] studies a special case of limited-perception games in detail. They consider the setting where the row player's perception of their opponent's payoff matrix may be corrupted by swapping two values. They enumerate all ordinal 2×2 bimatrix games (i.e., games whose entries of the payoff matrices form a permutation of $[\![1, 4]\!]$) to compute the row player's mean payoff change under different epistemic assumptions. However, their investigation does not go beyond reporting the empirical results on the 2×2 games. By contrast, the current research provides a general framework for analyzing limited-perception games and derives plentiful theoretical results besides presenting empirical observations on numerical experiments.

Matrix rank in game theory research This research uses the limited-rank game as a prominent example of limited-perception games. Researchers have also explored ranks in bimatrix games. Most previous research takes a perspective of computational complexity. For example, Lipton, Markakis, and Mehta [LMM03] shows that low-rank bimatrix games have a small-support exact equilibrium; Mehta [Meh14] studies the hardness of bimatrix games with a constant rank where the rank of a game is defined by the rank of the sum of the two payoff matrices. Other efforts exploit low-rank structures to scale up the computation of Nash strategy via linear programming in large zero-sum games [ZS20; FS22]. To the best of the author's knowledge, no prior research studies the setting of one-shot games where players perceive limited-rank approximations of the payoff matrices with the final outcome determined by hidden true payoff matrices.

Abstraction Reasoning with abstraction is a fundamental concept in computer science. One of the well-known applications of this concept is abstract interpretation in program analysis. Abstract interpretation uses an abstract value to represent a set of concrete values in the reasoning of program properties [Cou21]. The abstraction functions in program analysis are typically single-level and order-preserving on some lattice so that the existence of fixed points in loops or recursion can be guaranteed by the Knaster-Tarski theorem [Tar55]. By comparison, the abstraction functions in limited-perception games are bounded and hierarchical to model multiple levels of player capabilities.

Chapter 7

Efficient nonsmooth convex optimization

The analysis of efficiently solvable fully-capability-aware zero-sum games in Section 4.3.3 reduces equilibrium solving to solving the maximin strategy. Computing a maximin strategy can be formalized as a nonsmooth convex optimization problem (Theorem 4.19). As introduced in Section 1.3, nonsmooth convex optimization is a fundamental problem with applications in various fields.

This chapter presents a new optimization algorithm, the *Trust Region Adversarial Functional Subdifferential* (TRAFS), for unstructured nonsmooth convex optimization. Section 7.1 defines a new type of subdifferential, the functional subdifferential, that enables efficient computation while capturing enough local information to support effective optimization. Section 7.2 presents and analyzes the TRAFS algorithm that uses the functional

subdifferential for nonsmooth optimization. Section 7.3 presents numerical experiment results that demonstrate the effectiveness of the TRAFS algorithm. Section 7.4 discusses related work.

7.1 The functional subdifferential

7.1.1 Introducing the functional subdifferential

In this section, we first recall the definitions of subgradient and subdifferential. We then define the *relaxed subdifferential* as a natural idea to include the subdifferential of local nonsmooth points. We define a class of nonsmooth functions called the *sparsely nonsmooth* functions. We show that the relaxed subdifferential provides a local upper bound and global lower bound for sparsely nonsmooth functions. Finally, we generalize the definition of the relaxed subdifferential to the *functional subdifferential* by only requiring the same upper and lower bounds to be met.

Definition 7.1 (Subdifferential [Bec17, Definitions 3.1, 3.2])
Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be a convex function. A vector $\boldsymbol{g} \in \mathbb{R}^n$ is a *subgradient* of $f(\cdot)$ at $\boldsymbol{x} \in \mathbb{R}^n$ if and only if $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{g}^\intercal(\boldsymbol{y} - \boldsymbol{x})$ holds for all $\boldsymbol{y} \in \mathbb{R}^n$. The set of all subgradients of $f(\cdot)$ at $\boldsymbol{x}$ is called the *subdifferential* of $f(\cdot)$ at $\boldsymbol{x}$ and is denoted by $\partial f(\boldsymbol{x})$:

$$\partial f(\boldsymbol{x}) \coloneqq \{\boldsymbol{g} \in \mathbb{R}^n \mid \forall \boldsymbol{y} \in \mathbb{R}^n : \; f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{g}^\intercal(\boldsymbol{y} - \boldsymbol{x})\} \tag{7.1}$$

◁

Proposition 7.1
The subdifferential of a convex function $f : \mathbb{R}^n \mapsto \mathbb{R}$ satisfies a few properties [Bec17, Chapter 3]:

- *For all $\boldsymbol{x} \in \mathbb{R}^n$, $\partial f(\boldsymbol{x})$ is nonempty, convex, closed, and bounded.*
- *If $f(\cdot)$ is differentiable at $\boldsymbol{x}$, then $\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$.*
- *For a convex set $C \subset \mathbb{R}^n$, $\boldsymbol{x}^* \in \operatorname{argmin}^*_{\boldsymbol{x} \in C} f(\boldsymbol{x})$ if and only if there exists $\boldsymbol{g} \in \partial f(\boldsymbol{x}^*)$ such that $\forall \boldsymbol{y} \in C : \; \boldsymbol{g}^\intercal(\boldsymbol{y} - \boldsymbol{x}^*) \geq 0$.*
- *If $f(\cdot)$ is L-Lipschitz over an open set S, then*

$$\forall \boldsymbol{x} \in S : \; \forall \boldsymbol{g} \in \partial f(\boldsymbol{x}) : \; \|\boldsymbol{g}\|_2 \leq L \tag{7.2}$$

Definition 7.2 (Relaxed subdifferential)
For a convex function $f : \mathbb{R}^n \mapsto \mathbb{R}$, a set $S \subset \mathbb{R}^n$, and $\epsilon \in \mathbb{R}_+$, its ϵ-relaxed subdifferential at $\boldsymbol{x} \in \mathbb{R}^n$ constrained to S is defined as

$$\partial^{\mathrm{r}}_\epsilon f(\boldsymbol{x};\, S) \coloneqq \operatorname{cl}\operatorname{conv}\{\boldsymbol{g} \mid \boldsymbol{g} \in \partial f(\boldsymbol{y}) \text{ for } \boldsymbol{y} = \boldsymbol{x} \text{ or } (\boldsymbol{y} \in S \cap B_\epsilon[\boldsymbol{x}] \text{ and } |\partial f(\boldsymbol{y})| > 1)\}$$

◁

Remark 7.1 The relaxed subdifferential differs from the ϵ-subdifferential (defined as $\partial_\epsilon^{\mathrm{C}} f(\boldsymbol{x}) \coloneqq \{\boldsymbol{g} \in \mathbb{R}^n \mid \forall \boldsymbol{y} \in \mathbb{R}^n : \ f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{g}^\intercal(\boldsymbol{y} - \boldsymbol{x}) - \epsilon\}$, cf. [BKM14, Definition 2.17]) and the Goldstein ϵ-subdifferential ($\partial_\epsilon^{\mathrm{G}} f(\boldsymbol{x}) \coloneqq \operatorname{cl}\operatorname{conv} \bigcup_{\boldsymbol{y} \in B_\epsilon[\boldsymbol{x}]} \partial f(\boldsymbol{y})$, cf. [BKM14, Definition 3.3]) in that the relaxed subdifferential only considers the local nondifferentiable points can can be easier to compute.

Now we introduce sparse nonsmoothness and related properties. Informally, a sparsely nonsmooth function has finitely separated nondifferentiable points along any direction, and the gradient over differentiable intervals is Lipschitz continuous.

Definition 7.3 (Sparse nonsmoothness)
Given a function $f : \mathbb{R}^n \mapsto \mathbb{R}$ and a convex set $S \subset \mathbb{R}^n$, $f(\cdot)$ is *β-sparsely nonsmooth* over S if and only if:

For all $(\boldsymbol{x}_0, \boldsymbol{d}) \in S \times \mathbb{R}^n$ such that $\|\boldsymbol{d}\|_2 = 1$ and $\boldsymbol{x}_0 + k\boldsymbol{d} \in S$ for some $k \neq 0$, define $g(\lambda) \coloneqq f(\boldsymbol{x}_0 + \lambda \boldsymbol{d})$; define $E \coloneqq \{\lambda \in \mathbb{R} \mid \boldsymbol{x}_0 + \lambda \boldsymbol{d} \in S\}$ as the interval of valid values of λ regarding S. Then there exists a (possibly empty or infinite) sequence $(c_1, \ldots, c_m)$ in E such that $\left(\inf_{i \in [\![1, m-1]\!]} c_{i+1} - c_i\right) > 0$, $g(\cdot)$ is not differentiable at c_i for $i \in [\![1, m]\!]$, and $g(\cdot)$ is β-smooth over all intervals (c_i, c_{i+1}) for $i \in [\![0, m]\!]$, where $c_0 \coloneqq \inf E$ and $c_{m+1} \coloneqq \sup E$, i.e., for all $i \in [\![0, m]\!]$, for all $(\lambda_1, \lambda_2) \in (c_i, c_{i+1})^2$, we have $|g'(\lambda_2) - g'(\lambda_1)| \leq \beta|\lambda_2 - \lambda_1|$. ◁

Proposition 7.2
If a function $f(\cdot)$ is β-smooth over a convex set S, then it is also β-sparsely nonsmooth over S.

Proof. Given $\boldsymbol{x}_0$ and $\boldsymbol{d}$ as in Definition 7.3, we have $g'(\lambda) = \boldsymbol{d}^\intercal \nabla f(\boldsymbol{x}_0 + \lambda \boldsymbol{d})$. Therefore,

$$\begin{aligned} |g'(\lambda_2) - g'(\lambda_1)| &\leq \|\boldsymbol{d}\|_2 \|\nabla f(\boldsymbol{x}_0 + \lambda_2 \boldsymbol{d}) - \nabla f(\boldsymbol{x}_0 + \lambda_1 \boldsymbol{d})\|_2 \\ &\leq \beta \|(\boldsymbol{x}_0 + \lambda_2 \boldsymbol{d}) - (\boldsymbol{x}_0 + \lambda_1 \boldsymbol{d})\|_2 = \beta|\lambda_2 - \lambda_1| \end{aligned}$$

□

The relaxed subdifferential is related to the sparse nonsmoothness as the following:

Lemma 7.3
Given $\epsilon \in \mathbb{R}_+$, a convex function $f : \mathbb{R}^n \mapsto \mathbb{R}$, and an open convex set $S \subset \mathbb{R}^n$, if $f(\cdot)$ is L-Lipschitz over S, then for any $(\boldsymbol{x}, \boldsymbol{y}) \in S^2$,

$$f(\boldsymbol{y}) \geq \left(f(\boldsymbol{x}) + \max_{\boldsymbol{g} \in \partial_\epsilon^{\mathrm{r}} f(\boldsymbol{x}; S)} \boldsymbol{g}^\intercal (\boldsymbol{y} - \boldsymbol{x}) \right) - 2\epsilon L \tag{7.3}$$

Moreover, if $f(\cdot)$ is also β-sparsely nonsmooth over S and $\|\boldsymbol{y} - \boldsymbol{x}\|_2 \leq \epsilon$, then

$$f(\boldsymbol{y}) \leq \left(f(\boldsymbol{x}) + \max_{\boldsymbol{g} \in \partial_\epsilon^{\mathrm{r}} f(\boldsymbol{x}; S)} \boldsymbol{g}^\intercal (\boldsymbol{y} - \boldsymbol{x}) \right) + \frac{\beta}{2} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \tag{7.4}$$

Proof. A proof of (7.3) can found at Bagirov, Karmitsa, and Mäkelä [BKM14, Theorem 3.12].

Before proving (7.4), let's recall the definition of directional derivative: for a convex function $h : \mathbb{R}^m \mapsto \mathbb{R}$ and a vector $\boldsymbol{d} \in \mathbb{R}^m$, the directional derivative of $h(\cdot)$ at $\boldsymbol{x} \in \mathbb{R}^m$ in the direction $\boldsymbol{d}$ is defined as $h'(\boldsymbol{x}; \boldsymbol{d}) \coloneqq \lim_{\alpha \to 0^+} \frac{h(\boldsymbol{x} + \alpha \boldsymbol{d}) - h(\boldsymbol{x})}{\alpha}$. It is related to the subdifferential

through $\max_{\boldsymbol{g}\in\partial h(\boldsymbol{x})} \boldsymbol{g}^\intercal \boldsymbol{d} = h'(\boldsymbol{x};\boldsymbol{d})$ [Bec17, Theorem 3.26]. When the dimension $m = 1$, we denote $h'_+(x) \coloneqq h'(x;\,1)$ and $h'_-(x) \coloneqq h'(x;\,-1)$. If $m = 1$ and $h(\cdot)$ is differentiable over the open interval $(x,\,y)$, then $h'_+(x) = \lim_{\alpha\to x^+} h'(\alpha)$ and $h'_-(y) = \lim_{\alpha\to y^-} h'(\alpha)$ [Roc70, Theorem 24.1].

Now let's assume $f(\cdot)$ is β-sparsely nonsmooth over S and $\|\boldsymbol{y}-\boldsymbol{x}\|_2 \le \epsilon$. Let $r \coloneqq \|\boldsymbol{y}-\boldsymbol{x}\|_2$ and $\boldsymbol{d} \coloneqq (\boldsymbol{y}-\boldsymbol{x})/r$. Define $g(\lambda) \coloneqq f(\boldsymbol{x}+\lambda\boldsymbol{d})$. Due to the definition of sparse nonsmoothness, there is a finite sequence $(c_0 \coloneqq 0,\, c_1,\, \ldots,\, c_m,\, c_{m+1} = r)$ in $[0,\,r]$ such that $g(\cdot)$ is β-smooth over $(c_i,\,c_{i+1})$ for $i \in [\![0,\,m]\!]$ and $g(\cdot)$ is not differentiable at c_i for $i \in [\![1,\,m]\!]$. Thus we have $\partial f(\boldsymbol{x}+c_i\boldsymbol{d}) \subset \partial^{\mathrm{r}}_\epsilon f(\boldsymbol{x};\,S)$ for $i \in [\![0,\,m]\!]$.

For each $i \in [\![0,\,m]\!]$, we define a β-smooth function $g_i(\lambda)$ over $[c_i,\,c_{i+1}]$ such that $g_i'(\lambda) = g'(\lambda)$ for $c_i < \lambda < c_{i+1}$, $g_i'(c_i) = g'_+(c_i)$, $g_i'(c_{i+1}) = g'_-(c_{i+1})$, and $g_i(c_i) = g(c_i)$. It can be easily shown $g_i(\lambda) = g(\lambda)$ for $c_i \le \lambda \le c_{i+1}$.

Let $M \coloneqq \max_{\boldsymbol{g}\in\partial^{\mathrm{r}}_\epsilon f(\boldsymbol{x};\,S)} \boldsymbol{g}^\intercal(\boldsymbol{y}-\boldsymbol{x})$. Let $s_i \coloneqq c_{i+1}-c_i$. For $i \in [\![0,\,m]\!]$, we have $g_i'(c_i) = g'_+(c_i) = f'(\boldsymbol{x}+c_i\boldsymbol{d};\boldsymbol{d}) = \max_{\boldsymbol{g}\in\partial f(\boldsymbol{x}+c_i\boldsymbol{d})} \boldsymbol{g}^\intercal\boldsymbol{d} \le \max_{\boldsymbol{g}\in\partial^{\mathrm{r}}_\epsilon f(\boldsymbol{x};\,S)} \boldsymbol{g}^\intercal\boldsymbol{d} = \frac{M}{r}$. The β-smoothness of $g_i(\cdot)$ implies that $g_i(c_{i+1}) - g_i(c_i) \le g_i'(c_i)(c_{i+1}-c_i) + \frac{\beta}{2}(c_{i+1}-c_i)^2 \le \frac{M}{r}s_i + \frac{\beta}{2}s_i^2$. Therefore,

$$
\begin{aligned}
f(\boldsymbol{y}) - f(\boldsymbol{x}) = g(r) - g(0) &= \sum_{i=0}^{m} \left(g_i(c_{i+1}) - g_i(c_i)\right) \le \sum_{i=0}^{m} \left(\frac{M}{r}s_i + \frac{\beta}{2}s_i^2\right) \\
&\le \frac{M}{r}\left(\sum_{i=0}^{M} s_i\right) + \frac{\beta}{2}\left(\sum_{i=0}^{m} s_i\right)^2 = \max_{\boldsymbol{g}\in\partial^{\mathrm{r}}_\epsilon f(\boldsymbol{x};\,S)} \boldsymbol{g}^\intercal(\boldsymbol{y}-\boldsymbol{x}) + \frac{\beta}{2}\|\boldsymbol{y}-\boldsymbol{x}\|_2^2 \qquad \square
\end{aligned}
$$

Eqs. (7.3) and (7.4) are the core properties that would enable the convergence rates of the proposed TRAFS algorithm. One crucial observation is that the relaxed subdifferential is not the only mapping that satisfies these properties. Therefore, we propose the following *functional subdifferential* that captures any mapping satisfying (7.3) and (7.4).

Definition 7.4 (Functional subdifferential)
Given a convex function $f : \mathbb{R}^n \mapsto \mathbb{R}$ and an open convex set $S \subset \mathbb{R}^n$, a pair $\left(\mathring{D}_\epsilon f,\, \partial_\epsilon f\right)$ is called a *functional subdifferential* of $f(\cdot)$ over S where $\epsilon \in \mathbb{R}_{\ge 0}$ (called the *slack*), $\partial_\epsilon f : S \mapsto 2^{\mathbb{R}^n}$, and $\mathring{D}_\epsilon f : S \mapsto \mathbb{R}_{\ge 0} \cup \{+\infty\}$, if and only if the following properties hold:

- For any $\boldsymbol{x} \in S$, $\partial_\epsilon f(\boldsymbol{x})$ is a nonempty, convex, closed, and bounded set.
- There exists $L \in \mathbb{R}_+ \cup \{0^+\}$ such that $\mathring{D}_\epsilon f(\boldsymbol{x}) \ge \frac{\epsilon}{2L}$ for $\boldsymbol{x} \in S$ and $\epsilon \in \mathbb{R}_+$.
- For any $\boldsymbol{x} \in S$, $\epsilon \in \mathbb{R}_{\ge 0}$, and $\boldsymbol{y} \in S$, it holds that

$$
f(\boldsymbol{y}) \ge \left(f(\boldsymbol{x}) + \max_{\boldsymbol{g}\in\partial_\epsilon f(\boldsymbol{x})} \boldsymbol{g}^\intercal(\boldsymbol{y}-\boldsymbol{x})\right) - \epsilon \tag{7.5}
$$

- There exists $\beta \in \mathbb{R}_{\ge 0}$ such that for any $\boldsymbol{x} \in S$, $\epsilon \in \mathbb{R}_{\ge 0}$, and $\boldsymbol{y} \in S$ such that $\|\boldsymbol{y}-\boldsymbol{x}\|_2 \le \mathring{D}_\epsilon f(\boldsymbol{x})$, it holds that

$$
f(\boldsymbol{y}) \le \left(f(\boldsymbol{x}) + \max_{\boldsymbol{g}\in\partial_\epsilon f(\boldsymbol{x})} \boldsymbol{g}^\intercal(\boldsymbol{y}-\boldsymbol{x})\right) + \frac{\beta}{2}\|\boldsymbol{y}-\boldsymbol{x}\|_2^2 \tag{7.6}
$$

The constants L and β are called the *associated constants* of the functional subdifferential $\left(\mathring{D}_\epsilon f,\, \partial_\epsilon f\right)$. Of note, for $L' \geq L$ and $\beta' \geq \beta$, $(L',\, \beta')$ is also a pair of associated constants of $\left(\mathring{D}_\epsilon f,\, \partial_\epsilon f\right)$. ◁

Remark 7.2 We have included $\mathring{D}_\epsilon f(\boldsymbol{x})$ in the functional subdifferential definition to simplify the presentation and analysis; alternatively, we could define $\mathring{D}_\epsilon f(\boldsymbol{x})$ as the maximum distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ such that (7.6) holds. The TRAFS algorithm only needs the solution of a minimax problem involving $\partial_\epsilon f(\boldsymbol{x})$.

7.1.2 Rules for computing the functional subdifferential

This subsection presents compositional rules to compute the functional subdifferential. We assume all functions are convex in $\mathbb{R}^n$. For a function $f(\cdot)$, we denote its functional subdifferential as $\left(\mathring{D}_\epsilon f,\, \partial_\epsilon f\right)$ and the associated constants as $(L_f,\, \beta_f)$. We assume all functional subdifferentials are defined over an open convex set $S \subset \mathbb{R}^n$ unless otherwise specified.

We start with four terminal cases for constructing the functional subdifferential.

Proposition 7.4 (Functional subdifferential for smooth functions) *If $f(\cdot)$ is β-smooth over S, then $\left(\mathring{D}_\epsilon f,\, \partial_\epsilon f\right)$ is a functional subdifferential of $f(\cdot)$ with associated constants $(0^+,\, \beta)$ where*

$$\mathring{D}_\epsilon f(\boldsymbol{x}) \coloneqq +\infty, \quad \partial_\epsilon f(\boldsymbol{x}) \coloneqq \{\nabla f(\boldsymbol{x})\}$$

Proof. It is straightforward to verify the properties in Definition 7.4. □

Proposition 7.5 (Functional subdifferential for sparsely nonsmooth functions) *Assume $f(\cdot)$ is L-Lipschitz and β-sparsely nonsmooth over S. Let $\partial^{\mathrm{r}}_\epsilon f(\boldsymbol{x};\, S)$ be the relaxed subdifferential of $f(\cdot)$ as defined in Definition 7.2. Then $\left(\mathring{D}_\epsilon f,\, \partial_\epsilon f\right)$ is a functional subdifferential of $f(\cdot)$ with associated constants $(L,\, \beta)$ where*

$$\mathring{D}_\epsilon f(\boldsymbol{x}) \coloneqq \frac{\epsilon}{2L}, \quad \partial_\epsilon f(\boldsymbol{x}) \coloneqq \partial^{\mathrm{r}}_{\frac{\epsilon}{2L}} f(\boldsymbol{x};\, S)$$

Proof. Use Lemma 7.3 to verify the properties in Definition 7.4. □

Proposition 7.6 (Functional subdifferential from the ϵ-subdifferential) *If $f(\cdot)$ is L-Lipschitz over S, then $\left(\mathring{D}_\epsilon f,\, \partial_\epsilon f\right)$ is a functional subdifferential of $f(\cdot)$ with associated constants $(L,\, 0)$ where*

$$\mathring{D}_\epsilon f(\boldsymbol{x}) \coloneqq \frac{\epsilon}{2L}, \quad \partial_\epsilon f(\boldsymbol{x}) \coloneqq \{\boldsymbol{g} \in \mathbb{R}^n \mid \forall \boldsymbol{y} \in \mathbb{R}^n :\ f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{g}^\intercal(\boldsymbol{y} - \boldsymbol{x}) - \epsilon\}$$

Proof. A proof of the first property in Definition 7.4 can found at Bagirov, Karmitsa, and Mäkelä [BKM14, Theorem 2.32]. Eq. (7.5) holds by the definition of $\partial_\epsilon f(\boldsymbol{x})$. For $(\boldsymbol{x},\, \boldsymbol{y},\, \epsilon) \in S^2 \times \mathbb{R}_{\geq 0}$ such that $\|\boldsymbol{y} - \boldsymbol{x}\|_2 \leq \mathring{D}_\epsilon f(\boldsymbol{x})$, we have $\partial f(\boldsymbol{y}) \subset \partial_\epsilon f(\boldsymbol{x})$ [BKM14, Theorem 2.33]. For any $\boldsymbol{g}_y \in \partial f(\boldsymbol{y})$, we have $f(\boldsymbol{x}) \geq f(\boldsymbol{y}) + \boldsymbol{g}_y^\intercal(\boldsymbol{x} - \boldsymbol{y})$, which implies $\max_{\boldsymbol{g} \in \partial_\epsilon f(\boldsymbol{x})} \boldsymbol{g}^\intercal(\boldsymbol{y} - \boldsymbol{x}) \geq \boldsymbol{g}_y^\intercal(\boldsymbol{y} - \boldsymbol{x}) \geq f(\boldsymbol{y}) - f(\boldsymbol{x})$ and thus proves (7.6). □

Proposition 7.7 (Functional subdifferential from the Goldstein subdifferential) *If $f(\cdot)$ is L-Lipschitz over S, then $\left(\mathring{D}_\epsilon f,\, \partial_\epsilon f\right)$ is a functional subdifferential of $f(\cdot)$ with associated constants $(L,\, 0)$ where*

$$\mathring{D}_\epsilon f(\boldsymbol{x}) \coloneqq \frac{\epsilon}{2L}, \quad \partial_\epsilon f(\boldsymbol{x}) \coloneqq \partial^{\mathrm{G}}_{\frac{\epsilon}{2L}} f(\boldsymbol{x}), \quad \partial^{\mathrm{G}}_\epsilon f(\boldsymbol{x}) \coloneqq \operatorname{cl}\operatorname{conv} \bigcup_{\boldsymbol{y}\in B_\epsilon[\boldsymbol{x}]} \partial f(\boldsymbol{y})$$

Proof. A proof of (7.5) can found at Bagirov, Karmitsa, and Mäkelä [BKM14, Theorem 3.12]. Eq. (7.6) can be proven similarly to Proposition 7.6 since $f(\boldsymbol{y}) \subset \partial_\epsilon f(\boldsymbol{x})$ by definition. □

Next we present a few compositional rules for computing the functional subdifferential through common operations.

Proposition 7.8 (Rule of composition) *Let $F(\boldsymbol{x}) \coloneqq h(f(\boldsymbol{x}))$ where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a convex function and $h : \mathbb{R} \mapsto \mathbb{R}$ is a convex non-decreasing function. Let $S \subset \mathbb{R}^n$ be an open convex set. Assume $f(\cdot)$ is L_1-Lipschitz over S and $\left(\mathring{D}_\epsilon f,\, \partial_\epsilon f\right)$ is a functional subdifferential of $f(\cdot)$ over S with associated constants $(L_f,\, \beta_f)$. Let $T \coloneqq \{f(\boldsymbol{x}) \mid \boldsymbol{x} \in S\}$. Assume $h(\cdot)$ is L_h-Lipschitz and β_h-sparsely nonsmooth over T and $\left(\mathring{D}_\epsilon h,\, \partial_\epsilon h\right)$ is the functional subdifferential of $h(\cdot)$ over T defined by Proposition 7.5 or Proposition 7.4. Define*

$$\xi(\boldsymbol{x},\, \epsilon) \coloneqq \max\left\{\xi \in \left[0,\, \frac{\epsilon}{\bar{h}'_\epsilon(\boldsymbol{x})}\right] \;\middle|\; \begin{smallmatrix}\forall \boldsymbol{y}\in S:\, \|\boldsymbol{y}-\boldsymbol{x}\|_2 \le \xi \implies \\ |f(\boldsymbol{y})-f(\boldsymbol{x})| \le \mathring{D}_{\epsilon - \bar{h}'_\epsilon(\boldsymbol{x})\xi} h(f(\boldsymbol{x}))\end{smallmatrix}\right\} \tag{7.7}$$
$$\text{where } \bar{h}'_\epsilon(\boldsymbol{x}) \coloneqq \sup_{\epsilon'\in[0,\,\epsilon]} \max \partial_{\epsilon'} h(f(\boldsymbol{x}))$$

Then $F(\cdot)$ is a convex function that has a functional subdifferential $\left(\mathring{D}_\epsilon F,\, \partial_\epsilon F\right)$ with associated constants $(L_F,\, \beta_F)$ where

$$\begin{aligned}
\mathring{D}_\epsilon F(\boldsymbol{x}) &\coloneqq \mathring{D}_{\xi(\boldsymbol{x},\,\epsilon)} f(\boldsymbol{x}) \\
\partial_\epsilon F(\boldsymbol{x}) &\coloneqq \left\{\alpha \boldsymbol{g} \;\middle|\; \alpha \in \partial_{\gamma(\boldsymbol{x},\,\epsilon)} h(f(\boldsymbol{x})),\, \boldsymbol{g} \in \partial_{\xi(\boldsymbol{x},\,\epsilon)} f(\boldsymbol{x})\right\} \\
\gamma(\boldsymbol{x},\, \epsilon) &\coloneqq \epsilon - \bar{h}'_\epsilon(\boldsymbol{x})\xi(\boldsymbol{x},\, \epsilon) \\
L_F \le L_t L_f, \quad L_t &\coloneqq L_h(2L_1+1), \quad \beta_F \coloneqq L_h\beta_f + L_1^2\beta_h
\end{aligned}$$

Alternatively, one can set $\xi(\boldsymbol{x},\, \epsilon) \coloneqq \frac{\epsilon}{L_t}$, which may yield smaller $\mathring{D}_\epsilon F(\boldsymbol{x})$.

Proof. It is straightforward to verify that $F(\cdot)$ is convex and that the first property in Definition 7.4 is satisfied. Of note, $\partial_\epsilon h(\cdot)$ is a closed interval, and we have $0 \le \bar{h}'_\epsilon(\boldsymbol{x}) \le L_h$ due to $h(\cdot)$ being non-decreasing and (7.2). We have $\frac{\epsilon}{L_t} \in \left[0,\, \frac{\epsilon}{\bar{h}'_\epsilon(\boldsymbol{x})}\right]$ since $\frac{\epsilon}{L_t} \le \frac{\epsilon}{L_h} \le \frac{\epsilon}{\bar{h}'_\epsilon(\boldsymbol{x})}$. We also have $\mathring{D}_{\epsilon - \bar{h}'_\epsilon(\boldsymbol{x})\frac{\epsilon}{L_t}} h(f(\boldsymbol{x})) \ge \frac{1}{2L_h}\left(\epsilon - \bar{h}'_\epsilon(\boldsymbol{x})\frac{\epsilon}{L_t}\right) \ge \frac{\epsilon}{2L_h}\left(1 - \frac{L_h}{L_t}\right) = \frac{\epsilon L_1}{L_h(2L_1+1)} = L_1\frac{\epsilon}{L_t} \ge |f(\boldsymbol{y}) - f(\boldsymbol{x})|$ when $\|\boldsymbol{y}-\boldsymbol{x}\|_2 \le \frac{\epsilon}{L_t}$. Combining these two facts yields $\xi(\boldsymbol{x},\, \epsilon) \ge \frac{\epsilon}{L_t}$ and $\mathring{D}_\epsilon F(\boldsymbol{x}) \ge \frac{\xi(\boldsymbol{x},\,\epsilon)}{2L_f} \ge \frac{\epsilon}{2L_t L_f}$, which proves the second property in Definition 7.4 and the last statement in Proposition 7.8.

For any $(\boldsymbol{x}, \boldsymbol{y}, \epsilon) \in S^2 \times \mathbb{R}_{\geq 0}$, define $\epsilon_1 \coloneqq \xi(\boldsymbol{x}, \epsilon)$ and $\epsilon_2 \coloneqq \gamma(\boldsymbol{x}, \epsilon)$ for simplicity of notation. Eq. (7.5) is then proven by

$$\begin{aligned}
F(\boldsymbol{y}) - F(\boldsymbol{x}) &\geq \max_{\alpha \in \partial_{\epsilon_2} h(f(\boldsymbol{x}))} \alpha(f(\boldsymbol{y}) - f(\boldsymbol{x})) - \epsilon_2 \\
&\geq \max_{\alpha \in \partial_{\epsilon_2} h(f(\boldsymbol{x}))} \alpha \left(\max_{\boldsymbol{g} \in \partial_{\epsilon_1} f(\boldsymbol{x})} \boldsymbol{g}^\intercal (\boldsymbol{y} - \boldsymbol{x}) - \epsilon_1 \right) - \epsilon_2 \\
&\geq \max_{\alpha \in \partial_{\epsilon_2} h(f(\boldsymbol{x}))} \alpha \left(\max_{\boldsymbol{g} \in \partial_{\epsilon_1} f(\boldsymbol{x})} \boldsymbol{g}^\intercal (\boldsymbol{y} - \boldsymbol{x}) \right) - \left(\max_{\alpha \in \partial_{\epsilon_2} h(f(\boldsymbol{x}))} \alpha \epsilon_1 \right) - \epsilon_2 \\
&\geq \max_{\boldsymbol{g}_F \in \partial_\epsilon F(\boldsymbol{x})} \boldsymbol{g}_F^\intercal (\boldsymbol{y} - \boldsymbol{x}) - \bar{h'}_\epsilon(\boldsymbol{x}) \epsilon_1 - \epsilon_2 = \max_{\boldsymbol{g}_F \in \partial_\epsilon F(\boldsymbol{x})} \boldsymbol{g}_F^\intercal (\boldsymbol{y} - \boldsymbol{x}) - \epsilon
\end{aligned}$$

Suppose $\|\boldsymbol{y} - \boldsymbol{x}\|_2 \leq \mathring{D}_\epsilon F(\boldsymbol{x})$. Eq. (7.7) yields $|f(\boldsymbol{y}) - f(\boldsymbol{x})| \leq \mathring{D}_{\epsilon_2} h(f(\boldsymbol{x}))$. Eq. (7.6) is thus proven by

$$\begin{aligned}
F(\boldsymbol{y}) - F(\boldsymbol{x}) &\leq \max_{\alpha \in \partial_{\epsilon_2} h(f(\boldsymbol{x}))} \alpha(f(\boldsymbol{y}) - f(\boldsymbol{x})) + \frac{\beta_h}{2} |f(\boldsymbol{y}) - f(\boldsymbol{x})|^2 \\
&\leq \max_{\alpha \in \partial_{\epsilon_2} h(f(\boldsymbol{x}))} \alpha \left(\max_{\boldsymbol{g} \in \partial_{\epsilon_1} f(\boldsymbol{x})} \boldsymbol{g}^\intercal (\boldsymbol{y} - \boldsymbol{x}) + \frac{\beta_f}{2} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \right) + \frac{\beta_h L_1^2}{2} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \\
&\leq \max_{\boldsymbol{g}_F \in \partial_\epsilon F(\boldsymbol{x})} \boldsymbol{g}_F^\intercal (\boldsymbol{y} - \boldsymbol{x}) + \frac{\bar{h'}_\epsilon(\boldsymbol{x}) \beta_f + L_1^2 \beta_h}{2} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \\
&\leq \max_{\boldsymbol{g}_F \in \partial_\epsilon F(\boldsymbol{x})} \boldsymbol{g}_F^\intercal (\boldsymbol{y} - \boldsymbol{x}) + \frac{\beta_F}{2} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2
\end{aligned}$$

□

Corollary 7.9 (Rule of outer linearity)
If $F(\boldsymbol{x}) \coloneqq af(\boldsymbol{x}) + b$ where $a \in \mathbb{R}_+$ and $b \in \mathbb{R}$, then $\left(\mathring{D}_\epsilon F,\ \partial_\epsilon F \right)$ is a functional subdifferential of $F(\cdot)$ with associated constants $(aL_f,\ a\beta_f)$ where

$$\mathring{D}_\epsilon F(\boldsymbol{x}) \coloneqq \mathring{D}_{\frac{\epsilon}{a}} f(\boldsymbol{x}), \quad \partial_\epsilon F(\boldsymbol{x}) \coloneqq \left\{ a\boldsymbol{g} \;\middle|\; \boldsymbol{g} \in \partial_{\frac{\epsilon}{a}} f(\boldsymbol{x}) \right\}$$

Proof. Set $h(x) \coloneqq ax + b$ in Proposition 7.8. □

Proposition 7.10 (Rule of inner linearity)
Let $F(\boldsymbol{x}) \coloneqq f(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$ where $f : \mathbb{R}^m \mapsto \mathbb{R}$ is convex, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, and $\boldsymbol{b} \in \mathbb{R}^m$. Let $S \subset \mathbb{R}^n$ be an open convex set. Let $T \coloneqq \{\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \mid \boldsymbol{x} \in S\}$. Assume $\left(\mathring{D}_\epsilon f,\ \partial_\epsilon f \right)$ is a functional subdifferential of $f(\cdot)$ over T with associated constants $(L_f,\ \beta_f)$. Then $\left(\mathring{D}_\epsilon F,\ \partial_\epsilon F \right)$ is a functional subdifferential of $F(\cdot)$ over S with associated constants $(\sigma_{\max}(\boldsymbol{A}) L_f,\ \sigma_{\max}(\boldsymbol{A})^2 \beta_f)$ where

$$\begin{aligned}
\mathring{D}_\epsilon F(\boldsymbol{x}) &\coloneqq \frac{1}{\sigma_{\max}(\boldsymbol{A})} \mathring{D}_\epsilon f(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}) \\
\partial_\epsilon F(\boldsymbol{x}) &\coloneqq \{\boldsymbol{A}^\intercal \boldsymbol{g} \mid \boldsymbol{g} \in \partial_\epsilon f(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})\} \\
\sigma_{\max}(\boldsymbol{A}) &\coloneqq \sup_{\boldsymbol{x}:\ \|\boldsymbol{x}\|_2 \leq 1} \|\boldsymbol{A}\boldsymbol{x}\|_2
\end{aligned}$$

Of note, $\sigma_{\max}(\boldsymbol{A})$ is the operator norm of $\boldsymbol{A}$ induced by the ℓ_2-norm, which equals to the largest singular value of $\boldsymbol{A}$.

Proof. It is straightforward to verify the first two properties in Definition 7.4.

For any $(\boldsymbol{x},\, \boldsymbol{y},\, \epsilon) \in S^2 \times \mathbb{R}_{\geq 0}$, we have

$$\begin{aligned} F(\boldsymbol{y}) - F(\boldsymbol{x}) &= f(\boldsymbol{A}\boldsymbol{y}+\boldsymbol{b}) - f(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}) \geq \max_{\boldsymbol{g}_0 \in \partial_\epsilon f(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b})} \boldsymbol{g}_0^\intercal(\boldsymbol{A}\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}) - \epsilon \\ &= \max_{\boldsymbol{g}_0 \in \partial_\epsilon f(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b})} (\boldsymbol{A}^\intercal \boldsymbol{g}_0)^\intercal(\boldsymbol{y}-\boldsymbol{x}) - \epsilon = \max_{\boldsymbol{g} \in \partial_\epsilon F(\boldsymbol{x})} \boldsymbol{g}^\intercal(\boldsymbol{y}-\boldsymbol{x}) - \epsilon, \end{aligned}$$

which proves (7.5).

If $\|\boldsymbol{y}-\boldsymbol{x}\|_2 \leq \mathring{D}_\epsilon F(\boldsymbol{x})$, then $\|\boldsymbol{A}\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2 \leq \sigma_{\max}(\boldsymbol{A})\|\boldsymbol{y}-\boldsymbol{x}\|_2 \leq \mathring{D}_\epsilon f(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b})$. Thus (7.6) is proven by

$$\begin{aligned} F(\boldsymbol{y}) - F(\boldsymbol{x}) &= f(\boldsymbol{A}\boldsymbol{y}+\boldsymbol{b}) - f(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}) \\ &\leq \max_{\boldsymbol{g}_0 \in \partial_\epsilon f(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b})} \boldsymbol{g}_0^\intercal(\boldsymbol{A}\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}) + \frac{\beta_f}{2}\|\boldsymbol{A}\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2 \\ &\leq \max_{\boldsymbol{g} \in \partial_\epsilon F(\boldsymbol{x})} \boldsymbol{g}^\intercal(\boldsymbol{y}-\boldsymbol{x}) + \frac{\sigma_{\max}(\boldsymbol{A})^2\beta_f}{2}\|\boldsymbol{y}-\boldsymbol{x}\|_2^2 \end{aligned}$$

□

Proposition 7.11 (Rule of sum)
Let $F(\boldsymbol{x}) \coloneqq \sum_{i=1}^m f_i(\boldsymbol{x})$. Then $\left(\mathring{D}_\epsilon F,\, \partial_\epsilon F\right)$ is a functional subdifferential of $F(\cdot)$ with associated constants $(L_F,\, \beta_F)$ where

$$\begin{aligned} &\mathring{D}_\epsilon F(\boldsymbol{x}) \coloneqq \min_{i \in [\![1,\, m]\!]} \mathring{D}_{\xi_i(\boldsymbol{x},\, \epsilon)} f_i(\boldsymbol{x}), \quad \boldsymbol{\xi}(\boldsymbol{x},\, \epsilon) \in \operatorname*{argmax}^*_{\boldsymbol{\xi} \in \epsilon\Delta_m} \min_{i \in [\![1,\, m]\!]} \mathring{D}_{\xi_i} f_i(\boldsymbol{x}) \\ &\partial_\epsilon F(\boldsymbol{x}) \coloneqq \left\{ \sum_{i=1}^m \boldsymbol{g}_i \;\middle|\; \forall i \in [\![1,\, m]\!] :\; \boldsymbol{g}_i \in \partial_{\xi_i(\boldsymbol{x},\, \epsilon)} f_i(\boldsymbol{x}) \right\} \\ &L_F \leq m \max_{i \in [\![1,\, m]\!]} L_{f_i}, \quad \beta_F \coloneqq \sum_{i=1}^m \beta_{f_i} \end{aligned}$$

One can also set $\boldsymbol{\xi}(\boldsymbol{x},\, \epsilon) = \bar{\boldsymbol{\xi}} \in \epsilon\Delta_m$ as a constant. Then $L_F = \max_{i \in [\![1,\, m]\!]} \frac{L_{f_i}}{\bar{\xi}_i}$.

Proof. It is straightforward to verify that $\partial_\epsilon F(\boldsymbol{x})$ is nonempty, convex, closed, and bounded. By taking $\boldsymbol{\xi}(\boldsymbol{x},\, \epsilon) = \frac{\epsilon}{m}\mathbf{1}$, we have $\mathring{D}_\epsilon F(\boldsymbol{x}) \geq \min_{i \in [\![1,\, m]\!]} \mathring{D}_{\frac{\epsilon}{m}} f_i(\boldsymbol{x}) \geq \frac{\epsilon}{2} \min_{i \in [\![1,\, m]\!]} \frac{1}{mL_{f_i}}$. Thus the first two properties in Definition 7.4 are satisfied. One can also verify that $\max_{\boldsymbol{g} \in \partial_\epsilon F(\boldsymbol{x})} \boldsymbol{g}^\intercal(\boldsymbol{y}-\boldsymbol{x}) = \sum_{i=1}^m \max_{\boldsymbol{g}_i \in \partial_{\xi_i(\boldsymbol{x},\, \epsilon)} f_i(\boldsymbol{x})} \boldsymbol{g}_i^\intercal(\boldsymbol{y}-\boldsymbol{x})$, which implies (7.5) and (7.6). □

Corollary 7.12 (Sum of smooth and nonsmooth functions)
Let $F(\boldsymbol{x}) \coloneqq f(\boldsymbol{x}) + g(\boldsymbol{x})$ where $f(\cdot)$ is β_f-smooth and $g(\cdot)$ has a functional subdifferential $\left(\mathring{D}_\epsilon g,\, \partial_\epsilon g\right)$ with associated constants $(L_g,\, \beta_g)$. Then $\left(\mathring{D}_\epsilon g,\, \partial_\epsilon F\right)$ is a functional subdifferential of $F(\cdot)$ with associated constants $(L_g,\, \beta_f + \beta_g)$ where

$$\partial_\epsilon F(\boldsymbol{x}) \coloneqq \{\nabla f(\boldsymbol{x}) + \boldsymbol{c} \mid \boldsymbol{c} \in \partial_\epsilon g(\boldsymbol{x})\}$$

Proof. Take Proposition 7.4 into Proposition 7.11 and set $\boldsymbol{\xi}(\boldsymbol{x},\, \epsilon) = \begin{bmatrix} 0 & \epsilon \end{bmatrix}^\intercal$. □

Remark 7.3 Corollary 7.12 is an example where the functional subdifferential is easier to compute than the relaxed/Clarke/Goldstein subdifferentials. Computing the functional subdifferential of $F(\boldsymbol{x}) = f(\boldsymbol{x}) + g(\boldsymbol{x})$ at $\boldsymbol{x}_0$ needs $\nabla f(\boldsymbol{x}_0)$. However, computing the relaxed subdifferential at $\boldsymbol{x}_0$ requires $\nabla f(\boldsymbol{x}')$ for nonsmooth points $\boldsymbol{x}'$ of $g(\cdot)$, and computing the other two subdifferentials is more complicated.

Proposition 7.13 (Rule of max)
Let $F(\boldsymbol{x}) \coloneqq \max_{i\in[\![1,\,m]\!]} f_i(\boldsymbol{x})$. Assume $f_i(\cdot)$ is L_i-Lipschitz. Note that L_i can be different from L_{f_i}, the associated constant of the functional subdifferential $\left(\mathring{D}_\epsilon f_i,\, \partial_\epsilon f_i\right)$. Then $\left(\mathring{D}_\epsilon F,\, \partial_\epsilon F\right)$ is a functional subdifferential of $F(\cdot)$ with associated constants $(L_F,\, \beta_F)$ where

$$
\begin{aligned}
\partial_\epsilon F(\boldsymbol{x}) &\coloneqq \operatorname{cl}\operatorname{conv} \bigcup_{i\in\mathcal{A}_\epsilon(\boldsymbol{x})} \mathcal{P}_{\epsilon;\,i}(\boldsymbol{x}) \\
\mathring{D}_\epsilon F(\boldsymbol{x}) &\coloneqq \min_{i\in[\![1,\,m]\!]} \left(\begin{cases} \mathcal{D}_{\epsilon;\,i}(\boldsymbol{x}) + \frac{\delta_i(\boldsymbol{x})}{2L_i} & \text{if } i \in \mathcal{A}_\epsilon(\boldsymbol{x}) \\ \frac{\delta_i(\boldsymbol{x})}{\max_{j\neq k}(L_j+L_k)} & \text{otherwise} \end{cases} \right) \\
L_F &\leq \max_{i\in[\![1,\,m]\!]} \max\{L_i,\, L_{f_i}\}, \quad \beta_F \coloneqq \max_{i\in[\![1,\,m]\!]} \beta_{f_i} \\
\text{where} \quad \delta_i(\boldsymbol{x}) &\coloneqq F(\boldsymbol{x}) - f_i(\boldsymbol{x}), \quad \mathcal{A}_\epsilon(\boldsymbol{x}) \coloneqq \{i \mid i \in [\![1,\,m]\!] \text{ and } \delta_i(\boldsymbol{x}) \leq \epsilon\}, \\
\mathcal{P}_{\epsilon;\,i}(\boldsymbol{x}) &\coloneqq \partial_{\epsilon-\delta_i(\boldsymbol{x})} f_i(\boldsymbol{x}), \quad \mathcal{D}_{\epsilon;\,i}(\boldsymbol{x}) \coloneqq \min\left\{\mathring{D}_{\epsilon-\delta_i(\boldsymbol{x})} f_i(\boldsymbol{x}),\, \mathring{D}_\epsilon f_i(\boldsymbol{x})\right\}
\end{aligned}
$$

Proof. It is straightforward to verify the first two properties in Definition 7.4.

One can verify that for any $(\boldsymbol{x},\, \boldsymbol{y}) \in S^2$, there exist convex multipliers $\{\alpha_i\}_{i\in\mathcal{A}_\epsilon(\boldsymbol{x})}$ such that

$$
\begin{aligned}
\max_{\boldsymbol{g}\in\partial_\epsilon F(\boldsymbol{x})} \boldsymbol{g}^\intercal(\boldsymbol{y}-\boldsymbol{x}) &= \sum_{i\in\mathcal{A}_\epsilon(\boldsymbol{x})} \alpha_i \left(\max_{\boldsymbol{g}_i\in\mathcal{P}_{\epsilon;\,i}(\boldsymbol{x})} \boldsymbol{g}_i^\intercal(\boldsymbol{y}-\boldsymbol{x}) \right) \\
&\leq \sum_{i\in\mathcal{A}_\epsilon(\boldsymbol{x})} \alpha_i (f_i(\boldsymbol{y}) - f_i(\boldsymbol{x}) + \epsilon - \delta_i(\boldsymbol{x})) \\
&= \sum_{i\in\mathcal{A}_\epsilon(\boldsymbol{x})} \alpha_i (f_i(\boldsymbol{y}) + \epsilon - F(\boldsymbol{x})) \leq F(\boldsymbol{y}) - F(\boldsymbol{x}) + \epsilon,
\end{aligned}
$$

which proves (7.5).

Take any $(\boldsymbol{x},\, \boldsymbol{y}) \in S^2$ such that $\|\boldsymbol{y}-\boldsymbol{x}\|_2 \leq \mathring{D}_\epsilon F(\boldsymbol{x})$. To prove (7.6), we consider the following cases for $k \in [\![1,\, m]\!]$. Let $\boldsymbol{g}_k \in \operatorname{argmax}^*_{\boldsymbol{g}\in\mathcal{P}_{\epsilon;\,k}(\boldsymbol{x})} \boldsymbol{g}^\intercal(\boldsymbol{y}-\boldsymbol{x})$.

Case 1: $k \in \mathcal{A}_\epsilon(\boldsymbol{x})$ and $\|\boldsymbol{x}-\boldsymbol{y}\|_2 \leq \mathring{D}_\epsilon f_k(\boldsymbol{x})$. We have

$$
\begin{aligned}
f_k(\boldsymbol{y}) &\leq f_k(\boldsymbol{x}) + \boldsymbol{g}_k^\intercal(\boldsymbol{y}-\boldsymbol{x}) + \frac{\beta_{f_k}}{2}\|\boldsymbol{y}-\boldsymbol{x}\|_2^2 \\
&\leq F(\boldsymbol{x}) + \max_{\boldsymbol{g}\in\partial_\epsilon F(\boldsymbol{x})} \boldsymbol{g}^\intercal(\boldsymbol{y}-\boldsymbol{x}) + \frac{\beta_F}{2}\|\boldsymbol{y}-\boldsymbol{x}\|_2^2
\end{aligned}
$$

Case 2: $k \in \mathcal{A}_\epsilon(\boldsymbol{x})$ and $\|\boldsymbol{x}-\boldsymbol{y}\|_2 > \mathring{D}_\epsilon f_k(\boldsymbol{x})$. Let $\boldsymbol{y}_0 \coloneqq \boldsymbol{x} + \mathring{D}_\epsilon f_k(\boldsymbol{x}) \frac{\boldsymbol{y}-\boldsymbol{x}}{\|\boldsymbol{y}-\boldsymbol{x}\|_2}$. Then $\|\boldsymbol{y}_0-\boldsymbol{x}\|_2 = \mathring{D}_\epsilon f_k(\boldsymbol{x})$. Since $\|\boldsymbol{y}-\boldsymbol{y}_0\|_2 = \|\boldsymbol{y}-\boldsymbol{x}\|_2 - \mathring{D}_\epsilon f_k(\boldsymbol{x}) \leq \mathcal{D}_{\epsilon;\,k}(\boldsymbol{x}) + \frac{\delta_k(\boldsymbol{x})}{2L_k} - \mathring{D}_\epsilon f_k(\boldsymbol{x}) \leq$

$\frac{\delta_k(\boldsymbol{x})}{2L_k}$, we have

$$|f_k(\boldsymbol{y}) - f_k(\boldsymbol{y}_0)| \le \frac{\delta_k(\boldsymbol{x})}{2}, \quad |g_k^\intercal(\boldsymbol{y} - \boldsymbol{y}_0)| \le \frac{\delta_k(\boldsymbol{x})}{2} \tag{7.8}$$

Let $\boldsymbol{h}_k \in \operatorname{argmax}^*_{\boldsymbol{g} \in \mathcal{P}_{\epsilon;\,k}(\boldsymbol{x})} \boldsymbol{g}^\intercal(\boldsymbol{y}_0 - \boldsymbol{x})$. The property of functional subdifferential of $f_k(\cdot)$ implies $f_k(\boldsymbol{y}_0) - f_k(\boldsymbol{x}) \le \boldsymbol{h}_k^\intercal(\boldsymbol{y}_0 - \boldsymbol{x}) + \frac{\beta_{f_k}}{2}\|\boldsymbol{y}_0 - \boldsymbol{x}\|_2^2$. Since $\boldsymbol{y}_0 - \boldsymbol{x}$ and $\boldsymbol{y} - \boldsymbol{x}$ are collinear, we have $\boldsymbol{h}_k = \boldsymbol{g}_k$, yielding

$$f_k(\boldsymbol{y}_0) - f_k(\boldsymbol{x}) \le \boldsymbol{g}_k^\intercal(\boldsymbol{y}_0 - \boldsymbol{x}) + \frac{\beta_{f_k}}{2}\|\boldsymbol{y}_0 - \boldsymbol{x}\|_2^2 \le \boldsymbol{g}_k^\intercal(\boldsymbol{y}_0 - \boldsymbol{x}) + \frac{\beta_F}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \tag{7.9}$$

Combining (7.8) and (7.9) yields

$$\begin{aligned}
f_k(\boldsymbol{y}) &= f_k(\boldsymbol{x}) + (f_k(\boldsymbol{y}) - f_k(\boldsymbol{y}_0)) + (f_k(\boldsymbol{y}_0) - f_k(\boldsymbol{x})) \\
&\le f_k(\boldsymbol{x}) + \frac{\delta_k(\boldsymbol{x})}{2} + \left(\boldsymbol{g}_k^\intercal(\boldsymbol{y}_0 - \boldsymbol{y}) + \boldsymbol{g}_k^\intercal(\boldsymbol{y} - \boldsymbol{x}) + \frac{\beta_F}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2\right) \\
&\le f_k(\boldsymbol{x}) + \frac{\delta_k(\boldsymbol{x})}{2} + \left(\frac{\delta_k(\boldsymbol{x})}{2} + \max_{\boldsymbol{g} \in \partial_\epsilon F(\boldsymbol{x})} \boldsymbol{g}^\intercal(\boldsymbol{y} - \boldsymbol{x}) + \frac{\beta_F}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2\right) \\
&= F(\boldsymbol{x}) + \max_{\boldsymbol{g} \in \partial_\epsilon F(\boldsymbol{x})} \boldsymbol{g}^\intercal(\boldsymbol{y} - \boldsymbol{x}) + \frac{\beta_F}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2
\end{aligned}$$

Case 3: $k \in [\![1, m]\!] \setminus \mathcal{A}_\epsilon(\boldsymbol{x})$. Let $t \in \operatorname{argmax}^*_{i \in [\![1, m]\!]} f_i(\boldsymbol{x})$. Define $r_k(\boldsymbol{z}) \coloneqq f_t(\boldsymbol{z}) - f_k(\boldsymbol{z})$ for $\boldsymbol{z} \in S$. Then $|r_k(\boldsymbol{x}) - r_k(\boldsymbol{y})| \le (L_t + L_k)\|\boldsymbol{x} - \boldsymbol{y}\|_2$. We also have $r_k(\boldsymbol{x}) = \delta_k(\boldsymbol{x})$. Since $\|\boldsymbol{y} - \boldsymbol{x}\|_2 \le \frac{\delta_k(\boldsymbol{x})}{L_t + L_k}$, we have $|r_k(\boldsymbol{y}) - r_k(\boldsymbol{x})| \le \delta_k(\boldsymbol{x})$ and thus $r_k(\boldsymbol{y}) \ge 0$, which implies $f_k(\boldsymbol{y}) \le f_t(\boldsymbol{y})$.

Combining the above three cases proves (7.6). □

Proposition 7.14 (Functional subdifferential of the absolute value)
Let $f(\boldsymbol{x}) \coloneqq |x|$ for $x \in \mathbb{R}$. Then $\left(\mathring{D}_\epsilon f,\, \partial_\epsilon f\right)$ is a functional subdifferential of $f(\cdot)$ with associated constants $(1,\, 0)$ where

$$\mathring{D}_\epsilon f(x) \coloneqq \begin{cases} |x| & \text{if } |x| > \frac{\epsilon}{2} \\ +\infty & \text{otherwise} \end{cases}, \quad \partial_\epsilon f(x) \coloneqq \begin{cases} \{\operatorname{sign}(x)\} & \text{if } |x| > \frac{\epsilon}{2} \\ [-1,\, 1] & \text{otherwise} \end{cases}$$

Proof. It is straightforward to verify all properties in Definition 7.4. □

Proposition 7.15 (Functional subdifferential of the ℓ_1-norm)
Let $f(\boldsymbol{x}) \coloneqq \|\boldsymbol{x}\|_1 = \sum_{i=1}^n |x_i|$ for $\boldsymbol{x} \in \mathbb{R}^n$. Without loss of generality, assume $|x_i| \le |x_{i+1}|$ for $i \in [\![1, n-1]\!]$. Then $\left(\mathring{D}_\epsilon f,\, \partial_\epsilon f\right)$ is a functional subdifferential of $f(\cdot)$ with associated constants $(n,\, 0)$ where

$$\mathring{D}_\epsilon f(\boldsymbol{x}) \coloneqq \begin{cases} |x_{T+1}| & \text{if } T < n \\ +\infty & \text{otherwise} \end{cases}, \quad \partial_\epsilon f(\boldsymbol{x}) \coloneqq \{\boldsymbol{g} \in \mathbb{R}^n \mid \boldsymbol{l} \le \boldsymbol{g} \le \boldsymbol{u}\}$$

$$\text{where } \begin{cases} l_i = -1,\, u_i = 1 & \text{if } i \le T \\ l_i = u_i = \operatorname{sign}(x_i) & \text{otherwise} \end{cases}, \quad T \coloneqq \max\left\{ t \in [\![1, n]\!] \;\middle|\; \sum_{i=1}^t |x_i| \le \frac{\epsilon}{2} \right\}$$

Algorithm 7.1 Trust Region Adversarial Functional Subdifferential (TRAFS)

Input: A convex function $f : \mathbb{R}^n \mapsto \mathbb{R}$
Input: A bounded closed convex set $C \subset \mathbb{R}^n$
Input: A functional subdifferential oracle $\partial_\epsilon f(\cdot)$ over $S \subset \mathbb{R}^n$ with $C \subset S$
Input: A starting point $\boldsymbol{x}_0 \in C$
Input: A sequence of functional subdifferential slack parameters $\epsilon_0, \epsilon_1, \ldots \in \mathbb{R}_+$
Input: A sequence of trust region constraints $\eta_0, \eta_1, \ldots \in \mathbb{R}_+$
Input: Line search parameters $\tau \in (0, 1)$ and $\rho \in (0, 1)$ (default: $\tau = 0.8$, $\rho = 0.5$)

for $k = 0, 1, \ldots$ **do**
  Define a convex set $C_k \coloneqq \{\boldsymbol{d} \in \mathbb{R}^n \mid \boldsymbol{x}_k + \boldsymbol{d} \in C \text{ and } \|\boldsymbol{d}\|_2 \leq \eta_k\}$
  $\boldsymbol{d}_k \leftarrow \operatorname{argmin}^*_{\boldsymbol{d} \in C_k} \left(\max_{\boldsymbol{g} \in \partial_{\epsilon_k} f(\boldsymbol{x}_k)} \boldsymbol{g}^\intercal \boldsymbol{d} \right)$, $\boldsymbol{g}_k \leftarrow \operatorname{argmax}^*_{\boldsymbol{g} \in \partial_{\epsilon_k} f(\boldsymbol{x}_k)} \boldsymbol{g}^\intercal \boldsymbol{d}_k$ ▷ Find a descent direction
  if $\boldsymbol{g}_k^\intercal \boldsymbol{d}_k \geq 0$ **then** ▷ See Theorem 7.16
    $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k$
  else
    $\lambda_k \leftarrow 1$
    while $f(\boldsymbol{x}_k + \lambda_k \boldsymbol{d}_k) > f(\boldsymbol{x}_k) + \rho \lambda_k \boldsymbol{g}_k^\intercal \boldsymbol{d}_k$ **do** ▷ Backtracking line search
      $\lambda_k \leftarrow \tau \lambda_k$
    end while
    $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + \lambda_k \boldsymbol{d}_k$
  end if
end for

Proof. Treat $f(\cdot)$ as the sum of absolute values of coordinates. Then apply Proposition 7.11 and Proposition 7.14. □

Remark 7.4 Proposition 7.5 gives another candidate of the functional subdifferential of the ℓ_1 norm with better associated constants, but it poses a computational challenge to work with the set $\left\{ \boldsymbol{y} \;\middle|\; \min_i |y_i| = 0,\ \|\boldsymbol{y} - \boldsymbol{x}\|_2 \leq \frac{\epsilon}{2\sqrt{n}} \right\}$. Therefore, we use Proposition 7.15 in practice.

7.2 The TRAFS algorithm

7.2.1 The TRAFS algorithm and its convergence analysis

Algorithm 7.1 describes the TRAFS algorithm. Below we introduce some notation to facilitate our analysis.

Definition 7.5
In the analysis of Algorithm 7.1, we use the following notation:

- $\boldsymbol{x}^* \in \operatorname{argmin}^*_{\boldsymbol{x} \in C} f(\boldsymbol{x})$ is an optimal solution.
- $\boldsymbol{z}_k \coloneqq \boldsymbol{x}^* - \boldsymbol{x}_k$ is the error vector at iteration k.
- $R \coloneqq \sup_{(\boldsymbol{x}, \boldsymbol{y}) \in C^2} \|\boldsymbol{x} - \boldsymbol{y}\|_2$ is the diameter of C.
- $\delta_k \coloneqq f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*)$ is the optimality gap at iteration k.
- $\bar{\delta}_k \coloneqq f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*) - \epsilon_k$ is the optimality gap relative to ϵ_k at iteration k.

- $(L,\, \beta)$ are the associated constants of the functional subdifferential. ◁

We first show that the line search in Algorithm 7.1 is guaranteed to terminate.

Theorem 7.16
With the notation defined in Algorithm 7.1 and Definition 7.5, at each iteration k, it holds that:

- *If $\boldsymbol{g}_k^\intercal \boldsymbol{d}_k \geq 0$, then $f(\boldsymbol{x}_k) \leq f(\boldsymbol{x}^*) + \epsilon_k$.*
- *If $\boldsymbol{g}_k^\intercal \boldsymbol{d}_k < 0$, then the line search on Line 8 terminates in T_k iterations where*

$$T_k \leq \log_{\tau^{-1}} \max\left\{\frac{2L\eta_k}{\epsilon_k},\, \frac{\beta\eta_k^2}{2(1-\rho)(-\boldsymbol{g}_k^\intercal \boldsymbol{d}_k)},\, 1\right\} + 1 \tag{7.10}$$

Proof. Let $\boldsymbol{z}_k \coloneqq \boldsymbol{x}^* - \boldsymbol{x}_k$. Let $\boldsymbol{g}_k^* \in \operatorname{argmax}^*_{\boldsymbol{g}\in\partial_{\epsilon_k} f(\boldsymbol{x}_k)} \boldsymbol{g}^\intercal \boldsymbol{z}_k$. Let $\boldsymbol{d}_k^* \coloneqq \alpha_k \boldsymbol{z}_k$ where $\alpha_k \coloneqq \min\left\{\frac{\eta_k}{\|\boldsymbol{z}_k\|_2},\, 1\right\}$. Clearly $\boldsymbol{d}_k^* \in C_k$. If $\boldsymbol{g}_k^{*\intercal} \boldsymbol{z}_k < 0$, then

$$\boldsymbol{g}_k^\intercal \boldsymbol{d}_k = \min_{\boldsymbol{d}\in C_k} \max_{\boldsymbol{g}\in\partial_{\epsilon_k} f(\boldsymbol{x}_k)} \boldsymbol{d}^\intercal \boldsymbol{g} \leq \max_{\boldsymbol{g}\in\partial_{\epsilon_k} f(\boldsymbol{x}_k)} \boldsymbol{d}_k^{*\intercal} \boldsymbol{g} = \alpha_k \boldsymbol{g}_k^{*\intercal} \boldsymbol{z}_k < 0$$

Therefore, $\boldsymbol{g}_k^\intercal \boldsymbol{d}_k \geq 0$ implies $\boldsymbol{g}_k^{*\intercal} \boldsymbol{z}_k \geq 0$. With (7.5) we have $f(\boldsymbol{x}_k) \leq f(\boldsymbol{x}^*) - \boldsymbol{g}_k^{*\intercal} \boldsymbol{z}_k + \epsilon_k \leq f(\boldsymbol{x}^*) + \epsilon_k$ when $\boldsymbol{g}_k^\intercal \boldsymbol{d}_k \geq 0$, which proves the first statement in Theorem 7.16.

Assuming $\boldsymbol{g}_k^\intercal \boldsymbol{d}_k < 0$ and $\lambda_k \eta_k \leq \frac{\epsilon_k}{2L} \leq \mathring{D}_{\epsilon_k} f(\boldsymbol{x})$, a sufficient condition for the line search on Line 8 to terminate is:

$$\begin{aligned}
& f(\boldsymbol{x}_k + \lambda_k \boldsymbol{d}_k) \leq f(\boldsymbol{x}_k) + \rho\lambda_k \boldsymbol{g}_k^\intercal \boldsymbol{d}_k \\
\Longleftarrow\quad & \lambda_k \boldsymbol{g}_k^\intercal \boldsymbol{d}_k + \frac{\beta}{2}\lambda_k^2 \|\boldsymbol{d}_k\|_2^2 \leq \rho\lambda_k \boldsymbol{g}_k^\intercal \boldsymbol{d}_k && \text{Applying (7.6)} \\
\Longleftarrow\quad & \frac{\beta}{2}\lambda_k^2 \eta_k^2 \leq -(1-\rho)\lambda_k \boldsymbol{g}_k^\intercal \boldsymbol{d}_k && \text{Applying } \|\boldsymbol{d}_k\|_2 \leq \eta_k \\
\Longleftrightarrow\quad & \lambda_k \leq -\frac{2(1-\rho)}{\beta\eta_k^2} \boldsymbol{g}_k^\intercal \boldsymbol{d}_k
\end{aligned}$$

A sufficient termination condition of Line 8 is thus $\lambda_k \leq \min\left\{\frac{\epsilon_k}{2\eta_k L},\, -\frac{2(1-\rho)}{\beta\eta_k^2}\boldsymbol{g}_k^\intercal \boldsymbol{d}_k,\, 1\right\}$. Eq. (7.10) then follows. □

Now we proceed to analyze the convergence rate of Algorithm 7.1.

Lemma 7.17
With the notation defined in Algorithm 7.1 and Definition 7.5, at iteration k, assume $\eta_k \leq \|\boldsymbol{z}_k\|_2$ and $\bar{\delta}_k > 0$. Then

$$\boldsymbol{g}_k^\intercal \boldsymbol{d}_k \leq -\frac{\eta_k}{\|\boldsymbol{z}_k\|_2}\bar{\delta}_k \tag{7.11}$$

When the line search on Line 8 terminates, it holds that:

$$\lambda_k \eta_k \geq \min\{\tau\bar{\eta}_k,\, \eta_k\} \quad \text{where } \bar{\eta}_k \coloneqq \min\left\{\frac{\epsilon_k}{2L},\, \frac{2(1-\rho)}{\beta R}\bar{\delta}_k\right\} \tag{7.12}$$

Proof. Let $\boldsymbol{g}_k^* \in \operatorname{argmax}^*_{\boldsymbol{g}\in\partial_{\epsilon_k} f(\boldsymbol{x}_k)} \min_{\boldsymbol{d}\in C_k} \boldsymbol{g}^\intercal \boldsymbol{d}$ and $\boldsymbol{d}_k^* \in \operatorname{argmin}^*_{\boldsymbol{d}\in C_k} \boldsymbol{g}_k^{*\intercal} \boldsymbol{d}$. Since C_k and $\partial_{\epsilon_k} f(\boldsymbol{x}_k)$ are both convex and compact, we have $\boldsymbol{g}_k^{*\intercal}\boldsymbol{d}_k^* = \boldsymbol{g}_k^\intercal \boldsymbol{d}_k$ due to the minimax theorem. Since $\eta_k \le \|\boldsymbol{z}_k\|_2$ and C is convex, we have $\frac{\eta_k}{\|\boldsymbol{z}_k\|_2}\boldsymbol{z}_k \in C_k$. We thus have $\boldsymbol{g}_k^{*\intercal}\boldsymbol{d}_k^* \le \boldsymbol{g}_k^{*\intercal}\frac{\eta_k}{\|\boldsymbol{z}_k\|_2}\boldsymbol{z}_k$ due to the definition of $\boldsymbol{d}_k^*$. Eq. (7.5) implies $\boldsymbol{g}_k^{*\intercal}\boldsymbol{z}_k \le f(\boldsymbol{x}^*) - f(\boldsymbol{x}_k) + \epsilon_k = -\bar{\delta}_k$. Combining these results yields $\boldsymbol{g}_k^\intercal \boldsymbol{d}_k = \boldsymbol{g}_k^{*\intercal}\boldsymbol{d}_k^* \le \frac{\eta_k}{\|\boldsymbol{z}_k\|_2}\boldsymbol{g}_k^{*\intercal}\boldsymbol{z}_k \le -\frac{\eta_k}{\|\boldsymbol{z}_k\|_2}\bar{\delta}_k$, which proves (7.11).

Assuming $\lambda_k\eta_k \le \frac{\epsilon_k}{2L} \le \mathring{D}_{\epsilon_k} f(\boldsymbol{x})$, a sufficient condition for the line search on Line 8 to terminate is:

$$
\begin{aligned}
& f(\boldsymbol{x}_k + \lambda_k \boldsymbol{d}_k) \le f(\boldsymbol{x}_k) + \rho\lambda_k \boldsymbol{g}_k^\intercal \boldsymbol{d}_k \\
\Longleftarrow \quad & \lambda_k \boldsymbol{g}_k^\intercal \boldsymbol{d}_k + \frac{\beta}{2}\lambda_k^2 \|\boldsymbol{d}_k\|_2^2 \le \rho\lambda_k \boldsymbol{g}_k^\intercal \boldsymbol{d}_k && \text{Applying (7.6)} \\
\Longleftarrow \quad & \frac{\beta}{2}\lambda_k^2\eta_k^2 - (1-\rho)\frac{\lambda_k\eta_k}{R}\bar{\delta}_k \le 0 && \text{Applying (7.11) and } \|\boldsymbol{z}_k\|_2 \le R \\
\Longleftrightarrow \quad & \lambda_k\eta_k \le \frac{2(1-\rho)}{\beta R}\bar{\delta}_k
\end{aligned}
$$

Therefore, when $\lambda_k\eta_k \le \min\frac{\epsilon_k}{2L}, \frac{2(1-\rho)}{\beta R}\bar{\delta}_k = \bar{\eta}_k$, the line search termination condition is satisfied. If $\eta_k < \bar{\eta}_k$, then $\lambda_k = 1$ suffices. Otherwise, the search procedure can use at most one more iteration after $\lambda_k \le \frac{\bar{\eta}_k}{\eta_k}$ is satisfied. We thus have (7.12). □

Lemma 7.18

With the notation defined in Algorithm 7.1 and Definition 7.5, also assume $f(\cdot)$ is L-Lipschitz over S (we can take L to be the largest value of the Lipschitz constant and the associated constant of the functional subdifferential, which does not violate the definition of Lipschitz continuity or Definition 7.4). Let $D \in [\delta_0, c]$ be a constant where c is defined below. Given an arbitrary $\epsilon \in \left(0, \frac{2D}{e}\right)$ where e is the base of the natural logarithm, set $\epsilon_k = \frac{\epsilon}{2}$ and $\eta_k = \frac{\epsilon}{L}$ for $k \in \mathbb{N}$. Then

$$
\delta_T \le \frac{D}{e} + \frac{\epsilon}{2} \quad \textit{where } T := \left\lceil \frac{c}{\epsilon} \right\rceil,\ c := \max\left\{4L, \frac{\beta R}{1-\rho}\right\}\frac{R}{\rho\tau} \tag{7.13}
$$

Proof. First, note that the interval $[\delta_0, c]$ is non-empty since $c \ge LR \ge \delta_0$. We also have $\epsilon < \frac{2D}{e} < D \le c$ and thus $T \ge 2$.

If $\delta_k \le \epsilon$ for some $k \in [\![0, T]\!]$, then $\delta_T \le \delta_k \le \epsilon \le \frac{D}{e} + \frac{\epsilon}{2}$ since the objective value is non-increasing. If $\eta_k > \|\boldsymbol{z}_k\|_2$ for some $k \in [\![0, T]\!]$, then $\delta_k \le L\|\boldsymbol{z}_k\|_2 \le L\eta_k \le \epsilon$, which also implies (7.13).

Now let's assume $\eta_k \le \|\boldsymbol{z}_k\|_2$ and $\delta_k \ge \epsilon$ for all $k \in [\![0, T]\!]$. Then $\bar{\delta}_k \ge \epsilon - \epsilon_k = \frac{\epsilon}{2}$. Note that $\eta_k \ge \frac{\epsilon_k}{2L} \ge \bar{\eta}_k$, which implies $\lambda_k\eta_k \ge \tau\bar{\eta}_k$ by Lemma 7.17. Also note that $\frac{\rho\tau}{R}\bar{\eta}_k \ge \frac{\rho\tau}{R}\min\left\{\frac{\epsilon_k}{2L}, \frac{2(1-\rho)}{\beta R}(\epsilon - \epsilon_k)\right\} = \frac{\epsilon}{c}$. With Lemma 7.17 and $\|\boldsymbol{z}_k\|_2 \le R$, we have

$$
\begin{aligned}
& \bar{\delta}_{k+1} - \bar{\delta}_k = f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k) \le \rho\lambda_k \boldsymbol{g}_k^\intercal \boldsymbol{d}_k \le -\frac{\rho}{R}\lambda_k\eta_k\bar{\delta}_k \le -\frac{\rho\tau}{R}\bar{\eta}_k\bar{\delta}_k \le -\frac{\epsilon}{c}\bar{\delta}_k \\
& \delta_T \le \left(1 - \frac{\epsilon}{c}\right)^T \bar{\delta}_0 + \epsilon_T \le \left(1 - \frac{\epsilon}{c}\right)^{\frac{c}{\epsilon}} D + \frac{\epsilon}{2} \le \left(\lim_{x\to 0^+}(1-x)^{\frac{1}{x}}\right) D + \frac{\epsilon}{2} = \frac{D}{e} + \frac{\epsilon}{2}
\end{aligned}
$$

□

In Lemma 7.18, we can set $\epsilon = \left(1 - \frac{2}{e}\right)D$, so that after $T = \left\lceil \frac{ce}{(e-2)D} \right\rceil$ iterations, the optimality gap is reduced by half to $\delta_T \leq \frac{D}{2}$. We can recursively apply this procedure to obtain an $\mathcal{O}\left(\frac{1}{\epsilon}\right)$ algorithm, as shown in the following theorem.

Theorem 7.19
With the notation defined in Algorithm 7.1 and Definition 7.5, also assume $f(\cdot)$ is L-Lipschitz over S. Choose $D \in [\delta_0, c]$ where c is defined in Lemma 7.18. There exist sequences $\{\eta_k\}_{k\in\mathbb{N}}$ and $\{\epsilon_k\}_{k\in\mathbb{N}}$ constructed from D and c such that in Algorithm 7.1, for any $\epsilon^\dagger \in \mathbb{R}_+$, it holds that

$$\forall k \geq T^\dagger : f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*) \leq \epsilon^\dagger$$
$$\text{where } T^\dagger := \frac{2ce}{(e-2)\epsilon^\dagger} + \left\lceil \log_2 \frac{D}{\epsilon^\dagger} \right\rceil = \mathcal{O}\left(\frac{1}{\epsilon^\dagger}\right)$$

Proof. Let $\bar{\epsilon}^{(i)} := \frac{1}{2^i}\left(1 - \frac{2}{e}\right)D$ and $T^{(i)} := \left\lceil \frac{c}{\bar{\epsilon}^{(i)}} \right\rceil = \left\lceil \frac{ce}{(e-2)D} 2^i \right\rceil$ for $i \in \mathbb{N}$. For any $i \in \mathbb{N}$, define the sequences $\left\{\eta_k^{(i)}\right\}_{k\in[\![0, T^{(i)}-1]\!]}$ and $\left\{\epsilon_k^{(i)}\right\}_{k\in[\![0, T^{(i)}-1]\!]}$ as $\eta_k^{(i)} = \frac{\bar{\epsilon}^{(i)}}{L}$ and $\epsilon_k^{(i)} = \frac{\bar{\epsilon}^{(i)}}{2}$ as in Lemma 7.18. Define the sequences $\{\eta_k\}$ and $\{\epsilon_k\}$ by concatenating the sequences $\left\{\eta_k^{(i)}\right\}$ and $\left\{\epsilon_k^{(i)}\right\}$ for $i \in \mathbb{N}$.

Let $S_0 := 0$ and $S_{i+1} := S_i + T^{(i)}$ for $i \in \mathbb{N}$. By induction on i, it is easy to verify that $\delta_{S_i} \leq \frac{D}{2^i}$ using Lemma 7.18. Let $m := \left\lceil \log_2 \frac{D}{\epsilon^\dagger} \right\rceil$. We have $\delta_{S_m} \leq \epsilon^\dagger$ where

$$S_m = \sum_{i=0}^{m-1} T^{(i)} \leq \sum_{i=0}^{m-1} \left(\frac{ce}{(e-2)D} 2^i + 1 \right) \leq \frac{ce}{(e-2)D} \frac{2D}{\epsilon^\dagger} + m = T^\dagger \qquad \square$$

7.2.2 Convergence analysis for strongly convex functions

Next we show that TRAFS converges in $\mathcal{O}(\epsilon^{-0.5})$ iterations for strongly convex objective functions. Recall that a function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is α-strongly convex over $S \subset \mathbb{R}^n$ if and only if the function $\boldsymbol{x} \mapsto f(\boldsymbol{x}) - \frac{\alpha}{2}\|\boldsymbol{x}\|_2^2$ is convex over S, which is equivalent to $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{g}_x^\intercal(\boldsymbol{y} - \boldsymbol{x}) + \frac{\alpha}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2$ for $(\boldsymbol{x}, \boldsymbol{y}) \in S^2$ and $\boldsymbol{g}_x \in \partial f(\boldsymbol{x})$. We have the following property for strongly convex functions:

Lemma 7.20
If a function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is α-strongly convex over a bounded closed convex set $C \subset S$ and $\boldsymbol{x}^* \in \operatorname{argmin}^*_{\boldsymbol{x}\in C} f(\boldsymbol{x})$, then for any $\boldsymbol{x} \in C$,

$$\|\boldsymbol{x} - \boldsymbol{x}^*\|_2 \leq \sqrt{\frac{2(f(\boldsymbol{x}) - f(\boldsymbol{x}^*))}{\alpha}} \tag{7.14}$$

Proof. By Proposition 7.1, there exists $\boldsymbol{g} \in \partial f(\boldsymbol{x}^*)$ such that $\boldsymbol{g}^\intercal(\boldsymbol{x} - \boldsymbol{x}^*) \geq 0$. Eq. (7.14) is then proven by the following inequality due to strong convexity:

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}^*) + \boldsymbol{g}^\intercal(\boldsymbol{x} - \boldsymbol{x}^*) + \frac{\alpha}{2}\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2 \geq f(\boldsymbol{x}^*) + \frac{\alpha}{2}\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2 \qquad \square$$

Lemma 7.21
With the notation defined in Algorithm 7.1 and Definition 7.5, also assume $f(\cdot)$ is α-strongly convex and L-Lipschitz over S. At iteration k, assume $\eta_k \le \|\boldsymbol{z}_k\|_2$ and $\delta_k > \epsilon_k$. Then

$$\boldsymbol{g}_k^\intercal \boldsymbol{d}_k \le -\eta_k \sqrt{\frac{\alpha}{2}} \frac{\delta_k - \epsilon_k}{\sqrt{\delta_k}} \tag{7.15}$$

When the line search on Line 8 terminates, it holds that:

$$\begin{aligned} &\lambda_k \eta_k \ge \min\{\tau \tilde{\eta}_k,\, \eta_k\} \\ &\text{where } \tilde{\eta}_k \coloneqq \min\left\{\frac{\epsilon_k}{2L},\, \max\left\{\sqrt{\frac{2\alpha}{\delta_k}},\, \frac{\alpha}{L}\right\} \frac{(1-\rho)}{\beta}(\delta_k - \epsilon_k)\right\} \end{aligned} \tag{7.16}$$

Proof. Eq. (7.14) implies $\|\boldsymbol{z}_k\|_2 \le \sqrt{\frac{2\delta_k}{\alpha}} \le \sqrt{\frac{2L\|\boldsymbol{z}_k\|_2}{\alpha}}$. We thus have $\|\boldsymbol{z}_k\|_2 \le \frac{2L}{\alpha}$. Eq. (7.15) is obtained by substituting $\|\boldsymbol{z}_k\|_2 \le \sqrt{\frac{2\delta_k}{\alpha}}$ into (7.11). Substituting $\|\boldsymbol{z}_k\|_2 \le \min\left\{\frac{2L}{\alpha}, \sqrt{\frac{2\delta_k}{\alpha}}\right\}$ into the proof of Lemma 7.17 yields (7.16). □

Theorem 7.22
With the notation defined in Algorithm 7.1 and Definition 7.5, also assume $f(\cdot)$ is α-strongly convex and L-Lipschitz over S. Let $D \in \mathbb{R}_+$ be any constant such that $\delta_0 \le D$. Set

$$\eta_k = \frac{a}{(k+1)^2}, \quad \epsilon_k = \frac{aL}{4\max\{k^2,\, 1\}}$$
$$\text{where } a \coloneqq \max\left\{\left(\frac{5.4}{\rho\tau c}\right)^2 \frac{L}{2\alpha}, \frac{9D}{L}\right\}, \quad c \coloneqq \min\left\{\frac{1}{8}, \frac{5\alpha(1-\rho)}{16\beta}\right\}$$

Then

$$\forall k \ge 1 :\ \delta_k \le \frac{aL}{k^2} \tag{7.17}$$

Equivalently, for any $\epsilon \in \mathbb{R}_+$, it holds that

$$\forall k \ge \sqrt{\frac{aL}{\epsilon}} :\ f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*) \le \epsilon$$

Proof. Note that for $k \in [\![1, 3]\!]$, $\frac{aL}{k^2} \ge D \ge \delta_0 \ge \delta_k$ holds due to our choice of a.

Now we prove by induction. Assume $k \ge 3$. If $\eta_k \ge \|\boldsymbol{z}_k\|_2$, we have $\delta_{k+1} \le \delta_k \le L\eta_k \le \frac{aL}{(k+1)^2}$. If $\delta_k \le \epsilon_k$, then $\delta_{k+1} \le \delta_k \le \frac{aL}{4k^2} = \frac{aL}{(k+1)^2}\frac{1}{4}\left(1 + \frac{1}{k}\right)^2 \le \frac{aL}{(k+1)^2}$.

Thus we assume $k \ge 3$, $\eta_k < \|\boldsymbol{z}_k\|_2$, $\delta_k > \epsilon_k$, and $\frac{aL}{(k+1)^2} < \delta_k \le \frac{aL}{k^2}$. Substituting our assumptions and parameter choices into the definition of $\tilde{\eta}_k$ in (7.16) yields

$$\begin{aligned} \tilde{\eta}_k &\ge \min\left\{\frac{a}{8k^2}, \frac{\alpha(1-\rho)}{\beta L}\left(\frac{aL}{(k+1)^2} - \frac{aL}{4k^2}\right)\right\} \\ &= \min\left\{\frac{1}{8}, \frac{\alpha(1-\rho)}{\beta}\left(\frac{k^2}{(k+1)^2} - \frac{1}{4}\right)\right\}\frac{a}{k^2} \ge \min\left\{\frac{1}{8}, \frac{5\alpha(1-\rho)}{16\beta}\right\}\frac{a}{k^2} = \frac{ac}{k^2} \end{aligned}$$

For $k \geq 3$, we have $\frac{1}{(k+1)^2} = \frac{1}{k^2}\left(1 - \frac{1}{k+1}\right)^2 \geq \frac{9}{16k^2}$. We thus have $\eta_k = \frac{a}{(k+1)^2} \geq \frac{9a}{16k^2} \geq \frac{ac}{k^2} = \tilde{\eta}_k$. With Lemma 7.21, we have $\lambda_k \eta_k \geq \frac{ac\tau}{k^2}$ and

$$\delta_{k+1} - \delta_k \leq \rho \lambda_k \boldsymbol{g}_k^\intercal \boldsymbol{d}_k \leq -\rho\tau \frac{ac}{k^2}\sqrt{\frac{\alpha}{2}}\frac{\delta_k - \epsilon_k}{\sqrt{\delta_k}}$$

Note that $\delta_k \geq \frac{aL}{(k+1)^2} \geq \frac{9aL}{16k^2} = \frac{9}{4}\epsilon_k$ and $\delta_k \leq \frac{aL}{k^2} = 4\epsilon_k$, which is $\frac{3}{2}\sqrt{\epsilon_k} \leq \sqrt{\delta_k} \leq 2\sqrt{\epsilon_k}$. Consider the function $h(x) \coloneqq \frac{1}{x} - \frac{\epsilon_k}{x^3}$. We have $h'(x) = \frac{3\epsilon_k - x^2}{x^4}$, and thus the minimum of $f(\cdot)$ over $\left[\frac{3}{2}\sqrt{\epsilon_k}, 2\sqrt{\epsilon_k}\right]$ is obtained on the boundary, i.e., $h(x) \geq \frac{10}{27\sqrt{\epsilon_k}} = \frac{20}{27\sqrt{aL}}k$ for $x \in \left[\frac{3}{2}\sqrt{\epsilon_k}, 2\sqrt{\epsilon_k}\right]$. Therefore,

$$\frac{\delta_{k+1}}{\delta_k} \leq 1 - \rho\tau\frac{ac}{k^2}\sqrt{\frac{\alpha}{2}}h\left(\sqrt{\delta_k}\right) \leq 1 - \frac{10\rho\tau c}{27}\sqrt{\frac{2\alpha a}{L}}\frac{1}{k}$$

Consider the function $g_k(t) \coloneqq \frac{1-\frac{t}{k}}{\frac{k^2}{(k+1)^2}} = 1 + \frac{(2-t)k^2 + (1-2t)k - t}{k^3}$. We have $g_k(t) < 1$ when $t \geq 2$ and $k > 0$. Therefore, $\delta_{k+1} \leq \frac{k^2}{(k+1)^2} g_k(t)\delta_k \leq \frac{k^2}{(k+1)^2}\delta_k \leq \frac{aL}{(k+1)^2}$ where $t = \frac{10\rho\tau c}{27}\sqrt{\frac{2\alpha a}{L}} \geq 2$ due to our choice of a. □

7.2.3 Faster convergence with locally quadratic functional subdifferential

Compared to the projected gradient descent method with linear convergence for strongly convex smooth functions, Theorem 7.22 only guarantees a $\mathcal{O}(\epsilon^{-0.5})$ rate. The bottleneck is that (7.16) constrains $\tilde{\eta}_k = \mathcal{O}(\epsilon_k) = \mathcal{O}(\delta_k)$, so that (7.15) only decreases the objective by $\mathcal{O}\left(\eta_k\sqrt{\delta_k}\right) = \mathcal{O}(\delta_k^{1.5})$. We could achieve linear convergence of TRAFS if $\tilde{\eta}_k = \mathcal{O}\left(\sqrt{\delta_k}\right)$, which intuitively means that our functional subdifferential could "look ahead" quadratically further. Therefore, we introduce the *quadratic functional subdifferential*, which enables faster convergence of the TRAFS algorithm.

Definition 7.6 (Quadratic functional subdifferential)
Let $\left(\mathring{D}_\epsilon f, \partial_\epsilon f\right)$ be a functional subdifferential of $f(\cdot)$ over S as defined in Definition 7.4. Let $L_q \in \mathbb{R}_+ \cup \{0^+\}$ be a constant. We call $\left(\mathring{D}_\epsilon f, \partial_\epsilon f\right)$ a *L_q-quadratic functional subdifferential* at $(\boldsymbol{x}, \epsilon) \in S \times \mathbb{R}_+$ if $\mathring{D}_\epsilon f(\boldsymbol{x}) \geq \frac{\sqrt{\epsilon}}{L_q}$. ◁

Remark 7.5 If $f(\cdot)$ is smooth, then its functional subdifferential defined in Proposition 7.4 is a 0^+-quadratic functional subdifferential at any $(\boldsymbol{x}, \epsilon)$. For the ℓ_1 norm, Proposition 7.15 gives an L_q-quadratic functional subdifferential at $(\boldsymbol{x}, \epsilon)$ if there exists T such that $\sum_{i=1}^{T} |x_i| \leq \frac{\epsilon}{2}$ and $|x_{T+1}| \geq \frac{\sqrt{\epsilon}}{L_q}$.

Lemma 7.23
With the notation defined in Algorithm 7.1 and Definition 7.5, also assume $f(\cdot)$ is α-strongly convex over S. At iteration k, assume $\eta_k \leq \|\boldsymbol{z}_k\|_2$, $\delta_k > \epsilon_k$, and that $\left(\mathring{D}_\epsilon f, \partial_\epsilon f\right)$ is an L_q-quadratic functional subdifferential at $(\boldsymbol{x}_k, \epsilon_k)$. Then (7.15) still holds. When the line search

on Line 8 terminates, it holds that:

$$\begin{aligned} &\lambda_k \eta_k \geq \min\left\{\tau \mathring{\eta}_k,\, \eta_k\right\} \\ &\textit{where } \mathring{\eta}_k \coloneqq \min\left\{\frac{\sqrt{\epsilon_k}}{L_q},\, \frac{\sqrt{2\alpha}(1-\rho)}{\beta}\frac{\delta_k - \epsilon_k}{\sqrt{\delta_k}}\right\} \end{aligned} \tag{7.18}$$

Proof. Replace $\lambda_k \eta_k \leq \frac{\epsilon_k}{2L} \leq \mathring{D}_{\epsilon_k} f(\boldsymbol{x})$ with $\lambda_k \eta_k \leq \frac{\sqrt{\epsilon_k}}{L_q} \leq \mathring{D}_{\epsilon_k} f(\boldsymbol{x})$ and remove the Lipschitz assumption in the proof of Lemma 7.17 and Lemma 7.21. □

Lemma 7.24
With the notation defined in Algorithm 7.1 and Definition 7.5, also assume $f(\cdot)$ is α-strongly convex over S. At iteration k, assume $\eta_k > \|\boldsymbol{z}_k\|_2$, $\delta_k > \epsilon_k$, and that $\left(\mathring{D}_\epsilon f,\, \partial_\epsilon f\right)$ is a L_q-quadratic functional subdifferential at $(\boldsymbol{x}_k,\, \epsilon_k)$. Then

$$\boldsymbol{g}_k^\intercal \boldsymbol{d}_k \leq -\delta_k + \epsilon_k \tag{7.19}$$

A sufficient condition for the line search on Line 8 to terminate is:

$$\lambda_k \leq \min\left\{\frac{\sqrt{\epsilon_k}}{\eta_k L_q},\, \frac{2(1-\rho)}{\beta \eta_k^2}(\delta_k - \epsilon_k),\, 1\right\} \tag{7.20}$$

Proof. We have $\boldsymbol{z}_k \in C_k$ since $\|\boldsymbol{z}_k\|_2 < \eta_k$, which implies

$$\boldsymbol{g}_k^\intercal \boldsymbol{d}_k = \min_{\boldsymbol{d} \in C_k} \max_{\boldsymbol{g} \in \partial_\epsilon f(\boldsymbol{x}_k)} \boldsymbol{g}^\intercal \boldsymbol{d} \leq \max_{\boldsymbol{g} \in \partial_\epsilon f(\boldsymbol{x}_k)} \boldsymbol{g}^\intercal \boldsymbol{z}_k \leq f(\boldsymbol{x}^*) - f(\boldsymbol{x}_k) + \epsilon_k = -\delta_k + \epsilon_k$$

Assuming $\lambda_k \eta_k \leq \frac{\sqrt{\epsilon_k}}{L_q} \leq \mathring{D}_{\epsilon_k} f(\boldsymbol{x})$, (7.20) is proven similarly to Theorem 7.16. □

Theorem 7.25
With the notation defined in Algorithm 7.1 and Definition 7.5, also assume $f(\cdot)$ is α-strongly convex over S. Let $D_k \in [\delta_k,\, +\infty)$ and $L_q \in \mathbb{R}_+ \cup \{0^+\}$ be two constants. Set

$$\begin{aligned} &\epsilon_k = a\gamma D_k, \quad \eta_k = b(1-a)\sqrt{\gamma D_k} \\ &\textit{where } a \coloneqq \left(\frac{2bL_q}{\sqrt{4b^2L_q^2+1}+1}\right)^2, \quad b \coloneqq \frac{\sqrt{2\alpha}(1-\rho)}{\tilde{\beta}}, \quad \tilde{\beta} \coloneqq \max\{\alpha,\, \beta\}, \\ &\gamma \coloneqq \frac{1}{1+\rho\tau b(1-a)^2\sqrt{\frac{\alpha}{2}}} = \frac{1}{1+\frac{\alpha}{\tilde{\beta}}\rho(1-\rho)\tau(1-a)^2} \end{aligned}$$

If $\left(\mathring{D}_\epsilon f,\, \partial_\epsilon f\right)$ is an L_q-quadratic functional subdifferential at $(\boldsymbol{x}_k,\, \epsilon_k)$, then

$$\delta_{k+1} \leq \gamma D_k \tag{7.21}$$

Proof. If $\delta_k < \gamma D_k$, then $\delta_{k+1} \le \delta_k < \gamma D_k$, which yields (7.21). Thus we assume $\gamma D_k < \delta_k \le D_k$. Note that our choice of a satisfies $\frac{\sqrt{a}}{L_q} = b(1-a)$ when $L_q > 0$.

Case 1: $\eta_k \le \|\boldsymbol{z}_k\|_2$. With Lemma 7.23, we have

$$\begin{aligned}\mathring{\eta}_k &= \min\left\{\frac{\sqrt{\epsilon_k}}{L_q},\, b\frac{\delta_k-\epsilon_k}{\sqrt{\delta_k}}\right\} \ge \min\left\{\frac{\sqrt{a\gamma D_k}}{L_q},\, b\frac{\gamma D_k - a\gamma D_k}{\sqrt{\gamma D_k}}\right\} \\ &= \min\left\{\frac{\sqrt{a}}{L_q},\, b(1-a)\right\}\sqrt{\gamma D_k} = b(1-a)\sqrt{\gamma D_k} = \eta_k,\end{aligned}$$

which implies $\lambda_k\eta_k \ge \tau\eta_k$ with (7.18). Combining this with (7.15) yields

$$\begin{aligned}\delta_{k+1} - \delta_k &\le \rho\lambda_k \boldsymbol{g}_k^\intercal \boldsymbol{d}_k \le -\rho\tau b(1-a)\sqrt{\gamma D_k}\sqrt{\frac{\alpha}{2}}\frac{\delta_k-\epsilon_k}{\sqrt{\delta_k}} \\ &\le -\rho\tau b(1-a)\sqrt{\gamma D_k}\sqrt{\frac{\alpha}{2}}\frac{\gamma D_k - a\gamma D_k}{\sqrt{\gamma D_k}} = -\rho\tau b(1-a)^2\sqrt{\frac{\alpha}{2}}\gamma D_k\end{aligned}$$

We thus have $\delta_{k+1} \le \big(1 - \rho\tau b(1-a)^2\sqrt{\frac{\alpha}{2}}\gamma\big)D_k = \gamma D_k$ due to the definition of γ.

Case 2: $\eta_k > \|\boldsymbol{z}_k\|_2$. Since $\frac{\sqrt{a}}{L_q} = b(1-a)$ and $\alpha \le \tilde{\beta}$, we have

$$\frac{\sqrt{\epsilon_k}}{\eta_k L_q} = \frac{\sqrt{a}}{b(1-a)L_q} = 1 \tag{7.22}$$

$$\begin{aligned}\frac{2(1-\rho)}{\tilde{\beta}\eta_k^2}(\delta_k-\epsilon_k) &\ge \frac{2(1-\rho)}{\tilde{\beta}b^2(1-a)^2\gamma D_k}(\gamma D_k - a\gamma D_k) \\ &= \frac{2(1-\rho)}{\tilde{\beta}b^2(1-a)} \ge \frac{2(1-\rho)}{\tilde{\beta}b^2} = \frac{\tilde{\beta}}{\alpha(1-\rho)} \ge 1\end{aligned} \tag{7.23}$$

Substituting (7.22) and (7.23) into (7.20) yields $\lambda_k = 1$. With (7.19), we have

$$\delta_{k+1} - \delta_k \le \rho\lambda_k \boldsymbol{g}_k^\intercal \boldsymbol{d}_k \le -\rho(\delta_k - \epsilon_k) \le -\rho(1-a)\gamma D_k,$$

which implies $\delta_{k+1} \le (1-\rho(1-a)\gamma)D_k$. Since $\gamma \ge \frac{1}{1+\rho(1-a)}$, we have $\delta_{k+1} \le \gamma D_k$. □

Corollary 7.26
With the notation of Algorithm 7.1 and Definition 7.5, also assume $f(\cdot)$ is α-strongly convex and β-smooth over S. Use Proposition 7.4 to define the functional subdifferential. Let $D \in \mathbb{R}_+$ be any constant such that $\delta_0 \le D$. Set

$$\epsilon_k = 0, \quad \eta_k = b\sqrt{\gamma^k D}$$
$$\textit{where } b := \frac{\sqrt{2\alpha}(1-\rho)}{\beta}, \quad \gamma := \frac{1}{1+\frac{\alpha}{\beta}\rho(1-\rho)\tau}$$

Then

$$\forall k \in \mathbb{N}:\ \delta_k \le \gamma^k D \tag{7.24}$$

Equivalently, for any $\epsilon \in \mathbb{R}_+$, it holds that

$$\forall k \ge \frac{\log D - \log\epsilon}{\log\gamma^{-1}}:\ f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*) \le \epsilon$$

Proof. For strongly convex smooth functions, we have $\alpha \leq \beta$. With $L_q = 0$, $a = 0$ and $\tilde{\beta} = \beta$ in Theorem 7.25, we can prove (7.24) by induction on k. □

7.2.4 Almost-functional subdifferential for optimization beyond sparse nonsmoothness

When the objective function is only Lipschitz but not sparsely nonsmooth, a functional subdifferential that enables efficient solutions to the minimax problem can be harder to define. In this case, we can use the almost-functional subdifferential, defined as the following:

Definition 7.7 (Almost-functional subdifferential)
With the notation defined in Definition 7.4, a pair $\left(\mathring{D}_\epsilon f,\, \partial_\epsilon f\right)$ is called an *almost-functional subdifferential* of $f(\cdot)$ over S if all properties of Definition 7.4 are satisfied except that $\beta = \frac{\beta_0}{\epsilon}$ for a constant $\beta_0 \in \mathbb{R}_+$. We call $(L,\, \beta_0)$ the *associated constants* of $\left(\mathring{D}_\epsilon f,\, \partial_\epsilon f\right)$. ◁

Theorem 7.27
With the notation defined in Algorithm 7.1 and Definition 7.5, assume $f(\cdot)$ is L-Lipschitz and $\left(\mathring{D}_\epsilon f,\, \partial_\epsilon f\right)$ is an almost-functional subdifferential of $f(\cdot)$ with associated constants $(L,\, \beta_0)$. Given $\epsilon \in \mathbb{R}_+$, let $T = \operatorname{argmin}^* \{t \in \mathbb{N} \mid \delta_t \leq \epsilon\}$. Then $T = \mathcal{O}(\epsilon^{-2})$. If $f(\cdot)$ is also strongly convex, then $T = \mathcal{O}(\epsilon^{-1})$.

Proof. The first statement can be proven by setting $\beta = \frac{\beta_0}{\epsilon}$ in Theorem 7.19.

Now assume $f(\cdot)$ is α-strongly convex. Similar to Theorem 7.22, we set $\eta_k = \frac{b}{k+1}$ and $\epsilon_k = \frac{bL}{4\max\{k,1\}}$ for a constant b. By replacing β with $\frac{\beta_0}{\epsilon_k}$ in (7.16), we have $\tilde{\eta}_k \geq \frac{\sqrt{2\alpha}(1-\rho)}{\beta_0}\frac{\delta_k - \epsilon_k}{\sqrt{\delta_k}}\epsilon_k$ for sufficiently large k, which, when combined with (7.15), yields $\delta_{k+1} - \delta_k \leq -\frac{\rho\tau(1-\rho)\alpha}{\beta_0}\frac{(\delta_k-\epsilon_k)^2}{\delta_k}\epsilon_k$. With a proper choice of b and similar arguments to the proof of Theorem 7.22, we can prove $\delta_k \leq \frac{bL}{k}$, which yields $T = \mathcal{O}(\epsilon^{-1})$. □

One important example of a Lipschitz but not sparsely nonsmooth function is the ℓ_2 norm. It has the following almost-Functional subdifferential.

Proposition 7.28 (Almost-functional subdifferential of the ℓ_2 norm)
Let $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_2$. For $\boldsymbol{x} \in \mathbb{R}^n$ and $\epsilon \in \mathbb{R}_{\geq 0}$, define

$$\partial_\epsilon f(\boldsymbol{x}) \coloneqq \begin{cases} \left\{\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2}\right\} & \text{if } \|\boldsymbol{x}\|_2 > \frac{\epsilon}{2} \\ \{\boldsymbol{g} \in \mathbb{R}^n \mid \|\boldsymbol{g}\|_2 \leq 1\} & \text{otherwise} \end{cases} \tag{7.25}$$

Then $\partial_\epsilon f(\boldsymbol{x})$ satisfies (7.5) and the following inequalities:

$$\|\boldsymbol{x}\|_2 > \frac{\epsilon}{2} \implies \left(\begin{array}{l} \forall \boldsymbol{y} \;:\; \|\boldsymbol{y} - \boldsymbol{x}\|_2 \leq \frac{\epsilon}{4} \implies \\ \qquad f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \max\limits_{\boldsymbol{g} \in \partial_\epsilon f(\boldsymbol{x})} \boldsymbol{g}^\mathsf{T}(\boldsymbol{y} - \boldsymbol{x}) + \frac{2}{\epsilon}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \end{array} \right) \tag{7.26}$$

$$\|\boldsymbol{x}\|_2 \leq \frac{\epsilon}{2} \implies \left(\forall \boldsymbol{y} \in \mathbb{R}^n :\; f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \max_{\boldsymbol{g} \in \partial_\epsilon f(\boldsymbol{x})} \boldsymbol{g}^\mathsf{T}(\boldsymbol{y} - \boldsymbol{x}) \right) \tag{7.27}$$

Therefore, $\left(\mathring{D}_\epsilon f,\, \partial_\epsilon f\right)$ *is an almost-functional subdifferential of* $f(\cdot)$ *with associated constants* $L = 2$ *and* $\beta_0 = 4$ *where* $\mathring{D}_\epsilon f(\boldsymbol{x}) = \begin{cases} \frac{\epsilon}{4} & \text{if } \|\boldsymbol{x}\|_2 > \frac{\epsilon}{2} \\ +\infty & \text{otherwise} \end{cases}$.

Proof. **Case 1:** $\|\boldsymbol{x}\|_2 > \frac{\epsilon}{2}$. We have $\partial_\epsilon f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$, which implies (7.5) due to convexity. Given $\boldsymbol{y}$ such that $\|\boldsymbol{y} - \boldsymbol{x}\|_2 \le \frac{\epsilon}{4}$, let $\boldsymbol{d} \coloneqq \boldsymbol{y} - \boldsymbol{x}$. Let $a \coloneqq \|\boldsymbol{x}\|_2$, $b \coloneqq \|\boldsymbol{d}\|_2$, and $c \coloneqq \frac{\boldsymbol{x}^\intercal \boldsymbol{d}}{ab}$. Note that $c \in [-1,\, 1]$. Eq. (7.26) is then proven by

$$\begin{aligned} f(\boldsymbol{y}) - f(\boldsymbol{x}) - \nabla f(\boldsymbol{x})^\intercal(\boldsymbol{y} - \boldsymbol{x}) &= \|\boldsymbol{x} + \boldsymbol{d}\|_2 - \|\boldsymbol{x}\|_2 - \frac{\boldsymbol{x}^\intercal \boldsymbol{d}}{\|\boldsymbol{x}\|_2} \\ &= \sqrt{a^2 + b^2 + 2abc} - a - bc = \frac{(1 - c^2)b^2}{\sqrt{a^2 + b^2 + 2abc} + a + bc} \\ &\le \frac{(1 - c^2)b^2}{a + bc + a + bc} \le \frac{b^2}{2(a - b)} \le \frac{b^2}{\epsilon - \frac{\epsilon}{2}} = \frac{2}{\epsilon}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \end{aligned}$$

Case 2: $\|\boldsymbol{x}\|_2 \le \frac{\epsilon}{2}$. We have $\max_{\boldsymbol{g} \in \partial_\epsilon f(\boldsymbol{x})} \boldsymbol{g}^\intercal(\boldsymbol{y} - \boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{x}\|_2$. Eq. (7.5) follows from $\|\boldsymbol{y} - \boldsymbol{x}\|_2 \le \|\boldsymbol{y}\|_2 + \|\boldsymbol{x}\|_2 = f(\boldsymbol{y}) - f(\boldsymbol{x}) + 2\|\boldsymbol{x}\|_2 \le f(\boldsymbol{y}) - f(\boldsymbol{x}) + \epsilon$. Eq. (7.27) follows from $\|\boldsymbol{y} - \boldsymbol{x}\|_2 \ge \|\boldsymbol{y}\|_2 - \|\boldsymbol{x}\|_2 = f(\boldsymbol{y}) - f(\boldsymbol{x})$. □

7.2.5 An adaptive TRAFS implementation

It is often infeasible to set the values of η_k and ϵ_k based on our previous convergence analysis since the functional subdifferential constants and the Lipschitz constants can be challenging to compute. This section proposes heuristic strategies to adaptively adjust the values of η_k and ϵ_k.

We set η_k to be slightly larger than the maximum step length in the recent few iterations as an estimation of the step length of the current iteration. Formally, we set $\eta_k = \Gamma \max_{i \in [\![k - m_1,\, k-1]\!]} \lambda_i \|\boldsymbol{d}_i\|_2$. We choose $m_1 = 8$ and $\Gamma = \tau^{-2}$ in our implementation.

Our convergence analysis sets $\epsilon_{k+1} = c\delta_k = c(f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*))$ for some constant $c \in (0,\, 1)$. However, $f(\boldsymbol{x}^*)$ is typically unknown. Instead, we estimate δ_k from $f(\boldsymbol{x}_{k-1}) - f(\boldsymbol{x}_k)$. Assuming $\delta_k = ak^{-p}$ for $a \in \mathbb{R}_+$ and $p \in \mathbb{R}_+$, then we have

$$\frac{f(\boldsymbol{x}_{k-1}) - f(\boldsymbol{x}_k)}{\delta_k} = \frac{\delta_{k-1}}{\delta_k} - 1 = \left(1 + \frac{1}{k-1}\right)^p - 1 = \frac{p}{k} + o\left(\frac{1}{k}\right),$$

which suggests $\epsilon_{k+1} \approx \frac{c}{p} k(f(\boldsymbol{x}_{k-1}) - f(\boldsymbol{x}_k))$. In order to obtain a more robust estimation, we consider the recent history and set

$$\epsilon_{k+1} = t_k \min\{\epsilon_k,\, c_k s_k\}, \quad c_{k+1} = c_k t_k, \quad s_k \coloneqq \max_{j \in [\![k - m_2 + 1,\, k]\!]} j(f(\boldsymbol{x}_{j-1}) - f(\boldsymbol{x}_j))$$

The scaling factor c_k is initialized as $c_1 = 1$. We set $t_k \in \{1,\, \mu^-,\, \mu^+\}$ given parameters $\mu^- \in (0,\, 1)$ and $\mu^+ > 1$. If $\boldsymbol{g}_k^\intercal \boldsymbol{d}_k \ge 0$, we have $\delta_k \le \epsilon_k$ due to Theorem 7.16 and thus set $t_k = \mu^-$ to decrease ϵ_{k+1}. The other case is to set $t_k = \mu^+$ if $\delta_k \ge 2\mu^+\epsilon_k$. However, since δ_k is unknown, we assume $\eta_k / \|\boldsymbol{z}_k\|_2$ is large enough and consider the necessary condition

$\boldsymbol{g}_k^\intercal \boldsymbol{d}_k \le -(\delta_k - \epsilon_k) \le -(2\mu^+ - 1)\epsilon_k$ (see (7.11) and (7.19)) given $\delta_k \ge 2\mu^+\epsilon_k$. If $\boldsymbol{g}_k^\intercal \boldsymbol{d}_k \le -(2\mu^+ - 1)\epsilon_k$, we further check if using $\mu^+\epsilon_k$ results in a larger objective value decrease; if so, we set $t_k = \mu^+$. Moreover, we adopt a randomization strategy to tune t_k. Let $p_t \in (0,\, 1)$ be a parameter and $U_k \sim U(0,\, 1)$ be a random variable. At the k-th iteration, if $U_k \le p_t$, we randomly pick $\epsilon_k' \in \{\mu^-\epsilon_k,\, \mu^+\epsilon_k\}$ and set $t_k = \epsilon_k'/\epsilon_k$ if doing so results in a larger objective value decrease. The parameter p_t balances the effectiveness of the exploration against the additional computation cost. We choose $m_2 = 8$, $\mu^- = 0.5$, $\mu^+ = 1.5$, and $p_t = 0.2$ in our implementation.

Another important aspect of a practical implementation is the termination condition. We maintain a sequence $\{L_k\}$ as the lower bounds of the objective value:

$$\begin{aligned} L_{k+1} &= \max\{L_k,\, f(\boldsymbol{x}_k) - \epsilon_k + \Delta_k\} \\ \text{where } \Delta_k &\coloneqq \min_{\boldsymbol{d}\in C} \max_{\boldsymbol{g}\in\partial_\epsilon f(\boldsymbol{x}_k)} \boldsymbol{g}^\intercal \boldsymbol{d}, \quad L_0 = -\infty \end{aligned}$$

One can verify $f(\boldsymbol{x}^*) \ge L_k$. Given user-specified tolerance ϵ and maximum number of iterations T, the algorithm terminates when $f(\boldsymbol{x}_k) - L_k \le \epsilon$ or $k \ge T$. Since $C_k \subset C$, we have $\Delta_k \le \boldsymbol{g}_k^\intercal \boldsymbol{d}_k$ and thus we set $\Delta_k = -\infty$ without solving its minimax value if $\boldsymbol{g}_k^\intercal \boldsymbol{d}_k \le L_k + \epsilon_k - f(\boldsymbol{x}_k)$. For unconstrained problems, we assume the optimum is within $B_{\sqrt{n}}[\boldsymbol{x}_k]$ when $\eta_k \le 10^{-4}$, so we set $\Delta_k = \frac{\sqrt{n}}{\eta_k}\boldsymbol{g}_k^\intercal \boldsymbol{d}_k$ when $\eta_k \le 10^{-4}$ and set $\Delta_k = -\infty$ otherwise.

7.2.6 Solving the TRAFS minimax problem

Our implementation needs a user-provided callback function that returns the solution to the minimax problem on Line 3 in Algorithm 7.1. We restate the problem below:

$$\text{The TRAFS minimax subproblem:} \quad \operatorname*{argmin}_{\boldsymbol{d}\in C_k}{}^* \left(\max_{\boldsymbol{g}\in\partial_\epsilon f(\boldsymbol{x}_k)} \boldsymbol{g}^\intercal \boldsymbol{d} \right) \tag{7.28}$$

There are three general strategies for solving (7.28):

1. Directly solving the corresponding constrained optimization problem:

$$\begin{aligned} &\min_{(\boldsymbol{d},\, u)} \; u \\ &\text{subject to } \boldsymbol{d} \in C_k, \quad \max_{\boldsymbol{g}\in\partial_\epsilon f(\boldsymbol{x}_k)} \boldsymbol{g}^\intercal \boldsymbol{d} \le u \end{aligned} \tag{7.29}$$

2. When C is large enough so that $C_k = B_{\eta_k}[\boldsymbol{0}]$, the convexity and compactness of C_k and $\partial_\epsilon f(\boldsymbol{x}_k)$ imply:

$$\min_{\boldsymbol{d}\in C_k} \max_{\boldsymbol{g}\in\partial_\epsilon f(\boldsymbol{x}_k)} \boldsymbol{g}^\intercal \boldsymbol{d} = \max_{\boldsymbol{g}\in\partial_\epsilon f(\boldsymbol{x}_k)} \min_{\boldsymbol{d}:\, \|\boldsymbol{d}\|_2 \le \eta_k} \boldsymbol{g}^\intercal \boldsymbol{d} = -\eta_k \min_{\boldsymbol{g}\in\partial_\epsilon f(\boldsymbol{x}_k)} \|\boldsymbol{g}\|_2 \tag{7.30}$$

Let $\boldsymbol{g}^*$ be a solution to (7.30). Then $\boldsymbol{d}^* \coloneqq -\eta_k \frac{\boldsymbol{g}^*}{\|\boldsymbol{g}^*\|_2}$ is a solution to (7.28).

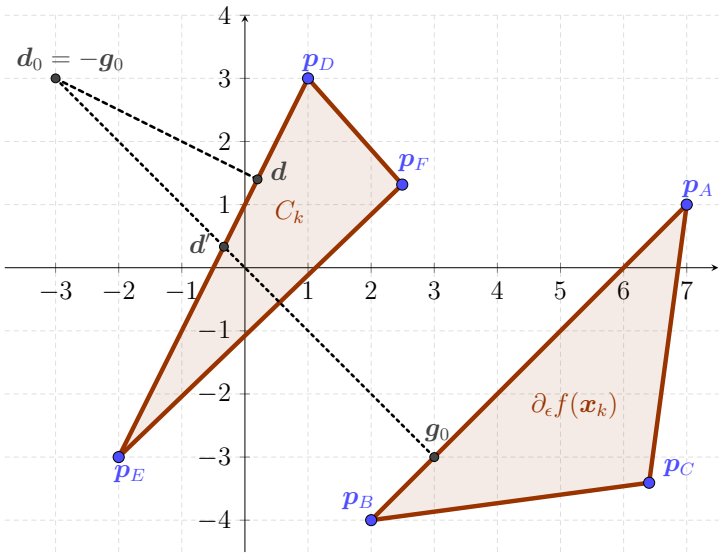


Figure 7.1: Incorrectness of projecting the unconstrained solution.
In this example, $\boldsymbol{d}$ is the projection of the unconstrained solution $\boldsymbol{d}_0$ onto the feasible set C_k. We have $\boldsymbol{d}^\intercal \boldsymbol{p}_A > 0$, which may lead to failed line search in a TRAFS iteration. By contrast, the point $\boldsymbol{d}'$ satisfies $\max_{\boldsymbol{g} \in \partial_\epsilon f(\boldsymbol{x}_k)} \boldsymbol{g}^\intercal \boldsymbol{d}' < 0$.

3. Under the same condition as above (i.e., when $C_k = B_{\eta_k}[\boldsymbol{0}]$), we can also solve a dual form:

$$\begin{aligned} &\min \|\boldsymbol{d}\|_2 \\ \text{subject to } &\max_{\boldsymbol{g} \in \partial_\epsilon f(\boldsymbol{x}_k)} \boldsymbol{g}^\intercal \boldsymbol{d} \leq -1 \end{aligned} \tag{7.31}$$

   Let $(\boldsymbol{d}_u^*, u)$ be a solution to (7.29) and $\boldsymbol{d}_v^*$ be a solution to (7.31). Define $v \coloneqq \|\boldsymbol{d}_v^*\|_2$. We have $u = 0$ if and only if (7.31) is infeasible. Otherwise, $\|\boldsymbol{d}_u^*\|_2 = \eta_k$ and $u < 0 < v$. Setting $\boldsymbol{d} = -\frac{1}{u}\boldsymbol{d}_u^*$ in (7.31) leads to $v \leq -\eta_k / u$. Similarly, setting $\boldsymbol{d} = \frac{\eta_k}{v}\boldsymbol{d}_v^*$ in (7.29) yields $u \leq -\eta_k / v$. We thus have $u = v$.

As will be shown in Section 7.3, different problems may use different formulations for best efficiency and/or best numerical stability. Here we consider a typical example where $C_k = B_{\eta_k}[\boldsymbol{0}]$ and $\partial_\epsilon f(\boldsymbol{x}_k)$ is a polytope with p vertices. Let the columns of $\boldsymbol{G} \in \mathbb{R}^{n \times p}$ be the vertices of $\partial_\epsilon f(\boldsymbol{x}_k)$. Eq. (7.29) becomes a Second-Order Cone Program (SOCP) with the constraints $\|\boldsymbol{d}\|_2 \leq \eta_k$ and $\boldsymbol{G}^\intercal \boldsymbol{d} \leq u$. Eq. (7.30) becomes a Quadratic Program (QP) $\min_{\boldsymbol{x} \in \Delta_p} \|\boldsymbol{G}\boldsymbol{x}\|_2$. Eq. (7.31) becomes another QP $\min_{\boldsymbol{G}^\intercal \boldsymbol{d} \leq -1} \|\boldsymbol{d}\|_2$. When $p \ll n$, solving (7.30) is often more efficient than solving (7.31). When p and n are comparable, working with (7.30) may be less numerically stable compared to (7.31) since we need to compute $\boldsymbol{g} = \boldsymbol{G}\boldsymbol{x}$ to obtain $\boldsymbol{d}$.

We remark that a tempting approach to solving (7.28), which is projecting the unconstrained solution, is incorrect. This method first solves $\boldsymbol{d}_0 \coloneqq \operatorname{argmin}^*_{\boldsymbol{d} \in B_{\eta_k}[\boldsymbol{0}]} \max_{\boldsymbol{g} \in \partial_\epsilon f(\boldsymbol{x}_k)} \boldsymbol{g}^\intercal \boldsymbol{d} = -\eta_k \frac{\boldsymbol{g}_0}{\|\boldsymbol{g}_0\|_2}$ with $\boldsymbol{g}_0 \coloneqq \operatorname{argmin}^*_{\boldsymbol{g} \in \partial_\epsilon f(\boldsymbol{x}_k)} \|\boldsymbol{g}\|_2$ and then sets $\boldsymbol{d} = \Pi_{C_k}(\boldsymbol{d}_0)$. As shown in Fig. 7.1, this approach does not necessarily yield a valid solution. One needs to properly solve or approximate the minimax problem to ensure the convergence of TRAFS.

7.3 Numerical experiments

7.3.1 Benchmark problems

We perform the numerical evaluation on nine classes of nonsmooth convex problems described below. For each problem class, we generate 50 instances. Our benchmark set thus contains 450 test cases.

The first six problem classes are convex nonsmooth benchmark problems used in previous work [HMM04; BKM14; NS15; Kar20]. They are defined by a single parameter n, the dimension of the problem. For each of them, we generate 50 instances by taking n uniformly spaced within $[\![10, n_{\max}]\!]$. We set $n_{\max} = 5000$ for functions with sparse gradients (MAXQ, CCB3B, and SPL) and $n_{\max} = 1200$ for others (DPL, CLQ, and CCB3A).

MAXQ: the generalization of MAXQ in Haarala, Miettinen, and Mäkelä [HMM04].

$$f(\boldsymbol{x}) = \max_{i \in [\![1,n]\!]} x_i^2, \quad x_{0i} = \begin{cases} i & \text{if } i \le \frac{n}{2} \\ -i & \text{otherwise} \end{cases}, \quad f_{\text{opt}} = 0$$

DPL: Dense Piecewise-Linear, a.k.a. the generalization of MXHILB in Haarala, Miettinen, and Mäkelä [HMM04]. Previous nonsmooth optimization methods failed to converge when $n \ge 1000$ [HMM04; BKM14; Kar20].

$$f(\boldsymbol{x}) = \max_{i \in [\![1,n]\!]} \left| \sum_{j=1}^{n} \frac{x_j}{i+j-1} \right|, \quad \boldsymbol{x}_0 = \mathbf{1}, \quad f_{\text{opt}} = 0$$

CLQ: Chained LQ in Haarala, Miettinen, and Mäkelä [HMM04].

$$f(\boldsymbol{x}) = \sum_{i=1}^{n-1} \max\left\{-x_i - x_{i+1},\, -x_i + x_{i+1} + (x_i^2 + x_{i+1}^2 - 1)\right\},$$
$$\boldsymbol{x}_0 = 0.5 \cdot \mathbf{1}, \quad f_{\text{opt}} = -(n-1)\sqrt{2}$$

CCB3A: Chained CB3 I in Haarala, Miettinen, and Mäkelä [HMM04].

$$f(\boldsymbol{x}) = \sum_{i=1}^{n-1} \max\left\{x_i^2 + x_{i+1}^2,\, (2-x_i)^2 + (2-x_{i+1})^2,\, 2e^{-x_i + x_{i+1}}\right\},$$
$$\boldsymbol{x}_0 = 2 \cdot \mathbf{1}, \quad f_{\text{opt}} = 2(n-1)$$

CCB3B: Chained CB3 II in Haarala, Miettinen, and Mäkelä [HMM04].

$$f(\boldsymbol{x}) = \max\left\{\sum_{i=1}^{n-1} x_i^2 + x_{i+1}^2,\, \sum_{i=1}^{n-1} (2-x_i)^2 + (2-x_{i+1})^2,\, \sum_{i=1}^{n-1} 2e^{-x_i + x_{i+1}}\right\},$$
$$\boldsymbol{x}_0 = 2 \cdot \mathbf{1}, \quad f_{\text{opt}} = 2(n-1)$$

SPL: Sparse Piecewise-Linear, the function (63) in Nesterov and Shikhman [NS15].

$$f(\boldsymbol{x}) = \max\left\{|x_1|,\ \max_{i\in[\![2,\,n]\!]}|x_i - 2x_{i-1}|\right\}, \quad \boldsymbol{x}_0 = \mathbf{1}, \quad f_{\text{opt}} = 0$$

The next two problems are classic sparse linear models for regression and classification. The parameters $m \in \mathbb{Z}_+$, $n \in \mathbb{Z}_+$, and $s \in (0,\,1)$ denote the number of samples, the number of features, and the sparsity of the solution, respectively. LLC has an additional parameter $k \in \mathbb{Z}_+$ for the number of classes. We generate the problem instances by sampling m and n uniformly from $[\![8,\,2048]\!]$ and k uniformly from $[\![3,\,10]\!]$ while rejecting those with $mn > 1024^2$ for LLR and $mnk > 5 \times 1024^2$ for LLC. We sample λ log-uniformly from $[10^{-4},\,0.1]$. We set the sparsity $s = 0.05$. Data matrices are sampled according to the distributions described below.

We use $\mathcal{B}(s;\,n)$ to denote a random vector in $\mathbb{R}^n$ whose entries are independent Bernoulli distributions with probability s. We use $\mathcal{U}(\Delta_n)$ to denote the uniform distribution over Δ_n, which can be sampled by normalizing n independent samples from an exponential distribution to have a unit ℓ_1-norm [Dev13].

LLR: Lasso Linear Regression.

$$f(\boldsymbol{x}) = \frac{1}{2m}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{x}\|_1, \quad \boldsymbol{x}_0 = \mathbf{0}$$
$$\text{where } \boldsymbol{A} \sim \mathcal{N}(\mathbf{0}_{m\times n},\,\boldsymbol{I}), \quad \boldsymbol{b} \coloneqq \boldsymbol{b}_0 + \boldsymbol{b}_N, \quad \boldsymbol{b}_0 \coloneqq \boldsymbol{A}(\boldsymbol{x}_T \odot \boldsymbol{x}_M),$$
$$\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}_n,\,\boldsymbol{I}), \quad \boldsymbol{x}_M \sim \mathcal{B}(s;\,n), \quad \boldsymbol{b}_N \sim \mathcal{N}\left(\mathbf{0}_m,\,\frac{0.05\|\boldsymbol{b}_0\|_1}{m}\boldsymbol{I}\right)$$

LLC: Lasso Linear Classification.

$$f(\boldsymbol{x}) = \frac{1}{m}\sum_{i=1}^{m} -\log\left(\frac{\exp\left(\boldsymbol{A}_i\boldsymbol{x}_{b_i}^\intercal\right)}{\sum_{j=1}^{k}\exp\left(\boldsymbol{A}_i\boldsymbol{x}_j^\intercal\right)}\right) + \lambda\|\boldsymbol{x}\|_1, \quad \boldsymbol{x} \in \mathbb{R}^{k\times n}, \quad \boldsymbol{x}_0 = \mathbf{0}$$
$$\text{where } \boldsymbol{A} \in \mathbb{R}^{m\times n},\ \boldsymbol{b} \in [\![1,\,k]\!]^m$$

To generate the data matrices for LLC, we first sample b_i uniformly from $[\![1,\,k]\!]$ for $i \in [\![1,\,m]\!]$. Then we generate a solution matrix $\boldsymbol{X} \in \mathbb{R}^{k\times n}$ by sampling each row $\boldsymbol{X}_i$ from $\mathcal{N}(\mathbf{0}_n,\,\boldsymbol{I}) \odot \mathcal{B}(s;\,n)$ while rejecting if its angle with any previous row is less than π/k. Next we generate a noisy coefficient matrix $\boldsymbol{C} \in \mathbb{R}^{m\times k}$ by setting $C_{ib_i} = 1$ and sampling C_{ij} for $j \neq b_i$ from $0.1\mathcal{U}(\Delta_{k-1})$ for each $i \in [\![1,\,m]\!]$. Finally, we set $\boldsymbol{A} \coloneqq \bar{\boldsymbol{A}} + \boldsymbol{A}_N$ where $\bar{\boldsymbol{A}} \coloneqq \boldsymbol{C}\boldsymbol{X}$ and $\boldsymbol{A}_N \sim \mathcal{N}\left(\mathbf{0}_{m\times n},\,\left(\frac{0.05}{mn}\sum|\bar{\boldsymbol{A}}_{ij}|\right)\boldsymbol{I}\right)$.

The above problems are all unconstrained. Next we introduce a constrained one.

DG: Distance Game

$$f(\boldsymbol{x}) = \max_{i\in[\![1,\,m]\!]}\sum_{j=1}^{3} f_i^{(j)}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Delta_n, \quad \boldsymbol{x}_0 = \frac{1}{n}\mathbf{1}$$
$$\text{where } f_i^{(1)}(\boldsymbol{x}) = |\boldsymbol{a}_i^\intercal\boldsymbol{x}|, \quad f_i^{(2)}(\boldsymbol{x}) = \|\boldsymbol{B}_i\boldsymbol{x}\|_2, \quad f_i^{(3)}(\boldsymbol{x}) \coloneqq \sum_k x_k \log\frac{x_k + \epsilon}{p_{ik} + \epsilon},$$
$$\boldsymbol{a}_i \in \mathbb{R}^n, \quad \boldsymbol{B}_i \in \mathbb{R}^{k\times n}, \quad \epsilon = 10^{-8}, \quad \boldsymbol{p}_i \in \Delta_n$$

The functions $f_i^{(j)}(\cdot)$ are three different convex distance functions (note that $f_i^{(3)}(\cdot)$ is the Kullback-Leibler divergence, which is not a metric). The parameters $m \in \mathbb{Z}_+$ and $n \in \mathbb{Z}_+$ control the scale of the problem. We generate $\boldsymbol{a}_i$ and $\boldsymbol{B}_i$ by sampling from the standard normal distributions. We set $k = \lfloor (n+3)/4 \rfloor$ so that Δ_n contains a nonsmooth point of $f_i^{(2)}(\cdot)$ with high probability ($\mathbb{P}\left[\min_{\boldsymbol{x} \in \Delta_n} f_i^{(2)}(\boldsymbol{x}) = 0\right]$ is 74% when $n = 10$ and 98% when $n = 30$ [Wen62]). We sample $\boldsymbol{p}_i$ from $\mathcal{U}(\Delta_n)$. Finally, we scale $\boldsymbol{a}_i$ and $\boldsymbol{B}_i$ so that the median values of the sets $S_j \coloneqq \left\{ f_i^{(j)}(\boldsymbol{x}_0) \;\middle|\; i \in [\![1, m]\!] \right\}$ are the same for $j \in \{1, 2, 3\}$. We generate the problem instances by sampling m and n from $[\![8, 1024]\!]$ while rejecting those with $mn > 400^2$.

7.3.2 Applying the TRAFS algorithm

The functional subdifferentials of all benchmark problems can be derived in closed form using rules given in Section 7.1.2. Below we outline how to solve the minimax problem (7.28) for each problem.

Three problems have closed-form solutions. The functional subdifferential of MAXQ is a polytope for which (7.30) can be written as $-\eta_k \min_{\boldsymbol{p} \in \Delta_d} \sqrt{\sum_{i=1}^d (a_i p_i)^2}$ where $a_i = 2x_{k_i}$ for indices $\{k_1, \cdots, k_d\}$ defined by Proposition 7.13. The solution is $p_i^* = \left(a_i^2 \sum_{j=1}^d a_j^{-2} \right)^{-1}$. For LLR and LLC, their functional subdifferential is a box $\{\boldsymbol{g} \mid \boldsymbol{u} \le \boldsymbol{g} \le \boldsymbol{v}\}$, which gives a solution to (7.30) as $g_i^* = \operatorname{argmin}^*_{g \in [u_i, v_i]} |g|$.

The functional subdifferentials of SPL, DPL, CLQ, CCB3A, and CCB3B can be represented as polytopes or sums of polytopes. We use (7.29) for DPL, CLQ, and CCB3A. For SPL, we use (7.29) when $n > 100$ for faster speed and (7.31) when $n \le 100$ for better numerical stability. For CCB3B, we use (7.30) since the polytope has at most three vertices.

The remaining problem is DG. We use Propositions 7.10 and 7.28 to compute the almost-functional subdifferential for $f_i^{(2)}(\cdot)$ in DG. Let V be the set of vertices of $\partial_\epsilon f(\boldsymbol{x}_k)$ for DG. We have $V = \{\boldsymbol{g}_t + \boldsymbol{M}_t^\intercal \boldsymbol{y}_t \mid t \in [\![1, T]\!], \|\boldsymbol{y}_t\|_2 \le 1\}$, where $T \le 2m$, $\boldsymbol{M}_t = \boldsymbol{B}_i^\intercal$ or $\boldsymbol{M}_t = \boldsymbol{0}$ depending on whether $\|\boldsymbol{B}_i \boldsymbol{x}\|_2 < \epsilon_t^{(2)}/2$, and $\boldsymbol{g}_t$ is the gradient of the smooth part of $\sum_j f_i^{(j)}(\boldsymbol{x})$. Note that if $|\boldsymbol{a}_i^\intercal \boldsymbol{x}| < \epsilon_t^{(1)}/2$, then two vertices are added, with $\boldsymbol{g}_t$ containing $\boldsymbol{a}_i$ and $\boldsymbol{g}_{t+1}$ containing $-\boldsymbol{a}_i$. Since $\max_{\boldsymbol{y}:\, \|\boldsymbol{y}\|_2 \le 1} \boldsymbol{d}^\intercal (\boldsymbol{g}_t + \boldsymbol{M}_t^\intercal \boldsymbol{y}) = \boldsymbol{g}_t^\intercal \boldsymbol{d} + \|\boldsymbol{M}_t \boldsymbol{d}\|_2$ holds for any $\boldsymbol{d} \in \mathbb{R}^n$, we formulate (7.29) as a SOCP with constraints $\boldsymbol{x}_k + \boldsymbol{d} \in \Delta_n$, $\|\boldsymbol{d}\|_2 \le \eta_k$, and $\|\boldsymbol{M}_t^\intercal \boldsymbol{d}\|_2 \le u - \boldsymbol{g}_t^\intercal \boldsymbol{d}$.

7.3.3 Comparison methods and implementation details

This section compares the TRAFS algorithm with the following methods:

- **GD**: The projected subgradient descent method that computes $\boldsymbol{x}_{k+1} = \Pi_C(\boldsymbol{x}_k - \eta_k \boldsymbol{g}_k)$, where $\eta_k \in \mathbb{R}_+$ is the step size, $\boldsymbol{g}_k \in \partial f(\boldsymbol{x}_k)$ is an arbitrary subgradient, and $\Pi_C(\boldsymbol{x}) \coloneqq \operatorname{argmin}^*_{\boldsymbol{y} \in C} \|\boldsymbol{y} - \boldsymbol{x}\|_2$ is the projection operator onto C. Setting $\eta_k = \frac{R}{L\sqrt{k}}$ yields an ergodic convergence rate of $\mathcal{O}(\epsilon^{-2})$ where R is the diameter of C and L is the Lipschitz constant of $f(\cdot)$ [Bub+15].

- **Bundle**: The proximal bundle method [MKW16], which is one of the fastest and most numerically stable methods among popular variants of bundle methods in a previous evaluation [BKM14, Chapter 17].

- **SA$_2$**: The subgradient method with double simple averaging, which provides convergence guarantees for the whole sequence of iterates (in contrast to GD that only guarantees ergodic convergence). SA$_2$ has demonstrated better performance than GD on SPL [NS15]. We test it on the unconstrained problems (problems other than DG) since applying it to the constrained case requires solving a nontrivial quadratic program at each iteration.

- **ISTA** and **FISTA**: Iterative Shrinkage-Thresholding Algorithm (ISTA) and Fast ISTA (FISTA) are proximal gradient methods to solve problems in the form $f(\boldsymbol{x}) = g(\boldsymbol{x}) + h(\boldsymbol{x})$ where $g(\cdot)$ is smooth and $h(\cdot)$ must be simple enough to admit a closed-form solution to the proximal operator. FISTA achieves $\mathcal{O}(\epsilon^{-0.5})$ convergence rate when $g(\cdot)$ is Lipschitz (possibly not strongly convex) by incorporating a momentum term. Among our benchmark problems, only LLR and LLC can be solved by ISTA and FISTA. We use the adaptive versions described in Beck and Teboulle [BT09].

Only TRAFS, Bundle, and GD are applicable to all of our benchmark problems. For Bundle, we use the MPBNGC Fortran implementation [Mäk03] with its Julia interface [Mil23]. For methods other than Bundle, we implement them in Python with numpy. TRAFS relies on external solvers to solve the QP and SOCP problems derived from (7.28). We use the open-source Clarabel [GC21] solver for SOCP and the PIQP [Sch+23] solver for QP for problems other than DPL and DG. For DPL and DG, we use the commercial Mosek solver since it is significantly faster than Clarabel on the two problems. Of note, Clarabel generates more accurate solutions than Mosek in our experiments.

We do not perform any problem-specific tuning for TRAFS, Bundle, ISTA, and FISTA. We use the default hyperparameters for Bundle except that we set the maximum line search iterations to 100 and the maximum number of stored subgradients to 50. We set the step growth parameter as 1.5 in ISTA and FISTA. For GD and SA$_2$, we use the step size $\eta_k = \frac{\eta_0}{\sqrt{k}}$ where $\eta_0 = \frac{\sqrt{n}}{L}$ for SPL, DPL, and LLR with L being the Lipschitz constant, $\eta_0 = 0.01$ for CCB3A and CCB3B, $\eta_0 = 10^{-4}$ for DG, and $\eta_0 = 1$ for others.

Our benchmark environment is a Linux workstation with an AMD Ryzen Threadripper 2970WX 24-core processor and 128 GiB of RAM. We use Python 3.11.6, numpy 1.26.2, openblas 0.3.25, Mosek 10.1.21, Clarabel 0.6.0, and PIQP 0.2.4. All methods and external solvers use a single thread. We set the maximum number of iterations to $50,000$ for all methods. For TRAFS and Bundle, we set the termination threshold of solution accuracy as 10^{-6}. All methods use double-precision floating-point numbers. The source code of experiments is available at https://github.com/jia-kai/trafs.

7.3.4 Metrics and results

For a minimization method and a problem instance, we define its error as

$$E \coloneqq \frac{f_{\text{meth}} - f_{\text{opt}}}{1 + |f_{\text{opt}}|}, \tag{7.32}$$

where f_{meth} is the objective value achieved by the method and f_{opt} is the optimal objective value (listed in Section 7.3.1 for the first six problems) or the best objective value found by any method (for LLR, LLC, and DG).

For each problem class and each method, we evaluate the following metrics:

1. For $\epsilon \in \{10^{-3}, 10^{-6}\}$, we evaluate the number of iterations and CPU time needed to achieve $E \leq \epsilon$. We normalize the metrics for each problem instance by the best method on that instance. We then summarize the metrics of this method by computing the geometric mean of the normalized metrics over problem instances on which the method achieves $E \leq \epsilon$. We also report the proportion of problem instances on which the method achieves $E \leq \epsilon$.

2. We report the arithmetic mean of numbers of iterations and CPU time over all problem instances before the method terminates, which could be due to reaching the maximum number of iterations of $50,000$, satisfying the user-defined accuracy of 10^{-6}, or encountering numerical issues. We also report the shifted geometric mean of the final error E over all problem instances, defined as below:

$$\text{SHM}(E_1, \cdots, E_n) \coloneqq \exp\left(\frac{1}{n}\sum_{i=1}^{n} \ln\left(E_i + s\right)\right) - s, \quad \text{where } s = 10^{-6} \tag{7.33}$$

Table 7.1 presents our evaluation results. TRAFS successfully solves all problem instances to $\epsilon = 10^{-3}$ accuracy. Under the setting of $\epsilon = 10^{-6}$, TRAFS solves 98.9% of the problem instances, compared to 49.3% of Bundle and 3.3% of GD. On problems other than LLR with $\epsilon = 10^{-6}$, TRAFS is the fastest method and successfully solves all instances; on LLR, TRAFS is slower than FISTA which is a more specialized method with better convergence guarantees for non-strongly convex problems. When compared to Bundle, the second-best method in terms of the number of successfully solved instances over all problem classes, TRAFS is 18.2 times faster on instances solved by both to $\epsilon = 10^{-3}$ and 39.1 times faster on instances solved by both to $\epsilon = 10^{-6}$.

Table 7.1: Experiment results

Problem	Method	$\epsilon = 10^{-3}$ Iter[a]	$\epsilon = 10^{-3}$ Time[a]	$\epsilon = 10^{-3}$ Solved[b]	$\epsilon = 10^{-6}$ Iter[a]	$\epsilon = 10^{-6}$ Time[a]	$\epsilon = 10^{-6}$ Solved[b]	Termination Iter[c]	Termination Time[c]	Termination Error[d]
MAXQ	TRAFS	**1.00**	**1.02**	**100%**	**1.00**	1.01	**100%**	146	0.05	$\mathbf{1.1\times10^{-10}}$
	Bundle	44.21	449.45	12%	5.98	158.94	4%	45607	1282.30	9.6×10^{3}
	GD	44.39	7.39	4%	7.82	**1.00**	2%	50000	1.47	1.2×10^{5}
	SA_2	-	-	0%	-	-	0%	50000	1.78	6.6×10^{5}
DPL	TRAFS	**1.00**	**1.51**	**100%**	**1.00**	**1.00**	**100%**	77	27.19	$\mathbf{7.2\times10^{-7}}$
	Bundle	24.10	1.68	**100%**	-	-	0%	1891	11.80	3.7×10^{-5}
	GD	-	-	0%	-	-	0%	50000	50.02	0.07
	SA_2	572.56	3.02	2%	-	-	0%	50000	50.24	0.25
CLQ	TRAFS	**1.18**	43.58	**100%**	**1.01**	**1.00**	**100%**	645	9.89	$\mathbf{7.6\times10^{-10}}$
	Bundle	1.50	254.18	**100%**	93.90	47.01	**100%**	46016	282.72	3.7×10^{-7}
	GD	868.94	45.74	8%	-	-	0%	50000	3.74	1.2×10^{-3}
	SA_2	4.00	**1.00**	**100%**	-	-	0%	50000	4.02	2.9×10^{-5}
CCB3A	TRAFS	1.78	17.21	**100%**	**1.00**	**1.00**	**100%**	137	2.90	$\mathbf{7.2\times10^{-10}}$
	Bundle	**1.08**	50.63	**100%**	74.29	30.28	**100%**	47645	352.36	1.8×10^{-7}
	GD	16.97	**1.03**	**100%**	-	-	0%	50000	5.74	1.1×10^{-4}
	SA_2	52.48	3.32	**100%**	-	-	0%	50000	5.93	8.6×10^{-5}
CCB3B	TRAFS	1.64	2.52	**100%**	2.10	**1.00**	**100%**	95	0.24	$\mathbf{1.1\times10^{-13}}$
	Bundle	**1.03**	107.50	**100%**	**1.00**	19.59	**100%**	39	1.65	3.5×10^{-10}
	GD	15.82	**1.04**	**100%**	-	-	0%	50000	6.25	6.7×10^{-5}
	SA_2	39.49	2.99	**100%**	-	-	0%	50000	6.63	5.5×10^{-5}
SPL	TRAFS	**1.00**	**1.00**	**100%**	**1.00**	**1.00**	**100%**	7	0.09	$\mathbf{1.3\times10^{-7}}$
	Bundle	5110.32	3751.44	60%	3041.50	2058.91	46%	37536	1021.29	2.4×10^{-4}
	GD	-	-	0%	-	-	0%	50000	14.94	0.21
	SA_2	-	-	0%	-	-	0%	50000	15.31	0.06
LLR	TRAFS	3.50	13.87	**100%**	3.66	15.10	90%	12914	10.94	4.5×10^{-7}*
	Bundle	3.73	258.57	**100%**	8.25	352.04	58%	13171	134.88	4.8×10^{-6}*
	GD	40.67	15.94	50%	39.40	17.43	28%	50000	9.78	3.3×10^{-4}*
	SA_2	897.55	327.85	34%	-	-	0%	50000	9.91	4.4×10^{-3}*
	ISTA	5.62	4.64	96%	3.18	2.66	80%	46344	13.20	1.8×10^{-6}*
	FISTA	**1.04**	**1.02**	**100%**	**1.10**	**1.04**	**100%**	11093	2.25	$\mathbf{2.0\times10^{-12}}$*
LLC	TRAFS	**1.12**	1.98	**100%**	**1.04**	**1.48**	**100%**	1412	7.31	$\mathbf{9.2\times10^{-11}}$*
	Bundle	30.44	263.16	98%	-	-	0%	41253	1864.14	1.5×10^{-4}*
	GD	276.10	126.48	8%	-	-	0%	50000	51.33	4.4×10^{-3}*
	SA_2	352.78	227.48	8%	-	-	0%	50000	50.87	2.8×10^{-3}*
	ISTA	77.68	37.68	88%	23.71	15.64	22%	50000	71.66	5.3×10^{-5}*
	FISTA	2.66	**1.26**	**100%**	3.90	1.53	**100%**	37156	58.70	3.3×10^{-10}*
DG	TRAFS	**1.00**	**1.00**	**100%**	**1.00**	**1.00**	**100%**	180	16.91	**0***
	Bundle	56.83	9.26	98%	8.97	2.45	36%	34648	234.19	1.9×10^{-5}*
	GD	435.84	14.20	60%	-	-	0%	50000	150.13	8.1×10^{-4}*
All	TRAFS	**1.18**	**1.61**	**100.0%**	**1.10**	**1.01**	**98.9%**	1735	8.39	$\mathbf{1.2\times10^{-7}}$*
	Bundle	8.94	31.70	85.3%	24.25	39.65	49.3%	29756	576.15	1.1×10^{-4}*
	GD	37.80	2.26	36.7%	30.86	3.00	3.3%	50000	32.60	0.01*

[a] Geometric mean of metrics normalized by the per-instance best result over instances successfully solved by the target method. Lower is better.
[b] Proportion of problem instances successfully solved by the target method. Higher is better.
[c] Arithmetic mean of metrics over all problem instances. Time is in seconds. Lower is better. Only TRAFS and Bundle support termination by user-defined accuracy.
[d] Shifted geometric mean (see (7.33)) of the final solution error when the method terminates. Lower is better.
* The true minimum objective value is unknown; f_{opt} is the best result among the evaluated methods.
** Bold numbers indicate the best method under the metric for each problem class.

7.4 Related work

There is a large body of work on smooth and nonsmooth convex optimization [BV04; Bub+15; Bag+20]. This section does not aim to provide a comprehensive review of all related work but rather focuses on the methods and results most relevant to TRAFS. I will discuss three classes of methods:

1. Methods that use subgradient oracles for unstructured problems. They typically deliver slower convergence rate than TRAFS.

2. Methods that exploit problem structures to achieve faster convergence on certain nonsmooth problems. Most of them have the same convergence rate as TRAFS; FISTA has a better theoretical rate than TRAFS.

3. Interior-point methods. Such methods are used to solve the minimax subproblems in TRAFS iterations. I compare them with first-order methods like TRAFS.

7.4.1 Subgradient oracle methods for unstructured problems

There are two classic methods that use the subgradient oracle. The projected subgradient descent iteratively updates the solution by moving in the opposite direction of an arbitrary subgradient of the objective function and projecting the result onto the feasible set. With proper parameters, the projected subgradient descent achieves an ergodic convergence rate (i.e., convergence of the average of all iterates) of $\mathcal{O}(\epsilon^{-2})$ [Bub+15]. The bundle method [Lem78; Mif82; HL96b] approximates the objective function at the k-th iteration by a piecewise affine function $\tilde{f}_k(\boldsymbol{x}) \coloneqq \max_{j \in J_k} \left(f(\boldsymbol{x}_j) + \boldsymbol{g}_j^\intercal (\boldsymbol{x} - \boldsymbol{x}_j)\right)$ where $J_k \subseteq [\![1, k]\!]$. Different variants may employ different strategies of defining J_k or computing $\boldsymbol{x}_{k+1}$ from $\tilde{f}_k(\cdot)$ [LNN95; Kiw06; MKW16]. It was recently shown that the proximal bundle method has iteration complexities of $\mathcal{O}(\epsilon^{-2})$ or $\mathcal{O}(\epsilon^{-1})$ for Lipschitz and strongly convex Lipschitz functions, respectively [DG23]. Compared to these two methods, TRAFS provides both better iteration complexity guarantees and substantially better performance on the benchmark problems (Section 7.3.4).

Any method that queries a subgradient oracle once per iteration (thus including the projected subgradient descent and the bundle method) needs at least $\Omega(\epsilon^{-2})$ or $\Omega(\epsilon^{-1})$ iterations in the worst case, depending on whether the function is strongly convex [NY83; IN14]. The same lower bounds also hold in the stochastic setting where the oracle delivers a subgradient with some zero-mean, bounded-variance additive noise, which can be proven from an information-theoretic perspective [Aga+09]. By contrast, TRAFS uses the functional subdifferential that includes multiple subgradients and incorporates sufficient local information to achieve a lower iteration complexity.

There are attempts to utilize the ϵ-subdifferential (see Remark 7.1 for a definition) for optimizing nonsmooth functions with special structures [BM71; CDW75]. Although rules exist to characterize the ϵ-subdifferentials mathematically [HL96b], computing the ϵ-subdifferential for unstructured functions is typically intractable [BLO02]. Gradient sampling methods [BLO02; BLO05; Bur+20] overcome this intractability to optimize unstructured functions by using gradients computed at m points uniformly sampled in $B_{\epsilon_k}[\boldsymbol{x}_k]$ in

each iteration to approximate the ϵ-subdifferential. However, gradient sampling methods are computationally expensive since most variants require $m \geq n+1$, where n is the dimension of the problem. Even the variant converging with a constant number of samples still recommends $m = n/10$ for practical performance [CQ13]. Previous evaluations typically work with small-scale problems with $n \leq 120$ [BLO05; CQ13; HSS17]. I am unaware of any global iteration complexity results for gradient sampling methods other than a local linear convergence result for a special class of functions [HSS17]. By contrast, the functional subdifferential deterministically characterizes the local behavior of the function to enable the guaranteed global convergence rates of TRAFS.

7.4.2 Structure-exploiting methods

A large body of work exploits problem structures to design efficient algorithms. Many algorithms incorporate proximal methods [PB+14] and/or primal-dual methods [CP11]. Let our goal be $\min_{\boldsymbol{x}\in C} f(\boldsymbol{x})$ as in TRAFS. If the objective function can be decomposed as $f(\boldsymbol{x}) = g(\boldsymbol{x}) + h(\boldsymbol{x})$ where $g(\cdot)$ is smooth and $h(\cdot)$ is nonsmooth but simple enough to admit a closed-form solution for the proximal operator defined as

$$\operatorname{prox}_{\mu h}(\boldsymbol{x}) \coloneqq \operatorname*{argmin}_{\boldsymbol{z}\in C}{}^{*} \left(h(\boldsymbol{z}) + \frac{1}{2\mu}\|\boldsymbol{z}-\boldsymbol{x}\|_2^2 \right),$$

then the proximal gradient method can be applied to achieve $\mathcal{O}(\epsilon^{-1})$ iteration complexity (and $\mathcal{O}\left(\epsilon^{-1/2}\right)$ complexity with acceleration [BT09]). For saddle-point problems defined as $f(\boldsymbol{x}) = \max_{\boldsymbol{y}\in D} g(\boldsymbol{x}, \boldsymbol{y})$ where $g(\cdot)$ is convex-concave and smooth, the mirror proxy algorithm [Nem04b] achieves $\mathcal{O}(\epsilon^{-1})$ iteration complexity. If the saddle-point problem has bilinear coupling with strongly-convex components, i.e., $g(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\intercal \boldsymbol{A}\boldsymbol{y} + \alpha(\boldsymbol{x}) - \beta(\boldsymbol{y})$ for some matrix $\boldsymbol{A}$ and strongly-convex and smooth functions $\alpha(\cdot)$ and $\beta(\cdot)$, then one can achieve $\mathcal{O}(\log \epsilon^{-1})$ iteration complexity with primal-dual methods [CP11]. Of note, for first-order methods on smooth problems, the lower bound is $\Omega\left(\epsilon^{-1/2}\right)$ [NY83]; Nesterov designed algorithms that achieve this lower bound [Nes83; Nes98]. Nestorov's accelerated method can be understood in a more unified primal-dual framework [AO17].

Instead of exploiting problem structures, TRAFS takes a complementary approach by utilizing more first-order information in each iteration. TRAFS does not incorporate ideas from proximal methods or primal-dual methods. For example, consider the objective function $f(\boldsymbol{x}) = g(\boldsymbol{x}) + \lambda\|\boldsymbol{x}\|_1$ where $g(\cdot)$ is convex and smooth. The proximal gradient method uses

the iterate

$$
\begin{aligned}
\boldsymbol{x}_{k+1} &= \operatorname*{argmin}_{\boldsymbol{z}\in\mathbb{R}^n}{}^{*} \left(\underbrace{g(\boldsymbol{x}_k) + \nabla^{\intercal} g(\boldsymbol{x}_k)(\boldsymbol{z}-\boldsymbol{x}_k) + \lambda\|\boldsymbol{z}\|_1}_{\text{approximating } f(\boldsymbol{z}) \text{ around } \boldsymbol{x}_k} + \underbrace{\frac{L_k}{2}\|\boldsymbol{z}-\boldsymbol{x}_k\|_2^2}_{\text{proximal term}} \right) \\
&= \operatorname*{argmin}_{\boldsymbol{z}\in\mathbb{R}^n}{}^{*} \left(\frac{L_k}{2}\left\|\boldsymbol{z} - \left(\boldsymbol{x}_k - \frac{1}{L_k}\nabla g(\boldsymbol{x}_k)\right)\right\|_2^2 + \lambda\|\boldsymbol{z}\|_1 \right) \\
&= \mathcal{T}_{\lambda/L_k}\left(\boldsymbol{x}_k - \frac{1}{L_k}\nabla g(\boldsymbol{x}_k)\right) \\
&\qquad \text{where } \mathcal{T}_\alpha(\boldsymbol{x}) \coloneqq \operatorname*{argmin}_{\boldsymbol{z}\in\mathbb{R}^n}{}^{*} \left(\alpha\|\boldsymbol{z}\|_1 + \frac{1}{2}\|\boldsymbol{z}-\boldsymbol{x}\|_2^2\right), \\
&\qquad\qquad \mathcal{T}_\alpha(\boldsymbol{x})_i = \max\{|x_i| - \alpha,\, 0\}\operatorname{sign}(x_i)
\end{aligned} \tag{7.34}
$$

By comparison, given Corollary 7.12, Proposition 7.15, and (7.30), TRAFS computes the iterate as

$$
\boldsymbol{g}_k = \operatorname*{argmin}_{\nabla g(\boldsymbol{x}_k)+\boldsymbol{l}_{\epsilon_k}(\boldsymbol{x}_k)\le\boldsymbol{z}\le\nabla g(\boldsymbol{x}_k)+\boldsymbol{u}_{\epsilon_k}(\boldsymbol{x}_k)}{}^{*} \|\boldsymbol{z}\|_2 \tag{7.35}
$$

where $\le$ is defined element-wise, and $\boldsymbol{l}_{\epsilon_k}(\cdot)$ and $\boldsymbol{u}_{\epsilon_k}(\cdot)$ are given as $\boldsymbol{l}$ and $\boldsymbol{u}$ in Proposition 7.15. Clearly, (7.35) differs from (7.34). TRAFS also performs differently from the proximal gradient methods (ISTA and FISTA) in the numerical experiments (Section 7.3.4). An interesting future direction is to combine functional subdifferential with existing methods to further improve the efficiency or applicability of nonsmooth optimization algorithms.

7.4.3 Interior-point methods

As discussed in Section 7.2.6, external solvers may be needed to solve the minimax problem in TRAFS iterations. Such subproblems encountered in TRAFS are typically linear/quadratic programs or second-order cone programs, for which practical solvers often use variants of the Interior-Point Methods (IPM) to achieve linear convergence; see Nemirovski [Nem04a] for a comprehensive review of IPM. The benchmark problems considered in Section 7.3.1 can also be reformulated as conic convex programs that can be handled by some IPM solvers like Clarabel [GC21]. One may wonder why not directly solving the original problem using those solvers. There are two reasons of preferring a first-order iterative method like TRAFS over IPM in certain cases:

1. **Efficiency**: For a problem with n variables and m constraints, IPM's complexity is $\mathcal{O}\big(m^{1.5}n^2 \log \frac{\mathcal{V}}{\epsilon}\big)$ where $\mathcal{V}$ is a data-dependent parameter and ϵ is the desired accuracy. IPM's complexity has a quadratic dependence on n; this quadratic term comes from the Newton step (which is a second-order method that works with the Hessian) in each IPM iteration. For large problems, the quadratic complexity can become a bottleneck. By comparison, the complexity of first-order methods like TRAFS often has a (quasi-) linear dependence on n. For example, solving (7.35) is $\mathcal{O}(n \log n)$ since computing the functional subdifferential of the ℓ_1 norm involves sorting $|x_i|$.

If an external IPM solver is used in TRAFS, then the k-th TRAFS iteration has a complexity term of $\mathcal{O}(n_k^2)$ where n_k is data-dependent. For example, if the original problem is unconstrained and $\partial_{\epsilon_k} f(\boldsymbol{x}_k)$ is a polytope with v_k vertices, then the quadratic program resulting from (7.30) has $n_k = v_k$. Related to this, there are two interesting future research questions: (i) characterizing n_k under different assumptions of problem formulations / data distributions; and (ii) designing efficient approximate algorithms to solve the TRAFS minimax subproblem when n_k is large.

2. **Applicability**: IPM can only work with certain problems. For example, there is currently no solver that supports a cosine objective $f(x) = -\cos(x)$ subject to $x \in [-\pi/2, \pi/2]$. The reason is that IPM (at least conceptually) first transforms any convex problem to a standard form that uses a linear objective $\boldsymbol{c}^\intercal \boldsymbol{x}$ with a set of constraint functions in the form $g_i(\boldsymbol{x}) \leq 0$ for $i \in \mathcal{M}$. For each $g_i(\cdot)$, IPM needs a *barrier function* $G_i(\cdot)$ such that $G_i(\boldsymbol{x}) \to \infty$ as $\boldsymbol{x}$ approaches $\partial \{\boldsymbol{x} \mid g_i(\boldsymbol{x}) \leq 0\}$. IPM then transforms the constrained problem into a sequence of unconstrained problems parameterized by $t > 0$: $\min_{\boldsymbol{x}} t\boldsymbol{c}^\intercal \boldsymbol{x} + \sum_{i \in \mathcal{M}} G_i(\boldsymbol{x})$. A solution to the original problem is obtained by solving the sequence of unconstrained problems with $t \to \infty$. To achieve linear convergence, the barrier function must satisfy a smoothness property called *self-concordant*. However, for some analytical forms of constraints, there is no known corresponding self-concordant barrier function that admits polynomial-time evaluation of its gradient and Hessian.

   By contrast, first-order methods like TRAFS do not require a barrier function and can be applied to a larger family of convex objective functions. Working with the above cosine objective is trivial for TRAFS. For example, for the objective $f(x) = -\cos(x) + |x|$, the TRAFS minimax subproblem can be solved by (7.35) as discussed earlier.

Given the above discussion, for a convex problem, one needs to consider its formulation and scale to decide the most appropriate optimization method.

Chapter 8

Limited-strategy games

This chapter studies limited-strategy games where a player's capability level limits the set of strategies they can access as introduced in Section 1.4. Section 8.1 defines the limited-strategy game, where a player with a higher capability level has access to a larger strategy space. Section 8.2 instantiates the limited-strategy game model with the *distance-bounded network congestion game with default action* (DncDa), which was first introduced in the author's previous work Yang, Jia, and Rinard [YJR22]. Section 8.2.1 reviews the symmetric-capability DncDa model analyzed in Yang, Jia, and Rinard [YJR22], where all players have the same capability level. Section 8.2.2 defines the *mixed-capability DncDa model* (McDncDa) that extends the DncDa model by allowing players to have different capability levels. Section 8.2.3 studies a special class of the McDncDa called the *Mixed Gold and Mines Game* (MGMG) where the network has a parameterized structure with two players. Section 8.3 discusses related work.

8.1 The limited-strategy game model

This section defines the limited-strategy games. Informally, in a limited-strategy game, each player has a hierarchy of strategy spaces. Their capability level in a game instance determines the strategy space they can use.

Definition 8.1 (One-shot limited-strategy game)
A *one-shot limited-strategy game* (*OLS-Game* for short) with n players is a tuple $G = \big(\mathcal{N}, (b_i)_{i\in\mathcal{N}}, (\mathcal{L}^i_j)_{i\in\mathcal{N}, j\in[\![1,b_i]\!]}, (u_i)_{i\in\mathcal{N}}\big)$ where:

- $\mathcal{N} = [\![1, n]\!]$ is the set of players.
- $b_i \in \mathbb{Z}_+$ is the maximal capability level of player i.
- $\mathcal{L}^i_j$ is a finite strategy space of player i when they have capability j. The strategy spaces of a player must form a hierarchy: $\forall j \in [\![1, b_i - 1]\!] : \mathcal{L}^i_j \subseteq \mathcal{L}^i_{j+1}$. The value of all $(\mathcal{L}^i_j)_{i\in\mathcal{N}, j\in[\![1,b_i]\!]}$ is called the *strategy hierarchy* of G.
- $u_i : \mathcal{L}^1_{b_1} \times \cdots \times \mathcal{L}^n_{b_n} \mapsto \mathbb{R}$ is the payoff function of player i. ◁

Comparing to the limited-perception game (Definition 2.1) where all players share the payoff perception function, a limited-strategy game does not have a strategy space reduction function shared by all players because the player's strategy spaces are typically different. This section focuses on establishing the definitions of limited-strategy games. Analysis of properties in limited-strategy games is left for games with more specific structures as exemplified in Section 8.2.

Similar to the limited-perception games, an instance of a limited-strategy game is defined by the capability profile.

Definition 8.2 (Game play of the OLS-Game)
In an instance of an OLS-Game as defined in Definition 8.1, each player i has a capability level $c_i \in [\![1, b_i]\!]$. The tuple $(c_i)_{i\in\mathcal{N}}$ is called the *capability profile* of this instance. Players know $\mathcal{N}$ and $(u_i)_{i\in\mathcal{N}}$ as common knowledge. The player i knows their capability level c_i, but may or may not know the capability levels of other players. Each player i chooses a strategy $s_i \in \mathcal{L}^i_{c_i}$. The payoff of player i is $u_i(s_1, \ldots, s_n)$. ◁

Similar to Definition 4.1, one can define capability-aware players in the limited-strategy games:

Definition 8.3 (Capability-aware player and fully-capability-aware game)
In an OLS-Game as defined in Definition 8.2, a *capability-aware* player knows everyone's capabilities $(c_j)_{j\in\mathcal{N}}$ in any game instance. In a *fully-capability-aware* OLS-Game, all players are capability-aware and know this as common knowledge. ◁

Similar to the pragmatically capability-positive property for limited-strategy games defined in Definition 4.25, we can define *capability-positive* limited-strategy games in the fully-capability-aware setting. The definition is easier than the limited-perception games because players do not need to reason about possible new perceptions of the game after their capability level increases.

Definition 8.4 (Capability-positive OLS-Game)
Given an OLS-Game as defined in Definition 8.1, assume the game is fully-capability-aware. Let $P_i(c_1, \ldots, c_n) \subset \mathbb{R}$ denote the set of possible payoffs of player i in Nash equilibria when the players' capability profile is $(c_1, \ldots, c_n)$.

The game is *capability-positive* for a capability profile $\boldsymbol{c} \in \prod_{i \in \mathcal{N}} [\![1, b_i]\!]$ if and only if for all $i \in \mathcal{N}$, it holds that

$$(c_i < b_i) \implies (\min P_i(c_i + 1, \boldsymbol{c}_{-i}) \geq \max P_i(c_i, \boldsymbol{c}_{-i}))$$

The game is *universally capability-positive* if it is capability-positive for all possible capability profiles. ◁

Not all limited-strategy games are capability-positive. Example 8.1 presents an example where a player's payoff changes arbitrarily as their capability level increases by one.

Example 8.1 (Arbitrary payoff change with increasing capability)
Consider a two-player two-action bimatrix game. Player 1 is the row player with two possible capability levels: $\mathcal{L}_1^1 = \{1\}$ and $\mathcal{L}_2^1 = \{1, 2\}$. Player 2, the column player, has one capability level: $\mathcal{L}_1^2 = \{1, 2\}$. Let $C \in \mathbb{R}$ be a constant. Their payoff matrices are:

$$\boldsymbol{A} \coloneqq \begin{bmatrix} 1 & C-1 \\ 2 & C \end{bmatrix}, \quad \boldsymbol{B} \coloneqq \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

When player 1 is has capability level 1, they can only use the first row, and player 2 plays the first column, which results in payoffs $(1, 2)$. When player 1 is allowed to use their full capability, the only Nash equilibrium is second row and second column, which results in payoffs $(C, 2)$. ◁

Unlike the limited-perception games, there is no truly capability-oblivious player in a limited-strategy game since a player at least knows their own capability level so that they can choose a valid strategy. When players are not fully capability-aware, a limited-strategy game can be modeled by existing incomplete-information game models. One of the simplest models is the Bayesian game, also called the Harsanyi model; see Harsanyi [Har67] and Maschler, Zamir, and Solan [MZS13, Section 9.4] for more details. Informally speaking, in a Bayesian game, each player has a private type that determines the player's strategy space and payoff function; players have a prior belief on the distribution of types of other players. In the context of limited-strategy games, the capability level of a player can be considered as their type. The *Bayesian limited-strategy game* is defined as follows.

Definition 8.5 (Bayesian limited-strategy game)
A *Bayesian limited-strategy game* is a tuple $G = \left(\mathcal{N}, (b_i)_{i \in \mathcal{N}}, (\mathcal{L}_j^i)_{i \in \mathcal{N}, j \in [\![1, b_i]\!]}, (u_i)_{i \in \mathcal{N}}, (p_i)_{i \in \mathcal{N}}\right)$ where $p_i \in \Delta_{\mathcal{C}}$ is player i's prior belief on the capability levels of other players with $\mathcal{C} \coloneqq \prod_{i \in \mathcal{N}} [\![1, b_i]\!]$. Other symbols are defined as in Definition 8.1.

In an instance of a Bayesian limited-strategy game, player i knows their capability level c_i. All players know $(p_i)_{i \in \mathcal{N}}$ as common knowledge. Player i chooses a strategy $s_i \in \mathcal{L}_{c_i}^i$. ◁

As in the Bayesian game, a player's pure strategy in a Bayesian limited-strategy is a function $s_i : [\![1, b_i]\!] \mapsto \mathcal{L}_{b_i}^i$ such that $s_i(c) \in \mathcal{L}_c^i$ for all $c \in [\![1, b_i]\!]$. Each player in a Bayesian limited-strategy game tries to maximize their expected payoff given their belief on the capability levels. Given a strategy profile $(s_j)_{j \in \mathcal{N}}$ where $s_j \in \left(\mathcal{L}_{b_j}^j\right)^{[\![1, b_j]\!]}$, the payoff of player i is their expected payoff according to their belief:

$$U_i(s_1, \ldots, s_n) \coloneqq \sum_{(c_1, \ldots, c_n) \in \mathcal{C}} u_i(s_1(c_1), \ldots, s_n(c_n)) p_i(c_1, \ldots, c_n) \tag{8.1}$$

Nash's theorem guarantees the existence of a mixed Nash equilibrium in a Bayesian limited-strategy game, where the mixed strategy of player i can be represented as a *behavior strategy* $\sigma_i : [\![1, b_i]\!] \mapsto \Delta_{\mathcal{L}^i_{b_i}}$ [MZS13, Theorem 9.47]. Since the Bayesian limited-strategy game is a special case of the Bayesian game, previous results on Bayesian games apply. In particular, if the normal-form game $\left(\mathcal{N}, \left(\mathcal{L}^i_{b_i}\right)_{i\in\mathcal{N}}, (u_i)_{i\in\mathcal{N}}\right)$ is a potential game and all players share a common belief (i.e., $p_i = p_j$ for all $i, j \in \mathcal{N}$), then the Bayesian limited-strategy game is also a potential game and thus has a pure Nash equilibrium [Heu+96, Corollary 5.4].

Since the prior belief in a Bayesian limited-strategy game heavily influences the game outcome, the rest of this chapter focuses on the simpler model of fully-capability-aware limited-strategy games.

8.2 Distance-bounded network congestion games

This section demonstrates a generic idea of building a strategy hierarchy to define a limited-strategy game. The idea is to define a strategy space as the set of programs in a Domain-Specific Language (DSL). A player's capability level determines the size of strategy programs accessible to the player.

I focus on variants of *distance-bounded network congestion games with default action* (DNCDA), which were first proposed in the author's earlier work [YJR22]. A player's strategy in DNCDA is represented by a program in a DSL that generates a path in a network. DNCDA is derived from *distance-bounded network congestion games* (DNC). DNC is inspired by classic network congestion games. In a network congestion game, players choose paths in a network to minimize their travel time from a source to a destination. The travel time is the sum of the travel time on each edge, which depends on the number of players using the edge. Network congestion games are a special case of *congestion games*, which are equivalent to *potential games* and have pure Nash equilibria.

Before introducing DNC and DNCDA games, let's first review potential games and congestion games.

Definition 8.6 (Potential game [MS96c])
A *potential game* is a tuple $G = \left(\mathcal{N}, (\mathcal{S}_i)_{i\in\mathcal{N}}, (u_i)_{i\in\mathcal{N}}, \Phi\right)$ where:

- $\mathcal{N} = [\![1, n]\!]$ is the set of players.
- $\mathcal{S}_i$ is the strategy space of player i.
- $u_i : \mathcal{S} \mapsto \mathbb{R}$ is the payoff function of player i, where $\mathcal{S} \coloneqq \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$ is the joint strategy space.
- $\Phi : \mathcal{S} \mapsto \mathbb{R}$ is the potential function of the game that satisfies the following condition:

$$\forall \boldsymbol{s} \in \mathcal{S}, \forall i \in \mathcal{N}, \forall s'_i \in \mathcal{S}_i : \ \Phi(\boldsymbol{s}) - \Phi(s'_i, \boldsymbol{s}_{-i}) = u_i(\boldsymbol{s}) - u_i(s'_i, \boldsymbol{s}_{-i}) \qquad \triangleleft$$

For a finite potential game, $\Phi(\cdot)$ is bounded. It is easy to verify that the maximizer of $\Phi(\cdot)$ is a pure Nash equilibrium of the game.

Definition 8.7 (Congestion game [Ros73])
A *congestion game* is a tuple $G = \left(\mathcal{N},\, \mathcal{E},\, (\mathcal{S}_i)_{i\in\mathcal{N}},\, (d_e)_{e\in\mathcal{E}}\right)$ where:

- $\mathcal{N} = [\![1,\, n]\!]$ is the set of players.
- $\mathcal{E}$ is a finite set of congestible resources.
- $\mathcal{S}_i \subset 2^{\mathcal{E}}$ is the strategy space of player i.
- $d_e : \mathbb{N} \mapsto \mathbb{R}$ is the delay function on the resource e, which is non-decreasing.

Given a strategy profile $\boldsymbol{s} \in \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$, the payoff of player i is defined as:

$$u_i(\boldsymbol{s}) \coloneqq -\sum_{e\in s_i} d_e(x_e(\boldsymbol{s}))$$
$$\text{where } x_e(\boldsymbol{s}) \coloneqq |\{j \in \mathcal{N} \mid e \in s_j\}| \qquad \triangleleft$$

It is easy to verify that for a congestion game, the function $\Phi(\boldsymbol{s}) \coloneqq -\sum_{e\in\mathcal{E}} \sum_{k=1}^{x_e(\boldsymbol{s})} d_e(k)$ is a potential function. Therefore, all congestion games are potential games. Monderer and Shapley [MS96c] also shows that every potential game can be reduced to a congestion game.

The complexity of finding a pure Nash equilibrium in a congestion game is PLS-complete [FPT04]. PLS stands for the *polynomial local search* complexity class, which is a subclass of NP. Informally, a problem is in PLS if a solution can be improved or certified to be locally optimal in polynomial time. There is no known polynomial-time algorithm to solve a PLS-complete problem.

A *network congestion game* is a special case of a congestion game where the resource set $\mathcal{E}$ is the edge set of a directed graph. The strategy space of each player is the set of paths from a source vertex to a sink vertex. A game is called *symmetric* if all players share the same source and sink vertices. Symmetric network congestion games and a special class of congestion games can be solved in polynomial time; asymmetric network congestion games are PLS-complete [FPT04; ARV08].

8.2.1 Symmetric-capability game models and properties

This section introduces symmetric-capability network congestion games and their properties. Most of the results have been published in Yang, Jia, and Rinard [YJR22].

8.2.1.1 DNC: The distance-bounded network congestion game

In a distance-bounded network congestion game, a player's capability level limits the length of the path they can choose. The game is defined as follows:

Definition 8.8 (Distance-bounded network congestion game)
A ***D**istance-bounded **N**etwork **C**ongestion game (DNC)* [YJR22, Definition 1] is defined by a tuple $G = (\mathcal{V},\, \mathcal{E},\, \mathcal{N},\, s,\, t,\, (d_e)_{e\in\mathcal{E}})$ where:

- $\mathcal{V}$ is the set of vertices in the network.

- $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges in the network.
- $\mathcal{N} = [\![1, n]\!]$ is the set of players.
- $s \in \mathcal{V}$ is the source vertex shared by all players.
- $t \in \mathcal{V}$ is the sink vertex shared by all players.
- $d_e : \mathbb{N} \mapsto \mathbb{R}$ is a non-decreasing delay function on edge e.

The source has no incoming edges and the sink has no outgoing edges; i.e., $(v, s) \notin \mathcal{E}$ and $(t, v) \notin \mathcal{E}$ for all $v \in \mathcal{V}$. There is at least one path from the source to the sink. The network has no negative-delay cycles, i.e., for each cycle $\mathcal{C}$, we require $\sum_{e \in \mathcal{C}} \min_{i \in \mathbb{Z}_+} d_e(i) = \sum_{e \in \mathcal{C}} d_e(1) \geq 0$.

Players in an instance of a DNC have a symmetric capability profile defined by a shared distance bound $b \in [\![1, \bar{b}]\!]$ where $\bar{b}$ is the length of the longest $s - t$ simple path in the network. The strategy space of a player contains all $s - t$ simple paths whose length does not exceed b:

$$\mathcal{L}_b \coloneqq \left\{ (p_0, \ldots, p_k) \;\middle|\; \begin{array}{l} p_0 = s,\, p_k = t,\, (p_i, p_{i+1}) \in \mathcal{E}, k \leq b, \\ p_i \neq p_j \text{ for } i \neq j \end{array} \right\} \qquad \triangleleft$$

The payoff of a player in a DNC instance is the negative delay they experience; i.e., a player's goal is to minimize their delay. Let $s_i \coloneqq (p_{i0}, \ldots, p_{ik_i}) \in \mathcal{L}_b$ denote the strategy of player i for $i \in \mathcal{N}$. Let $E_i \coloneqq \{(p_{ij}, p_{i,j+1}) \mid 0 \leq j < k_i\}$ denote the corresponding set of edges on the path chosen by player i. The *load* on an edge $e \in \mathcal{E}$ is defined as the number of players that occupy this edge: $x_e \coloneqq |\{i \mid e \in E_i\}|$. The delay experienced by player i is $d_i(\boldsymbol{s}) \coloneqq \sum_{e \in E_i} d_e(x_e)$ for a strategy profile $\boldsymbol{s} \coloneqq (s_1, \ldots, s_n)$. All players experience infinite delay if the distance bound permits no feasible solution (i.e., when $\mathcal{L}^b_= \emptyset$). *Social welfare* is defined as the negative total delay of all players where a larger welfare value means on average players experience less delay: $W(\boldsymbol{s}) \coloneqq -\sum_{i \in \mathcal{N}} d_i(\boldsymbol{s})$.

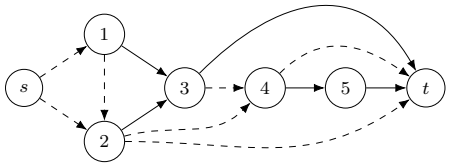
8.2.1.2 DncDa: A DNC variant with a DSL-based strategy space

DncDa is a variant of DNC to be introduced below, which allows the strategy space to be defined by programs in a DSL.

Definition 8.9 (Distance-bounded network congestion game with default action) A ***D**istance-bounded **N**etwork **C**ongestion game with **D**efault **A**ction* (DncDa) [YJR22, Definition 2] is a tuple $G = (\mathcal{V}, \mathcal{E}, \mathcal{N}, s, t, (d_e)_{e \in \mathcal{E}}, (w_e)_{e \in \mathcal{E}})$ where:

- $w_e \in \{0, 1\}$ is the length of edge e.
- All other symbols have the same meaning as in Definition 8.8.

The network has the following properties:



(a) Example graph structure. Solid arrows represent the default actions. Dashed arrows are unit-length edges.

```
if (u == s) { return 2; } else {
    if (u == 3) { return 4; } else {
        return DA(u);
    }
}
```

(b) The shortest program to represent the strategy $(s, 2, 3, 4, 5, t)$.

Figure 8.1: An example of the DncDa game and a program to represent a strategy.

- A default action, denoted as $\mathrm{DA}(\cdot)$, can be defined for every non-source, non-sink vertex $v \in \mathcal{V} \setminus \{s, t\}$ such that:

$$\Big(v, \mathrm{DA}(v)\Big) \in \mathcal{E}, \quad w_{(v, \mathrm{DA}(v))} = 0,$$
$$\forall u \in \mathcal{V} \setminus \{\mathrm{DA}(v)\} : \ (v, u) \in \mathcal{E} \implies w_{(v,u)} = 1$$

- Edges from the source have unit length: $\forall v \in \mathcal{V} : \ (s, v) \in \mathcal{E} \implies w_{(s,v)} = 1$
- The subgraph of zero-length edges is acyclic. Equivalently, starting from any non-source vertex, one can follow the default actions to reach the sink.

Players in an instance of a DNC have a symmetric capability profile defined by a shared distance bound $b \in \mathbb{Z}_+$. The strategy space of a player contains all $s - t$ simple paths whose length does not exceed b:

$$\mathcal{L}_b \coloneqq \left\{ (p_0, \ldots, p_k) \;\middle|\; \begin{array}{l} p_0 = s,\, p_k = t,\, (p_i, p_{i+1}) \in \mathcal{E}, \sum_{i=0}^{k-1} w_{(p_i, p_{i+1})} \le b, \\ p_i \neq p_j \text{ for } i \neq j \end{array} \right\} \qquad \triangleleft$$

The following context-free grammar [HMU14] defines the DSL that describes the strategy of a player in a DncDa instance:

$$\begin{aligned} \texttt{Program} &\to \textbf{return}\ \texttt{DA(}u\texttt{);} \\ &\quad \mid \textbf{if}\ \texttt{(}u\ \texttt{==}\ \texttt{V)}\ \texttt{\{}\textbf{return}\ \texttt{V;\}}\ \textbf{else}\ \texttt{\{Program\}} \\ \texttt{V} &\to v \in \mathcal{V} \end{aligned}$$

A program p in this DSL defines a computable function $f_p : \mathcal{V} \mapsto \mathcal{V}$ with semantics similar to the C language where the input vertex is stored in the variable u, as illustrated in Fig. 8.1. The strategy corresponding to the program p is a path $(v_0, \ldots, v_k)$ from s to t where:

$$v_0 = s \qquad v_{i+1} = f_p(v_i) \text{ for } i \ge 0 \text{ and } v_i \neq t \qquad k = i \text{ if } v_i = t$$

The capability level of a player in DncDa is defined as the maximum size of programs that they can use. The size of a program is the depth of its parse tree. Due to the properties of DncDa, the shortest program that encodes a path from s to t specifies the edge chosen at all divergent points in this path. The size of this program equals the length of the path. Hence the distance bound in the game configuration specifies the capability of each player in the game.

8.2.1.3 Properties of DNC and DncDa

Since the capability profile is symmetric in both DNC and DncDa instances, the games are fully-capability-aware as defined in Definition 8.3.

Computational complexity of game solving Yang, Jia, and Rinard [YJR22] has shown the following hardness results for both DNC and DncDa:

- Finding one pure Nash equilibrium in a DNC or DncDa instance is PLS-complete.
- Finding the pure Nash equilibrium with the best (resp. worst) social welfare (i.e., minimum (resp. maximum) total delay for all players) in a DNC or DncDa instance is NP-hard.
- Finding the pure strategy profile with the best social welfare where players behave cooperatively under a central authority in a DNC or DncDa instance is NP-hard.

As an interesting comparison, Fabrikant, Papadimitriou, and Talwar [FPT04] shows that finding a pure Nash equilibrium is in P for symmetric network congestion games but PLS-complete for asymmetric ones. A game is symmetric if and only if all players have the same strategy space, and the payoff of playing a particular strategy only depends on the set of strategies played by other players but not on who plays which strategy. In an asymmetric network congestion game, players may have different source or sink vertices. The DNC and DncDa games are both symmetric, but with the additional distance bound constraint, solving the game becomes PLS-complete.

The proof in Yang, Jia, and Rinard [YJR22] constructs hard games with multiple players. On the other hand, if a DNC game has only a few players, one pure Nash equilibrium can be found in polynomial time, as shown in the following theorem.

Theorem 8.1 (Solvable DNC with few players)
Given a DNC instance with n players, a pure Nash equilibrium can be found in $\mathcal{O}\big(b^n \left|\mathcal{V}\right|^{2n}\big)$.

Proof. We can minimize the potential function using dynamic programming. For $\boldsymbol{v} \in \mathcal{V}^n$ and $\boldsymbol{c} \in [\![1, b]\!]^n$, let $f(\boldsymbol{v}, \boldsymbol{c})$ be the minimum potential function value computed on strategy profiles $\boldsymbol{s}$ such that s_i is a length-c_i path from the source to v_i for $i \in \mathcal{N}$. Then $f(\boldsymbol{v}, \boldsymbol{c})$ can be recursively computed by considering all possible successors of $\boldsymbol{v}$. There are $(b\left|\mathcal{V}\right|)^n$ states and $\mathcal{O}(\left|\mathcal{V}\right|^n)$ transitions for each state. The total time complexity is thus $\mathcal{O}\big(b^n \left|\mathcal{V}\right|^{2n}\big)$. □

Capability transfer properties Since the capability profile in a DncDa instance is symmetric, a natural definition of capability transfer properties is to consider the change of the social welfare when the players' capability level increases.

Let's introduce some notation. Assuming the maximum capability level is $\bar{b}$ (which is the longest $s-t$ simple path in DncDa), we have $\mathcal{L}_b \subseteq \mathcal{L}_{b+1}$ for $1 \le b < \bar{b}$. Let $\mathrm{Equil}\,(b) \subseteq (\mathcal{L}_b)^n$ denote the set of all pure Nash equilibria at the capability level b. Define $W_b^+ \coloneqq \max_{\boldsymbol{s} \in \mathrm{Equil}\,(b)} W(\boldsymbol{s})$ to be the best social welfare at equilibrium and $W_b^- \coloneqq \min_{\boldsymbol{s} \in \mathrm{Equil}\,(b)} W(\boldsymbol{s})$ as the worst social welfare.

Due to the hardness of solving DncDa, Yang, Jia, and Rinard [YJR22] considers a special class of DncDa, called the *distance-bounded network congestion game with default action and shared delay* (DncDaS). In a DncDaS game, all edges share the same delay function $d(\cdot)$; i.e., $\forall e \in \mathcal{E} : d_e(\cdot) = d(\cdot)$. The delay function is also non-negative and non-decreasing. Yang, Jia, and Rinard [YJR22] proves the following capability transfer properties for DncDaS:

- Define *socially capability-positive* games as games where the social welfare at equilibrium never decreases as the capability level increases, i.e., $\forall 1 \leq b < \bar{b} : W_b^+ \leq W_{b+1}^-$. Then the sufficient and necessary condition on the delay function $d(\cdot)$ for all DncDaS games using $d(\cdot)$ to be socially capability-positive is that $d(\cdot)$ is a constant function.
- Define *socially max-capability-preferred* games as games where the worst social welfare at equilibrium with the maximum capability level is no worse than any social welfare at equilibrium with any lower capability level, i.e., $\forall 1 \leq b < \bar{b} : W_b^+ \leq W_{\bar{b}}^-$. Then the sufficient and necessary condition on the delay function $d(\cdot)$ for all DncDaS games using $d(\cdot)$ to be socially max-capability-preferred is that $d(\cdot)$ is a constant function.
- Define *socially capability-negative* games as games where the social welfare at equilibrium never increases as the capability level increases, i.e., $\forall 1 \leq b < \bar{b} : W_b^- \geq W_{b+1}^+$. Then the sufficient and necessary condition on the delay function $d(\cdot)$ for all DncDaS games using $d(\cdot)$ to be socially capability-negative is that $d(\cdot)$ is the zero function.
- Define *socially min-capability-preferred* games as games where the best social welfare at equilibrium with the minimum capability level is no worse than any social welfare at equilibrium with any higher capability level, i.e., $\forall b \geq 2 : W_1^- \geq W_b^+$. Then the sufficient and necessary condition on the delay function $d(\cdot)$ for all DncDaS games using $d(\cdot)$ to be socially min-capability-preferred is that $d(\cdot)$ is the zero function.

One interpretation of the above results is that there is no non-trivial delay function that guarantees any monotonicity of social welfare with respect to the capability level in DncDaS games. For all possible network configurations to satisfy any of the above capability transfer properties, the delay function must be either constant or zero, which implies the total delay experienced by players is independent of the capability level.

8.2.2 Mixed-capability game models and properties

This section extends the DncDa model to allow players with different capability levels. The new game is called *Mixed-capability DncDa* (McDncDa).

Definition 8.10 (Mixed-capability DncDa)
A *Mixed-capability DncDa* (McDncDa) is a DncDa game where the capability profile is not required to be symmetric. With the notation defined in Definition 8.9, in an instance of an McDncDa game, the capability profile is defined by $(c_i)_{i \in \mathcal{N}}$ where $c_i \in [\![1, \bar{b}]\!]$ is the capability level of player i, and $\bar{b}$ is the length of the longest $s-t$ simple path in the network. Player i's strategy space is all $s-t$ simple paths whose length does not exceed c_i. All players know the capability profile as common knowledge. ◁

Although the capability profile in an McDncDa instance can be asymmetric, the game is still symmetric in the sense that players with the same capability level have the same strategy space. Of note, McDncDa games are fully-capability-aware (Definition 8.3).

Theorem 8.2 (Hardness of solving McDncDa)
Finding a pure Nash equilibrium in an McDncDa instance is PLS*-complete. Given an n-player McDncDa instance, a pure Nash equilibrium can be found in $\mathcal{O}\big(\big(\prod_{i\in\mathcal{N}} c_i\big)\,|\mathcal{V}|^{2n}\big)$.*

Proof. Since McDncDa generalizes DncDa, finding one pure Nash equilibrium is at least PLS-hard. It is in PLS since the best response of any player can be computed by dynamic programming in polynomial time. Therefore, it is PLS-complete. The time complexity proof is similar to Theorem 8.1; of note, the state graph in the dynamic programming is acyclic since the default action edges lead to the sink and thus form an acyclic subgraph. □

A mixed-capability McDncDa is *capability-positive* if a player gets no longer delay in pure Nash equilibria when their capability level increases. The following definition is adapted from Definition 8.4:

Definition 8.11 (Capability-positive McDncDa)
Given a McDncDa game, let $D_i(c_1, \ldots, c_n) \subset \mathbb{R}$ denote the set of possible delays of player i in pure Nash equilibria when the players' capability profile is $(c_1, \ldots, c_n)$.

The game is *capability-positive* for a capability profile $\boldsymbol{c} \in [\![1, \bar{b}]\!]^n$ if and only if for all $i \in \mathcal{N}$, it holds that

$$\big(c_i < \bar{b}\big) \implies \big(\max D_i(c_i + 1, \boldsymbol{c}_{-i}) \leq \min D_i(c_i, \boldsymbol{c}_{-i})\big)$$

The game is *universally capability-positive* if it is capability-positive for all possible capability profiles. ◁

Below are a few theorems on the computational complexity of checking capability-positive properties in McDncDa games. Their results are informally summarized as follows:

- Computing the pure Nash equilibrium that is best or worst for a specific player is NP-hard for DNC, DncDa, and McDncDa games.
- Determining whether a McDncDa game is capability-positive for a given capability profile is coNP-complete.

Theorem 8.3 (Hardness of finding single-player best equilibrium in DNC)
In a DNC game with a symmetric capability profile as defined in Definition 8.8, let $\mathrm{Equil}\,(b) \subseteq (\mathcal{L}_b)^n$ *denote the set of all pure Nash equilibria at the capability level b. For a strategy profile $\boldsymbol{s} \in (\mathcal{L}_b)^n$, let $d_i(\boldsymbol{s})$ denote the delay experienced by player i.*

Given $i \in \mathcal{N}$, determining whether there exists $\boldsymbol{s} \in \mathrm{Equil}\,(b)$ such that $d_i(\boldsymbol{s}) \leq D$ for arbitrary $b \in [\![1, \bar{b}]\!]$ and $D \in \mathbb{R}$ is NP*-complete. Consequently, computing $\min_{\boldsymbol{s}\in\mathrm{Equil}\,(b)} d_i(\boldsymbol{s})$ is* NP*-hard.*

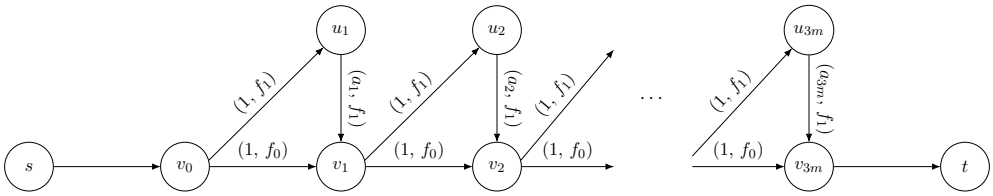


Figure 8.2: DNC game for the 3PARTITION reduction.
The label (i, f_j) on an edge indicates that the length of the edge is i, and the delay function is $f_j(\cdot)$. Non-unit-length edges can be implemented by a sequence of unit-length edges. Unlabelled edges have unit length and zero delay.

Proof. The NP-completeness argument follows from a reduction from the strongly NP-complete 3PARTITION problem [GJ79].

An instance of a 3PARTITION problem has $3m$ positive integers $(a_i)_{i\in[\![1,\,3m]\!]}$. Let $T \coloneqq \frac{1}{m}\sum_{i=1}^{3m} a_i$. The numbers satisfy $T/4 < a_i < T/2$. The question Q_1 is: Is there a partition $S_1, \ldots, S_m$ of $[\![1,\,3m]\!]$ such that $\sum_{j\in S_i} a_j = T$ for all $i \in [\![1,\,m]\!]$? Of note, if the answer to Q_1 is yes, then for each $i \in [\![1,\,m]\!]$, the set S_i has exactly three elements since $T/4 < a_i < T/2$. 3PARTITION is strongly NP-complete, which means that the problem remains NP-complete even when the numbers are encoded in unary. Therefore, the constructed DNC instance can be represented by $\mathsf{poly}(\sum_i a_i)$ bits.

Given a 3PARTITION instance, we construct a DNC instance as shown in Fig. 8.2. The DNC game has m players. The capability level is $b = T + 3m + 2$. There are two types of nonzero delay functions:

$$f_0(x) = \begin{cases} 0 & \text{if } x \le m-1 \\ 1 & \text{otherwise} \end{cases}, \quad f_1(x) = \begin{cases} 0 & \text{if } x = 1 \\ R & \text{otherwise} \end{cases},$$

where R is a sufficiently large constant so that in any pure Nash equilibrium, no player chooses an edge with a delay of R. It is easy to verify that $d_i(\boldsymbol{s}) \ge 0$ for $\boldsymbol{s} \in (\mathcal{L}_b)^n$ and $i \in [\![1,\,m]\!]$ since all delays are nonnegative.

Define the question Q_2: Does there exist $\boldsymbol{s} \in \mathrm{Equil}\,(b)$ such that $d_1(\boldsymbol{s}) \le 0$?

If the answer to Q_1 is yes, define the strategy profile $\boldsymbol{s}^*$ as follows: For each $i \in [\![1,\,m]\!]$, player i chooses $(v_{j-1},\, u_j,\, v_j)$ if $j \in S_i$, and $(v_{j-1},\, v_j)$ otherwise. One can verify that $d_i(\boldsymbol{s}^*) = 0$ for all $i \in [\![1,\,m]\!]$. Therefore, $\boldsymbol{s}^* \in \mathrm{Equil}\,(b)$, and the answer to Q_2 is yes.

If the answer to Q_2 is yes, the definition of $f_0(\cdot)$ implies that the load on edges $(v_{i-1},\, v_i)$ for $i \in [\![1,\,3m]\!]$ does not exceed $m-1$, which implies that $(v_{i-1},\, u_i,\, v_i)$ is covered by exactly one player. Since the distance bound $b = T + 3m + 2$, each player covers at most three $(u_i,\, v_i)$ edges. Since the m players cover the $3m$ $(u_i,\, v_i)$ edges, each player covers exactly three of them. Let S_i be the set of j such that player i covers $(v_{j-1},\, u_j,\, v_j)$. Then $\{S_i\}_{i\in[\![1,\,m]\!]}$ is a valid solution to the 3PARTITION instance, and the answer to Q_1 is yes.

The equivalence between Q_1 and Q_2 implies that Q_2 is NP-complete. □

Theorem 8.4 (Hardness of finding single-player worst equilibrium in DNC)
In a DNC game with a symmetric capability profile as defined in Definition 8.8, let $\mathrm{Equil}\,(b) \subseteq (\mathcal{L}_b)^n$ *denote the set of all pure Nash equilibria at the capability level* b*. For a strategy profile* $\boldsymbol{s} \in (\mathcal{L}_b)^n$*, let* $d_i(\boldsymbol{s})$ *denote the delay experienced by player* i*.*

Given $i \in \mathcal{N}$, determining whether there exists $\boldsymbol{s} \in \text{Equil}(b)$ such that $d_i(\boldsymbol{s}) \geq D$ for arbitrary $b \in [\![1, \bar{b}]\!]$ and $D \in \mathbb{R}$ is NP*-complete. Consequently, computing $\max_{\boldsymbol{s} \in \text{Equil}(b)} d_i(\boldsymbol{s})$ is* NP*-hard.*

Proof. With the setup in the proof of Theorem 8.3, modify the game to have $m+1$ players. Define the question Q_2': Does there exist $\boldsymbol{s} \in \text{Equil}(b)$ such that $d_1(\boldsymbol{s}) \geq 3m$?

If the answer to Q_1 is yes, define the strategy profile $\boldsymbol{s}^*$ as follows: players 2 to $m+1$ choose the strategies corresponding to the 3PARTITION solution. Player 1 chooses the strategy $(s, v_0, v_1, \ldots, v_{3m}, t)$. Then $d_1(\boldsymbol{s}^*) = 3m$, and the answer to Q_2' is yes.

If the answer to Q_2' is yes, let $\boldsymbol{s}'$ be a strategy profile such that $d_1(\boldsymbol{s}') \geq 3m$. Due to the definitions of the delay functions, s_1' must be the path $(s, v_0, v_1, \ldots, v_{3m}, t)$. Since s_1' is a Nash equilibrium, player 1 can not reduce their delay, which implies all edges (u_i, v_i) for $i \in [\![1, 3m]\!]$ are covered by exactly one player in $[\![2, m+1]\!]$. Their strategies can be reduced to a 3PARTITION solution similarly to the proof of Theorem 8.3,

The equivalence between Q_1 and Q_2' implies that Q_2' is NP-complete. □

Theorem 8.5 (Hardness of finding single-player extreme equilibrium in DncDa) *Theorems 8.3 and 8.4 hold for DncDa games.*

Proof. Modify the network structure in Fig. 8.2 as follows: for $v \in \{v_0\} \cup \{u_i, v_i\}_{i \in [\![1, 3m]\!]}$, add a zero-length edge from v to t with a constant delay of R. Then the modified game is a DncDa game. Let R be sufficiently large so that the zero-length edges and $f_2(x)$ for $x \geq 2$ are not chosen in any pure Nash equilibrium. The proofs of Theorem 8.3 and Theorem 8.4 then directly apply to the DncDa game. □

Corollary 8.6 (Hardness of finding single-player extreme equilibrium in McDncDa) *In a McDncDa game, for a capability profile $\boldsymbol{c} \in [\![1, \bar{b}]\!]^n$, let $\text{Equil}(\boldsymbol{c}) \subseteq \prod_{i \in \mathcal{N}} \mathcal{L}_{c_i}$ denote the set of all pure Nash equilibria in the instance with the capability profile $\boldsymbol{c}$. For a strategy profile $\boldsymbol{s} \in (\mathcal{L}_{\bar{b}})^n$, let $d_i(\boldsymbol{s})$ denote the delay experienced by player i.*

Given $i \in \mathcal{N}$, determining whether there exists $\boldsymbol{s} \in \text{Equil}(\boldsymbol{c})$ such that $d_i(\boldsymbol{s}) \leq D$ for arbitrary $\boldsymbol{c} \in [\![1, \bar{b}]\!]^n$ and $D \in \mathbb{R}$ is NP*-complete. Determining whether there exists $\boldsymbol{s} \in \text{Equil}(\boldsymbol{c})$ such that $d_i(\boldsymbol{s}) \geq D$ for arbitrary $\boldsymbol{c} \in [\![1, \bar{b}]\!]^n$ and $D \in \mathbb{R}$ is* NP*-complete. Consequently, computing $\min_{\boldsymbol{s} \in \text{Equil}(\boldsymbol{c})} d_i(\boldsymbol{s})$ or $\max_{\boldsymbol{s} \in \text{Equil}(\boldsymbol{c})} d_i(\boldsymbol{s})$ is* NP*-hard.*

Proof. Since DncDa is a special case of McDncDa, the hardness results in Theorem 8.5 apply to McDncDa games. □

Theorem 8.7 (Hardness of determining McDncDa capability transfer properties) *In a McDncDa game, for a capability profile $\boldsymbol{c} \in [\![1, \bar{b}]\!]^n$, let $\text{Equil}(\boldsymbol{c}) \subseteq \prod_{i \in \mathcal{N}} \mathcal{L}_{c_i}$ denote the set of all pure Nash equilibria in the instance with the capability profile $\boldsymbol{c}$. For a strategy profile $\boldsymbol{s} \in (\mathcal{L}_{\bar{b}})^n$, let $d_i(\boldsymbol{s})$ denote the delay experienced by player i.*

Given $i \in \mathcal{N}$, define a proposition $P_i(\boldsymbol{c})$ as:

$$
\begin{aligned}
P_i(\boldsymbol{c}): \quad & D_i'(\boldsymbol{c}) \leq D_i(\boldsymbol{c}) \\
\text{where } D_i(\boldsymbol{c}) &\coloneqq \min_{\boldsymbol{s} \in \text{Equil}(\boldsymbol{c})} d_i(\boldsymbol{s}) \\
D_i'(\boldsymbol{c}) &\coloneqq \max_{\boldsymbol{s} \in \text{Equil}(c_i+1, \boldsymbol{c}_{-i})} d_i(\boldsymbol{s})
\end{aligned}
$$

Then determining whether $P_i(\boldsymbol{c})$ holds for arbitrary $\boldsymbol{c} \in [\![1, \bar{b}]\!]^n$ is coNP*-complete.*

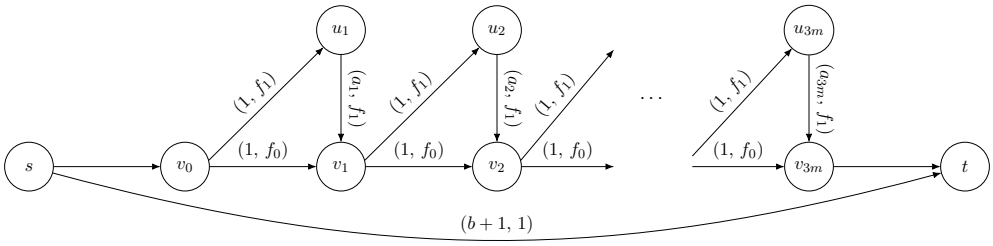


Figure 8.3: McDncDa game for the 3PARTITION reduction.
See Fig. 8.2 for the notation in the graph. All vertices except s and t have an unshown edge to t with zero length and sufficiently large delay so that it is not chosen in any pure Nash equilibrium.

Proof. With the same setup as in the proof of Theorem 8.3, define the capability profile $\boldsymbol{c} \coloneqq (b, b, \ldots, b)$. Construct a McDncDa game G as illustrated in Fig. 8.3. The game G differs from the game in Fig. 8.2 in two ways: (i) All vertices except s and t have an edge to t with zero length and sufficiently large delay to satisfy the requirement on the default action in DncDa games; and (ii) A new edge (s, t) is added with length $b+1$ and delay 1.

Given an instance of 3PARTITION, define a boolean variable S_{br} that indicates whether a solution to the 3PARTITION instance can be found by a best-response dynamics in the game G with the capability profile $\boldsymbol{c}$. Initially, all players choose the strategy $\boldsymbol{s}_0 = (p_0, p_0, \ldots, p_0)$ where p_0 is the path $(s, v_0, v_1, \ldots, v_{3m}, t)$. Note that $d_i(\boldsymbol{s}_0) = 3m$ for all $i \in \mathcal{N}$. For $k \geq 1$, let $\boldsymbol{s}_k$ be the strategy profile where an arbitrary player changes their strategy in $\boldsymbol{s}_{k-1}$ to reduce their own delay. One can easily verify that $d_i(\boldsymbol{s}_k) = 3m - n^*(\boldsymbol{s}_k)$ for all $i \in \mathcal{N}$ where

$$n^*(\boldsymbol{s}) \coloneqq |\{j \in [\![1, 3m]\!] \mid \text{The edge } (u_j, v_j) \text{ is not covered by any player in } \boldsymbol{s}\}|$$

Let r be the last round so that no player can reduce their delay in $\boldsymbol{s}_r$. Since in each round, some player reduces their delay by at least one, $d_1(\boldsymbol{s}_k) \leq d_1(\boldsymbol{s}_{k-1}) - 1$ holds for all $k \in [\![1, r]\!]$. It follows that $r \leq 3m$ and thus $\boldsymbol{s}_r$ is computable in polynomial time. Define $\mathrm{S}_{\mathrm{br}} = 1$ if $d_1(\boldsymbol{s}_r) = 0$ and $\mathrm{S}_{\mathrm{br}} = 0$ otherwise. If $\mathrm{S}_{\mathrm{br}} = 1$, then $\boldsymbol{s}_r$ corresponds to a solution to the 3PARTITION instance. If $\mathrm{S}_{\mathrm{br}} = 0$, then $\boldsymbol{s}_r$ is a Nash equilibrium and $d_1(\boldsymbol{s}_r) \geq 1$, which implies $D_1'(\boldsymbol{c}) \geq 1$. On the other hand, with the capability profile $(c_1 + 1, \boldsymbol{c}_{-1})$, player 1 can always choose the strategy that uses the edge (s, t), which has a delay of 1 regardless of other players' strategies, implying that $\forall \boldsymbol{s} \in \mathrm{Equil}\,(c_1 + 1, \boldsymbol{c}_{-1}) : d_1(\boldsymbol{s}) \leq 1$. Therefore, $\mathrm{S}_{\mathrm{br}} = 0$ implies $D_1'(\boldsymbol{c}) = 1$.

Proof of Theorem 8.3 shows that the answer to Q_1 is yes if and only if $D_1(\boldsymbol{c}) = 0$. Define the proposition Q_2' as $Q_2' \coloneqq (\mathrm{S}_{\mathrm{br}} = 1) \vee \neg P_1(\boldsymbol{c})$. If the answer to Q_1 is yes and $\mathrm{S}_{\mathrm{br}} = 0$, then $D_1(\boldsymbol{c}) = 0$ and $D_1'(\boldsymbol{c}) = 1$, which implies $P_1(\boldsymbol{c})$ is false and thus Q_2' is true. If Q_2' is true, then either $\mathrm{S}_{\mathrm{br}} = 1$, which derives a solution to the 3PARTITION instance, or $\mathrm{S}_{\mathrm{br}} = 0$ and $P_1(\boldsymbol{c})$ is false, which implies $D_1(\boldsymbol{c}) < D_1'(\boldsymbol{c}) = 1$, $D_1(\boldsymbol{c}) = 0$ follows, and thus the answer to Q_1 is yes. Therefore, Q_1 and Q_2' are equivalent. Since S_{br} is computable in polynomial time, $\neg P_1(\boldsymbol{c})$ is NP-complete, and therefore $P_1(\boldsymbol{c})$ is coNP-complete. □

Note the following facts:

- Theorem 8.7 implies that not all McDncDa games are capability-positive. Its proof constructs a game with multiple players and possibly multiple pure Nash equilibria.

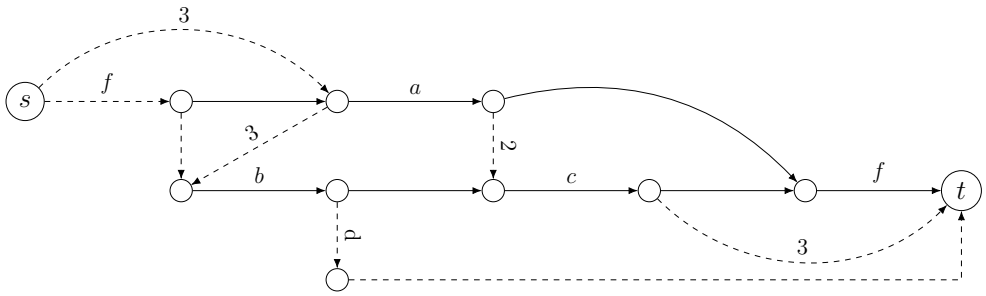


Figure 8.4: An example of a capability-negative McDncDa game. Player A gets a longer delay in the instance ($c_A = 2$, $c_B = 8$) compared to the instance ($c_A = 1$, $c_B = 8$).
Solid arrows represent the default actions (i.e., edges with zero distances). Dashed arrows are edges with non-zero distances. The numbers on the edges are the distances or 1 if omitted. Non-unit-length edges can be implemented by having multiple unit-length edges in series. If a node has no default action, then a default outgoing edge to the sink with infinite delay is assumed. Letter labels on the edges represent the resource type of the edge. Edges without letter labels have zero delays.

- Theorem 8.2 shows that using multiple players is necessary to achieve the hardness in Theorem 8.7.
- The next section (Section 8.2.3) will present a universally capability-positive two-player McDncDa game.
- Any single-player OLS-Game is universally capability-positive.

Given those facts, a natural question is whether all two-player McDncDa games are universally capability-positive. Example 8.2 answers this question negatively, with the additional requirement that the pure Nash equilibrium is unique.

Example 8.2 (A two-player capability-negative McDncDa game)
This example presents a two-player McDncDa game where a player gets a higher delay when their capability level increases. The pure Nash equilibrium is unique in the considered instances of this game. To simplify the presentation, I first give a congestion game. Then I present an equivalent McDncDa game.

Consider a congestion game with five resource types denoted by $R = \{a, b, c, d, f\}$. There are two players denoted as A and B. For $r \in R$, denote the delay function as $d_r = (x, y)$ so that the delay is x if one player covers the resource r and y if two players cover the resource r. The delay functions are defined as

$$d_a \coloneqq (1, 5), \quad d_b \coloneqq (0, 0), \quad d_c \coloneqq (0, 2), \quad d_d \coloneqq (4, 5), \quad d_f \coloneqq (0, M),$$

where $M \in \mathbb{R}_+$ is a sufficiently large constant so that the resource f is never covered by both players in any Nash equilibrium. The players have three capability levels with symmetric strategy spaces defined as

$$\mathcal{L}_1 \coloneqq \{\{a, f\}\}$$
$$\mathcal{L}_2 \coloneqq \{\{a, f\}, \{b, c, f\}\}$$
$$\mathcal{L}_3 \coloneqq \{\{a, f\}, \{b, c, f\}, \{a, c\}, \{b, d\}\}$$

Consider two instances where player A has capability levels 1 and 2, and player B has a fixed capability level 3. In the instance ($c_A = 1$, $c_B = 3$), player A chooses the only strategy in $\mathcal{L}_1$, and player B's best response is $\{b, d\}$. Their delays are 1 and 4 respectively.

In the instance ($c_A = 2$, $c_B = 3$), player A is motivated to switch to $\{b, c, f\}$ to reduce their delay to 0. Player B now has a better option $\{a, c\}$ which reduces their delay from 4 to 3. However, player A's delay now increases to 2 due to congestion at c. The Nash equilibrium is $(\{b, c, f\}, \{a, c\})$ with delays 2 and 3 for players A and B, respectively.

In this example, the increase in player A's capability level results in a higher delay for player A and a lower delay for player B. The social welfare remains the same in both instances.

Figure 8.4 presents an equivalent McDncDa game. In this game, player A has capability levels 1 and 2, and player B has a fixed capability level 8. In a Nash equilibrium, player A always covers the edges with label f, and player B avoids the edges with label f. In a Nash equilibrium, player A covers (f, a, f) when $c_A = 1$ and player A chooses within $\{(f, a, f), (f, b, c, f)\}$ when $c_A = 2$. Player B chooses within $\{(a, c), (b, d)\}$ since they want to avoid congestion on f. Therefore, the Nash equilibrium in this McDncDa game is the same as the Nash equilibrium in the congestion game discussed above. ◁

8.2.3 Case study: mixed gold and mines games

8.2.3.1 The game model and capability transfer properties

Given the nonexistence of interesting capability transfer properties in the DncDaS games as discussed in Section 8.2.1 and the hardness of determining the capability transfer properties in McDncDa games as discussed in Section 8.2.2, this section analyzes a restricted class of McDncDa games called *Mixed Gold and Mines Games* (MGMG). MGMG is parameterized by a few numbers. Its simple structure allows us to derive exact expressions for the payoffs given the game parameters and the capability profile. MGMG is universally capability-positive. MGMG extends the alternating ordering Gold and Mines Game analyzed in Yang, Jia, and Rinard [YJR22] by allowing players to have different capabilities.

MGMG is a two-player McDncDa game. The players are denoted as A and B. The game is parameterized by five numbers ($M \in \mathbb{Z}_+$, $\rho \in \mathbb{R}$, $\mu \in \mathbb{R}$, $c_\mathrm{A} \in \mathbb{Z}_+$, $c_\mathrm{B} \in \mathbb{Z}_+$). MGMG has resources arranged as a specific pattern; players use line segments to cover resources to receive payoffs. As in all congestion games, MGMG games always have pure Nash equilibria [Ros73].

Resource Layout: Each MGMG game has $4M$ resources arranged on two lines. Each resource is either a gold site or a mine site. Each line contains M gold sites and M mine sites in alternating order. Resources are placed at distinct horizontal locations $[\![0, 4M - 1]\!]$. For the resource at location i, $y_i = (i + 1) \bmod 2$ indicates which line it is placed on, and $t_i = \mathbb{1}_{i \bmod 4 \leq 1}$ indicates whether it is a gold site ($t_i = 1$) or a mine site ($t_i = 0$).

Game Objective: Two players maximize their payoff by using line segments to cover the resources. A *delay function* $r_t(n)$, where $t \in \{m, g\}$ is the resource type and $n \in \{1, 2\}$ is the number of players covering the resource, specifies the payoff for covering a resource. For gold sites, $r_g(1) = 1$ and $r_g(2) = \rho$, where $0 < \rho < 1$ so that if two players cover the same goad site they receive a smaller payoff. For mine sites, $r_m(n) = \mu < 0$ is a constant penalty.

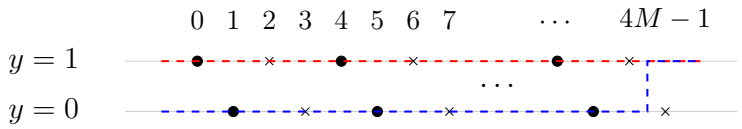


Figure 8.5: An example MGMG instance.
Each dot (resp. cross) is a gold (resp. mine) site. The dashed lines represent a pure Nash equilibrium when $c_{\mathrm{A}} = 1$ and $c_{\mathrm{B}} = 2$ (with $\rho < -\mu < 1$).

Strategy Space: Each player $p \in \{A, B\}$ uses a function $f_p : [\![0, 4M-1]\!] \mapsto \{0, 1\}$ to specify which line player p covers at each horizontal location. Player p covers the resource at location i if $f_p(i) = y_i = (i+1) \bmod 2$. The strategy space $\mathcal{L}_c$ of a player with capability level c contains all functions with no more than c segments:

$$\mathrm{Seg}(f) \coloneqq \left(\sum_{i=0}^{4M-2} \mathbb{1}_{f(i) \neq f(i+1)} \right) + 1$$
$$\mathcal{L}_c \coloneqq \{ f : [\![0, 4M-1]\!] \mapsto \{0, 1\} \mid \mathrm{Seg}(f) \leq c \}$$

Let c_{A} and c_{B} denote the capability levels of player A and B respectively, and use $f_{\mathrm{A}}(\cdot)$ and $f_{\mathrm{B}}(\cdot)$ for their strategies. For each resource, its default action is to cover the next resource on the same line; the player can choose to switch to the other line if they have sufficient capability. Therefore, MGMG is a special case of McDncDa. As in DncDa, the capability level has a natural interpretation as the size of programs in a Domain-Specific Language (DSL) describing the strategy space.

The payoffs of players in a pure Nash equilibrium of an MGMG game are given by the following theorem:

Theorem 8.8 (Payoffs in MGMG)
In a pure Nash equilibrium of an instance of MGMG parameterized by $(M, \rho, \mu, c_{\mathrm{A}}, c_{\mathrm{B}})$ *where* $0 < \rho < -\mu < 1$*, the players receive the following payoffs* u_{A} *and* u_{B}*:*

$$u_{\mathrm{A}} = \left\lfloor \frac{c'_{\mathrm{A}} + t - 1}{2} \right\rfloor \rho - \left\lfloor \frac{c'_{\mathrm{A}} - t}{2} \right\rfloor \mu + \left\lfloor \frac{c'_{\mathrm{B}} - t}{2} \right\rfloor (\rho - 1) + (\mu + 1)M$$
$$u_{\mathrm{B}} = \left\lfloor \frac{c'_{\mathrm{B}} - t}{2} \right\rfloor \rho - \left\lfloor \frac{c'_{\mathrm{B}} + t - 1}{2} \right\rfloor \mu + \left\lfloor \frac{c'_{\mathrm{A}} + t - 1}{2} \right\rfloor (\rho - 1) + (\mu + 1)M$$
$$\text{where } c'_{\mathrm{A}} = \min\{c_{\mathrm{A}}, 2M+1\}, \quad c'_{\mathrm{B}} = \min\{c_{\mathrm{B}}, 2M+1\}, \quad t \in \{0, 1\}$$

Three cases determine the value of t*:*

- *When* $\max\{c_{\mathrm{A}}, c_{\mathrm{B}}\} \leq 2M$*, there are two classes of Nash equilibria distinguished by* $t = 0$ *and* $t = 1$*.*
- *When* $\min\{c_{\mathrm{A}}, c_{\mathrm{B}}\} \leq 2M < \max\{c_{\mathrm{A}}, c_{\mathrm{B}}\}$*,* $t = 0$ *if* $c_{\mathrm{A}} \leq 2M$ *and* $t = 1$ *if* $c_{\mathrm{B}} \leq 2M$*.*
- *When* $\min\{c_{\mathrm{A}}, c_{\mathrm{B}}\} \geq 2M + 1$*, there is one Nash equilibrium. The above formulas give the same payoffs regardless of* $t = 0$ *or* $t = 1$*.*

Remark 8.1 If both players have the same capability level c, the social welfare is $u_{\mathrm{A}} + u_{\mathrm{B}} = (2\rho - \mu - 1)(\min\{c, 2M+1\} - 1) + 2(\mu+1)M$, which confirms Theorem 16 of Yang, Jia, and Rinard [YJR22] up to a constant offset because MGMG removes the last gold site on each line compared to the original Gold and Mines Game to simplify the analysis.

Corollary 8.9 (Capability transfer properties in MGMG)
Theorem 8.8 implies the following capability transfer properties:

- *MGMG is universally capability-positive. Specifically, if a player increases their capability level by one without exceeding $2M+1$, their payoff increases by ρ or $-\mu$.*
- *If a player increases their capability level by one without exceeding $2M+1$, their opponent's payoff stays the same or decreases by $1-\rho$.*
- *MGMG is socially capability-positive if the gold sites have increasing returns to scale, i.e., $\rho > 0.5$. Specifically, if both player's capability levels increase by one, the change of social welfare can be $2\rho-\mu-1$, -2μ, or $4\rho-2$. If $\rho > 0.5$, which intuitively means that the gold sites have increasing returns to scale, then the change of social welfare is always positive. In particular, if t changes or if $c_{\mathrm{A}} \equiv c_{\mathrm{B}} \pmod 2$, the social welfare increases by $2\rho-\mu-1$, which gives a weaker condition for positive social welfare change.*

Proof. A proof for the first statement is given. Other statements can be proved similarly. Assume one player's capability level increases from c to $c+1$ with $c \le 2M$, and the equilibrium parameter t is changed to t' with $s \coloneqq t' - t \in \{0, 1, -1\}$. Let $a \coloneqq \left\lfloor \frac{c+t-1}{2} \right\rfloor$, $b \coloneqq \left\lfloor \frac{c-t}{2} \right\rfloor$, $a' \coloneqq \left\lfloor \frac{c+1+t'-1}{2} \right\rfloor = \left\lfloor \frac{c+t-1+(s+1)}{2} \right\rfloor$, and $b' \coloneqq \left\lfloor \frac{c+1-t'}{2} \right\rfloor = \left\lfloor \frac{c-t+(1-s)}{2} \right\rfloor$. It is easy to verify that $(a', b') \in \{(a+1, b), (a, b+1)\}$, which implies that this player's payoff increases by either ρ or $-\mu$. □

8.2.3.2 Proof of Theorem 8.8

We introduce some notation:

- A pair $(f_{\mathrm{A}}, f_{\mathrm{B}})$ denotes a strategy profile, i.e., the pure strategies of both players.
- Given a strategy profile, $u_{\mathrm{A}}(f_{\mathrm{A}}, f_{\mathrm{B}})$ and $u_{\mathrm{B}}(f_{\mathrm{A}}, f_{\mathrm{B}})$ are the payoffs of individual players.
- Given a strategy $f(\cdot)$, $\mathcal{G}(f)$, $\mathcal{M}(f)$, ${}^{\#}\mathrm{g}(f)$, and ${}^{\#}\mathrm{m}(f)$ denote the locations and numbers of gold and mine sites covered by the strategy, respectively:

$$
\begin{aligned}
\mathcal{G}(f) &\coloneqq \{4i \mid 0 \le i < M \text{ and } f(4i) = 1\} \\
&\quad \cup \{4i+1 \mid 0 \le i < M \text{ and } f(4i+1) = 0\} \\
\mathcal{M}(f) &\coloneqq \{4i+2 \mid 0 \le i < M \text{ and } f(4i+2) = 1\} \\
&\quad \cup \{4i+3 \mid 0 \le i < M \text{ and } f(4i+3) = 0\} \\
{}^{\#}\mathrm{g}(f) &\coloneqq |\mathcal{G}(f)|, \quad {}^{\#}\mathrm{m}(f) \coloneqq |\mathcal{M}(f)|
\end{aligned}
$$

- Given a strategy $f_p(\cdot)$ for player p, *discontinuity points* (DPs) are the locations where $f_p(\cdot)$ changes the line that p covers. We also differentiate between *upward discontinuity points* (UDPs, denoted by $\mathcal{D}^{\uparrow}(f)$) and *downward discontinuity points* (DDPs, denoted by $\mathcal{D}^{\downarrow}(f)$):

$$\mathcal{D}^{\uparrow}(f) \coloneqq \{i \mid 0 \le i \le 4M-2 \text{ and } f(i)=0 \text{ and } f(i+1)=1\}$$
$$\mathcal{D}^{\downarrow}(f) \coloneqq \{i \mid 0 \le i \le 4M-2 \text{ and } f(i)=1 \text{ and } f(i+1)=0\}$$

  Note that $\mathrm{Seg}(f) = \left|\mathcal{D}^{\uparrow}(f)\right| + \left|\mathcal{D}^{\downarrow}(f)\right| + 1$.

- A strategy $f(\cdot)$ is a *perfect cover* for resources located between $[\![a, b]\!]$ if all gold sites are covered and all mine sites are avoided: $[\![a, b]\!] \setminus \mathcal{G}(f) = \emptyset$ and $[\![a, b]\!] \cap \mathcal{M}(f) = \emptyset$. We also call the resources $[\![a, b]\!]$ *perfectly covered* in this case, and *imperfectly covered* otherwise. Note that to perfectly cover resources $[\![4i, 4j-1]\!]$ for $i < j$, one needs $2(j-i)+1$ segments.

- *Strict strategy spaces* use exactly the given number of segments:

$$\widetilde{\mathcal{L}}_1 \coloneqq \mathcal{L}_1$$
$$\widetilde{\mathcal{L}}_c \coloneqq \mathcal{L}_c \setminus \mathcal{L}_{c-1} = \{f : [\![0, 4M-1]\!] \mapsto \{0, 1\} \mid \mathrm{Seg}(f) = c\}$$

- A strategy profile $(f_{\mathrm{A}}, f_{\mathrm{B}})$ is a *complete-gold-coverage* for a MGMG if both players cover all gold sites together, i.e., $|\mathcal{G}(f_{\mathrm{A}}) \cup \mathcal{G}(f_{\mathrm{B}})| = 2M$.

The first lemma shows that DPs only occur at certain locations:

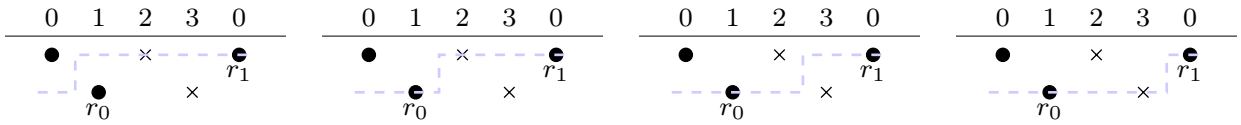
Lemma 8.10
Let $f(\cdot)$ be a best response of a player given the other player's strategy. Upward discontinuity points in $f(\cdot)$ occur only at neighboring mine sites, and downward discontinuity points occur only at neighboring gold sites:

$$\forall i \in \mathcal{D}^{\uparrow}(f) : i \bmod 4 = 2$$
$$\forall i \in \mathcal{D}^{\downarrow}(f) : i \bmod 4 = 0$$

Proof. Let's consider cases in a local region for different values of $f(4k)$ and $f(4k+4)$.

- $f(4k)=0$, $f(4k+4)=1$: Figure 8.6 shows the cases with one DP. The payoffs of covered gold sites are denoted as r_0 and r_1, which can be 1 or ρ depending on the opponent's strategy. Clearly, the payoff is maximized only when the DP is at location 2 modulo 4. It can be verified that using more DPs while maintaining $f(4k)=0$ and $f(4k+4)=1$ does not improve payoff.

- $f(4k)=1$, $f(4k+4)=0$: Similarly, the best response in this case has one DDP at location 0.



(a) Upward at 0. Payoff is $\mu + r_1$.

(b) Upward at 1. Payoff is $\mu + r_0 + r_1$.

(c) Upward at 2. Payoff is $r_0 + r_1$, which is the best.

(d) Upward at 3. Payoff is $\mu + r_0 + r_1$.

Figure 8.6: Cases of a $f(4k) = 0$ and $f(4k+4) = 1$ in a local region with one DP. The numbers are locations of resources modulo 4. Dashed lines indicate a local part of the strategy.

- $f(4k) = f(4k+4) = 0$: The best response should have no DP. If there are DPs, there should be one UDP and one DDP to cover one gold site and no mine site, but moving the DDP rightward to also cover the gold at $4k+4$ gives better payoff with the same number of segments.
- $f(4k) = f(4k+4) = 1$: The best response should either have no DP (covering two gold sites and one mine site) or two DPs (covering three gold sites and no mine site) at locations 0 and 2 modulo 4. □

The next lemma shows that the number of gold and mine sites covered by an optimal strategy only depends on $f(0)$ and $\mathrm{Seg}(f)$:

Lemma 8.11
If $f(\cdot)$ is a strategy that conforms to Lemma 8.10, then

$$\begin{aligned} {}^{\#}\mathrm{g}(f) &= M + \left\lfloor \frac{\mathrm{Seg}(f) + f(0) - 1}{2} \right\rfloor \\ {}^{\#}\mathrm{m}(f) &= M - \left\lfloor \frac{\mathrm{Seg}(f) - f(0)}{2} \right\rfloor \end{aligned} \tag{8.2}$$

Proof. Define a series of strategies $\{f_i\}$. Let $f_0 = f$ and define f_{i+1} to be the strategy obtained by removing the last DP of f_i, i.e., $f_{i+1}(x) = f_i(\min\{x, x_i\})$ where $x_i = \max\left(\mathcal{D}^{\uparrow}(f_i) \cup \mathcal{D}^{\downarrow}(f_i)\right)$. Lemma 8.10 implies that each DDP adds an extra gold site and each UDP avoids a mine site, which means either ${}^{\#}\mathrm{g}(f_i) - {}^{\#}\mathrm{g}(f_{i+1}) = 1$ or ${}^{\#}\mathrm{m}(f_{i+1}) - {}^{\#}\mathrm{m}(f_i) = 1$, depending on whether the last DP of f_i is DDP or UDP. It follows that ${}^{\#}\mathrm{g}(f_i) = {}^{\#}\mathrm{g}(f_j) + \left|\mathcal{D}^{\downarrow}(f_i)\right| - \left|\mathcal{D}^{\downarrow}(f_j)\right|$ and ${}^{\#}\mathrm{m}(f_i) = {}^{\#}\mathrm{m}(f_j) - \left|\mathcal{D}^{\uparrow}(f_i)\right| + \left|\mathcal{D}^{\uparrow}(f_j)\right|$ for any pair i, j.

First assume $f(0) = 1$. In this case, since UDPs and DDPs are interleaving, we have $\left|\mathcal{D}^{\downarrow}(f)\right| = \left\lfloor \frac{\mathrm{Seg}(f)}{2} \right\rfloor$ and $\left|\mathcal{D}^{\uparrow}(f)\right| = \left\lfloor \frac{\mathrm{Seg}(f)-1}{2} \right\rfloor$. Let $c = \mathrm{Seg}(f) - 1$. It holds that ${}^{\#}\mathrm{g}(f_c) = {}^{\#}\mathrm{m}(f_c) = M$ since f_c covers exactly one line. Therefore, ${}^{\#}\mathrm{g}(f) = {}^{\#}\mathrm{g}(f_0) = {}^{\#}\mathrm{g}(f_c) + \left|\mathcal{D}^{\downarrow}(f_0)\right| - \left|\mathcal{D}^{\downarrow}(f_c)\right| = M + \left\lfloor \frac{\mathrm{Seg}(f)}{2} \right\rfloor$ and ${}^{\#}\mathrm{m}(f) = M - \left\lfloor \frac{\mathrm{Seg}(f)-1}{2} \right\rfloor$. A similar analysis for the case $f(0) = 0$ gives ${}^{\#}\mathrm{g}(f) = M + \left\lfloor \frac{\mathrm{Seg}(f)-1}{2} \right\rfloor$ and ${}^{\#}\mathrm{m}(f) = M - \left\lfloor \frac{\mathrm{Seg}(f)}{2} \right\rfloor$. Lemma 8.11 summarizes these results compactly. □

Algorithm 8.1 Modify a strategy to use more segments

Require: Game scale $M \geq 2$
Require: Player capability c_{A} such that $c_{\mathrm{A}} \leq 2M - 1$
Require: A strategy $f'_{\mathrm{A}}(\cdot)$ that conforms to Lemma 8.10 such that $\mathrm{Seg}(f'_{\mathrm{A}}) \leq c_{\mathrm{A}}$ and the last four resources are imperfectly covered.
Ensure: A strategy $f_{\mathrm{A}}(\cdot)$ that conforms to Lemma 8.10 such that $\mathrm{Seg}(f_{\mathrm{A}}) = c_{\mathrm{A}}$, $\mathcal{G}(f_{\mathrm{A}}) \supseteq \mathcal{G}(f'_{\mathrm{A}})$, and $f_{\mathrm{A}}(0) = f'_{\mathrm{A}}(0)$.
$k \leftarrow 0$
$f_{\mathrm{A}} \leftarrow f'_{\mathrm{A}}$
while $c_{\mathrm{A}} - \mathrm{Seg}(f_{\mathrm{A}}) \geq 2$ **do**
    ▷ When entering the loop, all resources in $[\![4, 4k-1]\!]$ are perfectly covered and $f_{\mathrm{A}}(4k) = 1$ when $k \geq 1$. Perfectly covering $[\![4, 4k-1]\!]$ requires $\mathrm{Seg}(f_{\mathrm{A}}) \geq 2(k-1)+1$. We also have $\mathrm{Seg}(f_{\mathrm{A}}) \leq c_{\mathrm{A}} - 2 \leq 2M-3$ due to the loop condition, thus $2k-1 \leq \mathrm{Seg}(f_{\mathrm{A}}) \leq 2M-3$ which means $k \leq M-1$. When $k \geq 1$, each iteration modifies $f_{\mathrm{A}}(\cdot)$ to perfectly cover $[\![4k, 4k+3]\!]$ using no more than two new segments.
    if $f_{\mathrm{A}}(4k+3) = 0$ **then**
        ▷ Lemma 8.10 ensures $f_{\mathrm{A}}(4k+i) = 0$ for $2 \leq i \leq 5$.
        $f_{\mathrm{A}}(4k+3) \leftarrow 1$
        if $k+1 < M$ **then**
            $f_{\mathrm{A}}(4k+4) \leftarrow 1$
        end if
    else if $k > 0$ or $f_{\mathrm{A}}(4k) = 1$ **then**
        ▷ We now have $f_{\mathrm{A}}(4k) = f_{\mathrm{A}}(4k+3) = 1$. Lemma 8.10 ensures $f_{\mathrm{A}}(4k+4) = 1$.
        $f_{\mathrm{A}}(4k+1) \leftarrow 0$
        $f_{\mathrm{A}}(4k+2) \leftarrow 0$
    end if
    $k \leftarrow k+1$
end while
if $c_{\mathrm{A}} - \mathrm{Seg}(f_{\mathrm{A}}) = 1$ **then**
    ▷ We have $\mathrm{Seg}(f_{\mathrm{A}}) = c_{\mathrm{A}} - 1 < 2M-1$ in this case. Thus the resources $[4, 4M-1]$ are imperfectly covered. Due to our requirement on $f'_{\mathrm{A}}(\cdot)$ and the way we construct $f_{\mathrm{A}}(\cdot)$, the last four resources are imperfectly covered. We modify the strategy on the last few resources to use one more segment.
    if $f_{\mathrm{A}}(4M-1) = 0$ **then**
        $f_{\mathrm{A}}(4M-1) \leftarrow 1$
    else if $f_{\mathrm{A}}(4M-2) = 1$ **then**
        ▷ Lemma 8.10 implies $f_{\mathrm{A}}(4M-i) = 1$ for $1 \leq i \leq 4$
        $f_{\mathrm{A}}(4M-3) \leftarrow 0$
        $f_{\mathrm{A}}(4M-2) \leftarrow 0$
        $f_{\mathrm{A}}(4M-1) \leftarrow 0$
    else
        ▷ Now $f_{\mathrm{A}}(4M-2) = 0$ and $f_{\mathrm{A}}(4M-1) = 1$. Since the last four resources are imperfectly covered, we have $f_{\mathrm{A}}(4M-i) = 0$ for $2 \leq i \leq 6$.
        $f_{\mathrm{A}}(4M-5) \leftarrow 1$
        $f_{\mathrm{A}}(4M-4) \leftarrow 1$
        $f_{\mathrm{A}}(4M-1) \leftarrow 0$
    end if
end if
return f_{A}

Now let's shift our attention from one player's strategy to a strategy profile. The following lemma shows that at equilibrium, both players cover all gold sites together:

Lemma 8.12
If $\rho < -\mu < 1$ and $(f_\mathrm{A},\, f_\mathrm{B})$ is a pure Nash equilibrium when players are limited to the strict strategy spaces $\widetilde{\mathcal{L}}_{c_\mathrm{A}}$ and $\widetilde{\mathcal{L}}_{c_\mathrm{B}}$, then $(f_\mathrm{A},\, f_\mathrm{B})$ is a complete-gold-coverage.

Proof. Let's prove this statement in two steps. The first step shows that for any player, their payoff is maximized when they cover as many unoccupied gold sites as possible. The second step shows that complete-gold-coverage is always feasible.

Let $T(f_\mathrm{A},\, f_\mathrm{B})$ be the total number of gold sites covered by a strategy profile:

$$T(f_\mathrm{A},\, f_\mathrm{B}) \coloneqq |\mathcal{G}(f_\mathrm{A}) \cup \mathcal{G}(f_\mathrm{B})|$$

Without loss of generality, focus on player A. This paragraph shows that if there is a strategy f'_A such that $T(f_\mathrm{A},\, f_\mathrm{B}) < T(f'_\mathrm{A},\, f_\mathrm{B})$ where $\{f_\mathrm{A},\, f'_\mathrm{A}\} \subseteq \widetilde{\mathcal{L}}_{c_\mathrm{A}}$, then $(f_\mathrm{A},\, f_\mathrm{B})$ is not a Nash equilibrium because A can get better payoff by switching to f'_A. Note that the number of gold sites covered by both players in the strategy profile $(f_\mathrm{A},\, f_\mathrm{B})$ is ${}^{\#}\mathrm{g}(f_\mathrm{A}) + {}^{\#}\mathrm{g}(f_\mathrm{B}) - T(f_\mathrm{A},\, f_\mathrm{B})$, while the number of gold sites covered by A exclusively is $T(f_\mathrm{A},\, f_\mathrm{B}) - {}^{\#}\mathrm{g}(f_\mathrm{B})$, which implies:

$$\begin{aligned} u_\mathrm{A}(f_\mathrm{A},\, f_\mathrm{B}) = {} & \big(T(f_\mathrm{A},\, f_\mathrm{B}) - {}^{\#}\mathrm{g}(f_\mathrm{B})\big) \cdot 1 \\ & + \big({}^{\#}\mathrm{g}(f_\mathrm{A}) + {}^{\#}\mathrm{g}(f_\mathrm{B}) - T(f_\mathrm{A},\, f_\mathrm{B})\big) \cdot \rho + {}^{\#}\mathrm{m}(f_\mathrm{A}) \cdot \mu \end{aligned}$$

Substituting (8.2) into the above yields:

$$\begin{aligned} u_\mathrm{A}(f_\mathrm{A},\, f_\mathrm{B}) = {} & \rho \left\lfloor \frac{c_\mathrm{A} + f_\mathrm{A}(0) - 1}{2} \right\rfloor + (-\mu) \left\lfloor \frac{c_\mathrm{A} - f_\mathrm{A}(0)}{2} \right\rfloor + (1-\rho) T(f_\mathrm{A},\, f_\mathrm{B}) \\ & + (\rho - 1)\, {}^{\#}\mathrm{g}(f_\mathrm{B}) + (\rho + \mu) M \end{aligned} \tag{8.3}$$

Let $h(f) \coloneqq \rho \left\lfloor \frac{c_\mathrm{A} + f(0) - 1}{2} \right\rfloor + (-\mu) \left\lfloor \frac{c_\mathrm{A} - f(0)}{2} \right\rfloor$ be the first two terms. One can verify that $h(f'_\mathrm{A}) - h(f_\mathrm{A}) \in \{0,\, \rho + \mu,\, -\rho - \mu\}$ for all possible values of c_A, $f_\mathrm{A}(0)$, and $f'_\mathrm{A}(0)$. Note $\rho < -\mu$ implies $\rho + \mu < 0$. Thus $h(f'_\mathrm{A}) - h(f_\mathrm{A}) \geq \rho + \mu$.

$$\begin{aligned} u_\mathrm{A}(f'_\mathrm{A},\, f_\mathrm{B}) - u_\mathrm{A}(f_\mathrm{A},\, f_\mathrm{B}) &= h(f'_\mathrm{A}) - h(f_\mathrm{A}) + (T(f'_\mathrm{A},\, f_\mathrm{B}) - T(f_\mathrm{A},\, f_\mathrm{B}))(1-\rho) \\ &\geq (\rho + \mu) + (1 - \rho) > 0 \end{aligned}$$

Therefore, A first maximizes $T(f_\mathrm{A},\, f_\mathrm{B})$ and then maximizes $h(f_\mathrm{A})$ in their best response. The maximum possible value of $T(f_\mathrm{A},\, f_\mathrm{B})$ is $2M$ which is achieved when $(f_\mathrm{A},\, f_\mathrm{B})$ is a complete-gold-coverage.

The rest of this proof shows that complete-gold-coverage is always feasible. Assume without loss of generality that $c_\mathrm{A} \geq c_\mathrm{B}$. The goal is to show that for any strategy $f_\mathrm{B}(\cdot)$ that conforms to Lemma 8.10, there exists $f_\mathrm{A} \in \widetilde{\mathcal{L}}_{c_\mathrm{A}}$ such that $T(f_\mathrm{A},\, f_\mathrm{B}) = 2M$.

If $c_\mathrm{A} \geq 2M$, then A can cover all gold sites trivially. Now consider the case $1 \leq c_\mathrm{A} \leq 2M - 1$. First construct a strategy $f'_\mathrm{A}(\cdot)$ that may or may not use all the segments. For $0 \leq k < M$, set $f'_\mathrm{A}(4k) = 1 - f_\mathrm{B}(4k)$ and $f'_\mathrm{A}(4k+3) = 1 - f_\mathrm{B}(4k+3)$. If $f'_\mathrm{A}(4k) = f'_\mathrm{A}(4k+3) = y$, set

$f'_{\mathrm{A}}(4k+1) = f'_{\mathrm{A}}(4k+2) = y$; otherwise add one discontinuity point at $f'_{\mathrm{A}}(4k)$ or $f'_{\mathrm{A}}(4k+2)$ according to Lemma 8.10. Note that Lemma 8.10 also implies $f'_{\mathrm{A}}(4k-1) = f'_{\mathrm{A}}(4k)$. It is easy to verify that $T(f'_{\mathrm{A}}, f_{\mathrm{B}}) = 2M$ and $\mathrm{Seg}(f'_{\mathrm{A}}) \le \mathrm{Seg}(f_{\mathrm{B}}) \le c_{\mathrm{A}}$. Finally, derive f_{A} from f'_{A} using Algorithm 8.1 so that $\mathrm{Seg}(f_{\mathrm{A}}) = c_{\mathrm{A}}$. □

Proof (Theorem 8.8). Given a strategy profile $(f_{\mathrm{A}}, f_{\mathrm{B}})$ that is a complete-gold-coverage, (8.3) yields the following payoff for player A since $T(f_{\mathrm{A}}, f_{\mathrm{B}}) = 2M$:

$$u_{\mathrm{A}}(f_{\mathrm{A}}, f_{\mathrm{B}}) = \rho \left\lfloor \frac{\mathrm{Seg}(f_{\mathrm{A}}) + f_{\mathrm{A}}(0) - 1}{2} \right\rfloor + (-\mu) \left\lfloor \frac{\mathrm{Seg}(f_{\mathrm{A}}) - f_{\mathrm{A}}(0)}{2} \right\rfloor + (\rho - 1)\, {}^{\#}\mathrm{g}(f_{\mathrm{B}}) + (\mu - \rho + 2)M \tag{8.4}$$

For two strategy profiles $(f_{\mathrm{A}}, f_{\mathrm{B}})$ and $(f'_{\mathrm{A}}, f_{\mathrm{B}})$ that are both complete-gold-coverage, (8.4) implies the following:

- If $\mathrm{Seg}(f_{\mathrm{A}}) < \mathrm{Seg}(f'_{\mathrm{A}})$ and $\min\{-\mu, \rho\} > 0$, then $u_{\mathrm{A}}(f_{\mathrm{A}}, f_{\mathrm{B}}) < u_{\mathrm{A}}(f'_{\mathrm{A}}, f_{\mathrm{B}})$.
- If $f_{\mathrm{A}}(0) = 1$, $f'_{\mathrm{A}}(0) = 0$, $\mathrm{Seg}(f_{\mathrm{A}}) = \mathrm{Seg}(f'_{\mathrm{A}})$, and $\rho + \mu < 0$, then $u_{\mathrm{A}}(f_{\mathrm{A}}, f_{\mathrm{B}}) \le u_{\mathrm{A}}(f'_{\mathrm{A}}, f_{\mathrm{B}})$.

In other words, player p's best response $f^*_p(\cdot)$ to the other player's strategy $f_o(\cdot)$ satisfies:

- $\left(f^*_p, f_o\right)$ is a complete-gold-coverage. The DPs of $f^*_p(\cdot)$ satisfy Lemma 8.10.
- $f^*_p(\cdot)$ uses the full capability of player p up to $2M+1$ line segments, i.e., $\mathrm{Seg}(f^*_p) = \min\{c_p, 2M+1\}$.
- If there is a strategy that starts at line 0 (i.e., $f_p(0) = 0$) and satisfies both of the above constraints, then player p plays such a strategy. This is called the *preference of empty start* property.

Assume without loss of generality that $c_{\mathrm{B}} \le c_{\mathrm{A}}$. Consider the following cases to derive the payoffs of players at equilibrium:

- $2M + 1 \le c_{\mathrm{B}} \le c_{\mathrm{A}}$: All resources are perfectly covered by both players. They receive the same payoff of $2M\rho$.
- $c_{\mathrm{B}} < 2M + 1 \le c_{\mathrm{A}}$: A perfectly covers all resources. B starts at $f_{\mathrm{B}}(0) = 0$ and uses all their capability. Eq. (8.2) implies that

$$ {}^{\#}\mathrm{g}(f_{\mathrm{B}}) = M + \left\lfloor \frac{c_{\mathrm{B}} - 1}{2} \right\rfloor, \quad {}^{\#}\mathrm{m}(f_{\mathrm{B}}) = M - \left\lfloor \frac{c_{\mathrm{B}}}{2} \right\rfloor$$

  Therefore, the payoffs are:

$$u_{\mathrm{A}} = 2M + (\rho - 1)\, {}^{\#}\mathrm{g}(f_{\mathrm{B}}) = (\rho + 1)M + (\rho - 1) \left\lfloor \frac{c_{\mathrm{B}} - 1}{2} \right\rfloor$$

$$u_{\mathrm{B}} = {}^{\#}\mathrm{g}(f_{\mathrm{B}})\rho + {}^{\#}\mathrm{m}(f_{\mathrm{B}})\mu = \left\lfloor \frac{c_{\mathrm{B}} - 1}{2} \right\rfloor \rho - \left\lfloor \frac{c_{\mathrm{B}}}{2} \right\rfloor \mu + (\rho + \mu)M$$

- $c_B \leq c_A < 2M + 1$: Let B choose an arbitrary strategy $f_B(\cdot)$ satisfying Lemma 8.10. If $f_B(0) = 0$, A will set $f_A(0) = 1$ to ensure a complete-gold-coverage; otherwise if $f_B(0) = 1$, A will set $f_A(0) = 0$ due to the preference of empty start property noted above. Player A can derive a best response $f_A(\cdot)$ according to the proof of Lemma 8.12. Following a similar reasoning from B's perspective, it can be shown that $f_B(\cdot)$ is also a best response of B given A's strategy $f_A(\cdot)$. Therefore, (f_A, f_B) is a Nash equilibrium. There are two different classes of Nash equilibria: one with $f_A(0) = 0$ and $f_B(0) = 1$, and the other with $f'_A(0) = 1$ and $f'_B(0) = 0$. Let $f_A(0) = t$ and $f_B(0) = 1 - t$. Substituting $f_A(0) = t$ and $f_B(0) = 1 - t$ into (8.4) yields the payoffs:

$$u_A = \left\lfloor \frac{c_A + t - 1}{2} \right\rfloor \rho - \left\lfloor \frac{c_A - t}{2} \right\rfloor \mu + \left\lfloor \frac{c_B - t}{2} \right\rfloor (\rho - 1) + (\mu + 1)M$$
$$u_B = \left\lfloor \frac{c_B - t}{2} \right\rfloor \rho - \left\lfloor \frac{c_B + t - 1}{2} \right\rfloor \mu + \left\lfloor \frac{c_A + t - 1}{2} \right\rfloor (\rho - 1) + (\mu + 1)M$$
$$t \in \{0, 1\}$$

Theorem 8.8 summarizes the results of these three cases. □

8.3 Related work

My research goal is to formalize a hierarchy of player capabilities in terms of strategy spaces and analyze the associated capability transfer properties. This chapter realizes this idea by defining the strategy space as programs in a Domain-Specific Language (DSLs). There are previous efforts to characterize certain aspects of a game using formal computational models.

Tennenholtz [Ten04] proposes using loop-free and branch-based programs in a DSL to represent a player's strategy. They assume a player's strategy program has access to programs of other players and can choose a strategy based on syntactical comparison of the other players' programs. The strategy profile is obtained by executing all players' programs. An interesting result is that there is a program equilibrium of both parties cooperating in the one-shot prisoner's dilemma. Fortnow [For09] extends this model by allowing arbitrary Turing machines to represent player strategies and discounting the payoff by the computation time. They show the existence of program equilibria to achieve arbitrarily close payoffs to any purely individually rational correlated play. Both models use programs as a means to formalize *a player's commitment to the reasoning of strategies* in a game before the strategies are determined. By comparison, this work uses programs to define *hierarchical strategy spaces* based on player capabilities.

In repeated or dynamic games, automata have been used to model the complexity of a player's strategy [ASR13], bounded rationality [Ney85; Rub86; Pap94], and learning behavior [CK09; Ghn+06]. Papadimitriou [Pap92] distinguishes between *design complexity* and *implementation complexity*, and shows that limiting the latter (by limiting the number of states of automata representing strategies in a repeated prisoner's dilemma) increase the former (from P to NP-complete in finding an automaton). Their finding is in line with our result that imposing a distance bound makes the symmetric network congestion game PLS-complete. Neyman [Ney97] presents asymptotic results on equilibrium payoff in repeated

normal-form games when automaton sizes meet certain conditions. In contrast, this work focuses on formalizing a hierarchy of player capabilities in one-shot games and analyzing the capability transfer properties precisely instead of asymptotically.

Most results in this chapter are based variations of the network congestion game. Since the introduction of congestion games by Rosenthal [Ros73], they have been applied in many areas including drug design [NIT18], load balancing [ZW20], and network design [LWT20]. There is a rich literature on different aspects of congestion games including their computational characteristics [ARV08], efficiency of equilibria [CSS05], and variants such as weighted congestion games [Chr+20] or games with unreliable resources [NT21]. Structural strategy spaces in congestion games have also been explored. Ackermann and Skopalik [AS08] studies a network congestion game where each player has a set of forbidden edges. Chan and Jiang [CJ16] works with a class of congestion games where the resource masks of the strategy space are $\{0, 1\}$-vertices of a polytope. Unlike the current research, the above research has yet to focus on modeling a hierarchy of strategy spaces in congestion games.

Chapter 9

Conclusion

To the best of the my knowledge, the current research is the first systematic attempt to formalize a hierarchy of player capabilities in games and study the associated solution concepts, solvability, and capability transfer properties. This thesis presents a range of results; most of them have been summarized in Chapter 1. Below, I highlight some results and discuss their implications and/or limitations.

New equilibrium concepts The fully-capability-aware limited-perception game model uncovers a new setting where a player needs to reason about the opponents' perceptions and incentives to make strategic decisions to maximize the expected payoff dictated by an unknown true payoff function. This model requires a new formulation of the simultaneous-move equilibrium that consists of response functions of all players (Section 4.2.2). Although the domain of the response functions can be infinite, an approximate equilibrium can still be represented efficiently such that (i) the representation size is independent of the response function's domain size; and (ii) a mixed strategy can be efficiently computed from the representation (Section 4.2.3).

Capability transfer properties There is no simple characterization of capability transfer properties. Section 4.4 presents a few results on the capability transfer properties for fully-capability-aware limited-perception games. In particular, if a two-player game is maximin-attainable, then the game is pessimistically capability-positive for the lower-capability player for both Nash and Stackelberg equilibrium. However, in both capability-oblivious-limited-perception and limited-strategy games, for some game models examined in this research, determining whether a player gets a better payoff by increasing their capability level is at least as computationally hard as solving the game (Sections 3.4.1 and 8.2.2), which is unlikely to be polynomial-time solvable. One implication is that both game models have instances that are capability-positive and instances that are not. Therefore, it's not always better to have a higher capability level for a player with limited capabilities. The hardness also implies that in some scenarios, even if a player has a complete description of a game (e.g., the network configuration in a McDncDa game), it's still computationally challenging (likely practically infeasible) to decide whether they should increase their capability level. Since increasing the capability level can be seen as a form of exploration, the hardness of determining the effectiveness of such exploration provides another perspective on the exploration-exploitation

dilemma that arises in multiple research areas [Ber+14]. I also present concrete examples of fully-capability-aware games (see Example 4.6 for a zero-sum limited-perception game and Example 8.2 for a McDncDa game) where a player gets a worse payoff by increasing their capability level. An informal explanation is that in the fully-capability-aware setting where the capability profile is public knowledge, increasing one's capability may trigger a defense mechanism from the opponent that leads to a worse outcome. In the capability-oblivious setting, Example 3.1 presents a game where the change of payoff of a player can be easily manipulated by choosing game parameters. More refined characterizations of the capability transfer properties under various settings are left as open problems.

Regarding the limited-rank game, some interesting observations are made via numerical experiments (Section 5.3). If you allow me to philosophize a bit, the results in the capability-oblivious setting can be seen as a justification for egalitarianism from a self-interested perspective. In a society with selfish and capability-oblivious agents such that two agents interact by playing a one-shot bimatrix game with independent random payoffs, an agent should match their opponent's capability level to maximize their own average payoff; all agents should increase their capability levels simultaneously to improve their average payoffs. By contrast, in a totally competitive setting (i.e., zero-sum games), an agent should try to maximize their capability level to get the best average payoff. However, there are individual games that contradict these general trends. Analytically characterizing the distribution of games where such trends hold or break is left as an open problem.

Computational aspects From a computational perspective, limited-perception and limited-strategy games are both harder to solve than their unrestricted counterparts. Verifying a Nash equilibrium in a fully-capability-aware limited-perception game is NP-hard, compared to polynomial-time verifiability and PPAD-complete solvability of normal-form games. Solving a DncDa game is PLS-complete, compared to a symmetric network congestion game whose solution is in P. I also give positive results by identifying three classes of zero-sum fully-capability-aware games that can be solved efficiently via nonsmooth convex optimization. I further present a new algorithm, TRAFS, that solves nonsmooth convex optimization problems faster than previous methods. However, there lack practical algorithms to solve general-sum limited-rank or even more general classes of limited-perception games numerically; by practical, I mean algorithms that can solve certain nontrivial instances in a reasonable amount of time while being numerically robust. Future work may focus on developing practical algorithms to solve these games.

More game models This thesis has investigated capability-oblivious and fully-capability-aware one-shot non-cooperative games. There are many other settings to explore. For example, one may consider dynamic games where players learn about each other's (and their own) capabilities over time. One may also study cooperative games where players with different capabilities form coalitions. Another direction is to consider dynamic games with nonstationary capabilities, where a player can change their capability level (at some cost) to alter the dynamics of the future game. Those games are likely to be more complex and challenging to analyze than the games studied in this thesis, but they may provide new insights into the strategic interactions among agents with different capabilities in more

realistic scenarios. I hope this thesis can inspire further research in related directions.

References

[BKS50] HF Bohnenblust, S Karlin, and LS Shapley. "Solutions of discrete, two-person games". In: *Contributions to the Theory of Games* 1 (1950), pp. 51–72.

[Kuh50] Harold W Kuhn. "A simplified two-person poker". In: *Contributions to the Theory of Games* 1.97-103 (1950), p. 2.

[Nas50] John F Nash Jr. "Equilibrium points in n-person games". In: *Proceedings of the national academy of sciences* 36.1 (1950), pp. 48–49.

[Bro51] George W Brown. "Iterative solution of games by fictitious play". In: *Act. Anal. Prod Allocation* 13.1 (1951), p. 374.

[Nas51] John Nash. "Non-cooperative games". In: *Annals of mathematics* (1951), pp. 286–295.

[Rob51] Julia Robinson. "An iterative method of solving a game". In: *Annals of mathematics* (1951), pp. 296–301.

[Fan52] Ky Fan. "Fixed-point and minimax theorems in locally convex topological linear spaces". In: *Proceedings of the National Academy of Sciences* 38.2 (1952), pp. 121–126.

[Tar55] Alfred Tarski. "A lattice-theoretical fixpoint theorem and its applications." In: *Pacific Journal of Mathematics* 5.2 (1955), pp. 285–309.

[Bau58] Heinz Bauer. "Minimalstellen von funktionen und extremalpunkte". In: *Archiv der Mathematik* 9.4 (1958), pp. 389–393.

[Mir60] Leon Mirsky. "Symmetric gauge functions and unitarily invariant norms". In: *The quarterly journal of mathematics* 11.1 (1960), pp. 50–59.

[Wen62] James G Wendel. "A problem in geometric probability". In: *Mathematica Scandinavica* 11.1 (1962), pp. 109–111.

[LH64] Carlton E Lemke and Joseph T Howson Jr. "Equilibrium points of bimatrix games". In: *Journal of the Society for industrial and Applied Mathematics* 12.2 (1964), pp. 413–423.

[Sha64] Lloyd Shapley. "Some topics in two-person games". In: *Advances in game theory* 52 (1964), pp. 1–29.

[Ros65] J. B. Rosen. "Existence and Uniqueness of Equilibrium Points for Concave N-Person Games". In: *Econometrica* 33.3 (1965), pp. 520–534. ISSN: 00129682, 14680262. (Visited on 03/04/2024).

[Har67] John C Harsanyi. "Games with incomplete information played by "Bayesian" players, I–III Part I. The basic model". In: *Management science* 14.3 (1967), pp. 159–182.

[VM67] John Von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Science editions. J. Wiley, 1967. ISBN: 9780471911852.

[Sim69] Herbert A. Simon. *The Sciences of the Artificial*. Karl Taylor Compton lectures. M.I.T. Press, 1969. ISBN: 9780262690232.

[Roc70] Ralph Tyrell Rockafellar. *Convex Analysis*. Princeton: Princeton University Press, 1970. ISBN: 9781400873173.

[BM71] Dimitri P Bertsekas and Sanjoy K Mitter. "Steepest descent for optimization problems with nondifferentiable cost functionals". In: *Proc. 5th Annual Princeton Confer. Inform. Sci. Systems, Princeton, NJ* (1971), pp. 347–351.

[Ros73] Robert W Rosenthal. "A class of games possessing pure-strategy Nash equilibria". In: *International Journal of Game Theory* 2.1 (1973), pp. 65–67.

[CDW75] Jane Cullum, Wilm E Donath, and Philip Wolfe. "The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices". In: *Nondifferentiable optimization* (1975), pp. 35–55.

[Lem78] Claude Lemarechal. "Nonsmooth optimization and descent methods". In: (1978).

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Mathematical Sciences Series. Freeman, 1979. ISBN: 9780716710448.

[Man79] O.L. Mangasarian. "Uniqueness of solution in linear programming". In: *Linear Algebra and its Applications* 25 (1979), pp. 151–162. ISSN: 0024-3795. DOI: https://doi.org/10.1016/0024-3795(79)90014-4.

[Mif82] Robert Mifflin. *A modification and an extension of Lemaréchal's algorithm for nonsmooth minimization*. Springer, 1982.

[NY83] A.S. Nemirovskiĭ and D.B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. A Wiley-Interscience publication. Wiley, 1983. ISBN: 9780471103455.

[Nes83] Yurii Nesterov. "A method of solving a convex programming problem with convergence rate O (1/k** 2)". In: *Doklady Akademii Nauk SSSR* 269.3 (1983), p. 543.

[BD84] S.K. Berberian and J. Dixmier. *General Topology*. Undergraduate Texts in Mathematics. Springer New York, 1984. ISBN: 9781441928238.

[Kar84] Narendra Karmarkar. "A new polynomial-time algorithm for linear programming". In: *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. 1984, pp. 302–311.

[Ney85] Abraham Neyman. "Bounded complexity justifies cooperation in the finitely repeated prisoners' dilemma". In: *Economics letters* 19.3 (1985), pp. 227–229.

[Rub86] Ariel Rubinstein. "Finite automata play the repeated prisoner's dilemma". In: *Journal of Economic Theory* 39.1 (1986), pp. 83–96. ISSN: 0022-0531.

[AR88] Dilip Abreu and Ariel Rubinstein. "The structure of Nash equilibrium in repeated games with finite automata". In: *Econometrica: Journal of the Econometric Society* (1988), pp. 1259–1281.

[SB88] Reinhard Selten and R Selten Bielefeld. *Reexamination of the perfectness concept for equilibrium points in extensive games*. Springer, 1988.

[HPV89] Michael D Hirsch, Christos H Papadimitriou, and Stephen A Vavasis. "Exponential lower bounds for finding brouwer fix points". In: *Journal of Complexity* 5.4 (1989), pp. 379–416.

[HJS92] Pierre Hansen, Brigitte Jaumard, and Gilles Savard. "New branch-and-bound rules for linear bilevel programming". In: *SIAM Journal on scientific and Statistical Computing* 13.5 (1992), pp. 1194–1217.

[Pap92] Christos H Papadimitriou. "On players with a bounded number of states". In: *Games and Economic Behavior* 4.1 (1992), pp. 122–131. ISSN: 0899-8256.

[ECK93] Jürgen ECKHOFF. "CHAPTER 2.1 - Helly, Radon, and Carathéodory Type Theorems". In: *Handbook of Convex Geometry*. Ed. by P.M. GRUBER and J.M. WILLS. Amsterdam: North-Holland, 1993, pp. 389–448. ISBN: 978-0-444-89596-7. DOI: 10.1016/B978-0-444-89596-7.50017-1.

[Pap94] Christos H Papadimitriou. "On the complexity of the parity argument and other inefficient proofs of existence". In: *Journal of Computer and system Sciences* 48.3 (1994), pp. 498–532.

[LNN95] Claude Lemaréchal, Arkadii Nemirovskii, and Yurii Nesterov. "New variants of bundle methods". In: *Mathematical programming* 69 (1995), pp. 111–147.

[Sim95] Stephen Simons. "Minimax Theorems and Their Proofs". In: *Minimax and Applications*. Ed. by Ding-Zhu Du and Panos M. Pardalos. Boston, MA: Springer US, 1995, pp. 1–23. ISBN: 978-1-4613-3557-3. DOI: 10.1007/978-1-4613-3557-3_1.

[Heu+96] Robert van Heumen, Bezalel Peleg, Stef Tijs, and Peter Borm. "Axiomatic characterizations of solutions for Bayesian games". In: *Theory and Decision* 40 (1996), pp. 103–129.

[HL96a] J.B. Hiriart-Urruty and C. Lemarechal. *Convex Analysis and Minimization Algorithms I: Fundamentals*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 1996. ISBN: 9783540568506.

[HL96b] J.B. Hiriart-Urruty and C. Lemarechal. *Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 1996. ISBN: 9783540568520.

[LPR96] Z.Q. Luo, J.S. Pang, and D. Ralph. *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press, 1996. ISBN: 9781316582619.

[MS96a] Dov Monderer and Aner Sela. "A 2×2 game without the fictitious play property". In: *Games and Economic Behavior* 14.1 (1996), pp. 144–148.

[MS96b] Dov Monderer and Lloyd S Shapley. "Fictitious play property for games with identical interests". In: *Journal of economic theory* 68.1 (1996), pp. 258–265.

[MS96c] Dov Monderer and Lloyd S Shapley. "Potential games". In: *Games and economic behavior* 14.1 (1996), pp. 124–143.

[Tib96] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58.1 (1996), pp. 267–288.

[Cla+97] F.H. Clarke, Y.S. Ledyaev, R.J. Stern, and P.R. Wolenski. *Nonsmooth Analysis and Control Theory*. Graduate Texts in Mathematics. Springer New York, 1997. ISBN: 9780387983363.

[FV97] Dean P Foster and Rakesh V Vohra. "Calibrated learning and correlated equilibrium". In: *Games and Economic Behavior* 21.1-2 (1997), pp. 40–55.

[Ney97] Abraham Neyman. "Cooperation, Repetition, and Automata". In: *Cooperation: Game-Theoretic Approaches*. Ed. by Sergiu Hart and Andreu Mas-Colell. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 233–255. ISBN: 978-3-642-60454-6.

[TB97] L.N. Trefethen and D. Bau. *Numerical Linear Algebra*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, 1997. ISBN: 9780898713619.

[Ber98] Dimitri P. Bertsekas. *Network Optimization: Continuous and Discrete Models*. Athena scientific optimization and computation series. Athena Scientific, 1998. ISBN: 9781886529021.

[FV98] Dean P Foster and Rakesh V Vohra. "Asymptotic calibration". In: *Biometrika* 85.2 (1998), pp. 379–390.

[Gol98] R. Goldblatt. *Lectures on the Hyperreals: An Introduction to Nonstandard Analysis*. Graduate Texts in Mathematics. Springer New York, 1998. ISBN: 9780387984643.

[Nes98] Yurii Nesterov. "Introductory lectures on convex programming volume i: Basic course". In: *Lecture notes* 3.4 (1998), p. 5.

[FL99] Drew Fudenberg and David K Levine. "Conditional universal consistency". In: *Games and Economic Behavior* 29.1-2 (1999), pp. 104–130.

[ABB00] Orly Alter, Patrick O Brown, and David Botstein. "Singular value decomposition for genome-wide expression data processing and modeling". In: *Proceedings of the National Academy of Sciences* 97.18 (2000), pp. 10101–10106.

[DDV00] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. "A multilinear singular value decomposition". In: *SIAM journal on Matrix Analysis and Applications* 21.4 (2000), pp. 1253–1278.

[HM00] Sergiu Hart and Andreu Mas-Colell. "A simple adaptive procedure leading to correlated equilibrium". In: *Econometrica* 68.5 (2000), pp. 1127–1150.

[KU00] Jacek B Krawczyk and Stanislav Uryasev. "Relaxation algorithms to find Nash equilibria with economic applications". In: *Environmental Modeling & Assessment* 5 (2000), pp. 63–73.

[NO00] Abraham Neyman and Daijiro Okada. "Repeated games with bounded entropy". In: *Games and Economic Behavior* 30.2 (2000), pp. 228–247.

[BLO02] James V Burke, Adrian S Lewis, and Michael L Overton. "Approximating subdifferentials by random sampling of gradients". In: *Mathematics of Operations Research* 27.3 (2002), pp. 567–584.

[HM03] Sergiu Hart and Andreu Mas-Colell. "Uncoupled dynamics do not lead to Nash equilibrium". In: *American Economic Review* 93.5 (2003), pp. 1830–1836.

[LMM03] Richard J Lipton, Evangelos Markakis, and Aranyak Mehta. "Playing large games using simple strategies". In: *Proceedings of the 4th ACM Conference on Electronic Commerce*. 2003, pp. 36–41.

[Mäk03] Marko M Mäkelä. "Multiobjective proximal bundle method for nonconvex nonsmooth optimization: Fortran subroutine MPBNGC 2.0". In: *Reports of the Department of Mathematical Information Technology, Series B. Scientific Computing, B* 13 (2003), p. 2003.

[SJ03] Nathan Srebro and Tommi Jaakkola. "Weighted low-rank approximations". In: *Proceedings of the 20th international conference on machine learning (ICML-03)*. 2003, pp. 720–727.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. ISBN: 9781107394001.

[Cha04] Antonin Chambolle. "An algorithm for total variation minimization and applications". In: *Journal of Mathematical imaging and vision* 20 (2004), pp. 89–97.

[FPT04] Alex Fabrikant, Christos Papadimitriou, and Kunal Talwar. "The complexity of pure Nash equilibria". In: *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*. 2004, pp. 604–612.

[HMM04] Marjo Haarala, Kaisa Miettinen, and Marko M Mäkelä. "New limited memory bundle method for large-scale nonsmooth optimization". In: *Optimization Methods and Software* 19.6 (2004), pp. 673–692.

[Nem04a] Arkadi Nemirovski. "Interior point polynomial time methods in convex programming". In: *Lecture notes* 42.16 (2004), pp. 3215–3224.

[Nem04b] Arkadi Nemirovski. "Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems". In: *SIAM Journal on Optimization* 15.1 (2004), pp. 229–251.

[Ten04] Moshe Tennenholtz. "Program equilibrium". In: *Games and Economic Behavior* 49.2 (2004), pp. 363–373. ISSN: 0899-8256.

[Ber05] Ulrich Berger. "Fictitious play in $2 \times n$ games". In: *Journal of Economic Theory* 120.2 (2005), pp. 139–154.

[BLO05] James V Burke, Adrian S Lewis, and Michael L Overton. "A robust gradient sampling algorithm for nonsmooth, nonconvex optimization". In: *SIAM Journal on Optimization* 15.3 (2005), pp. 751–779.

[CSS05] José R Correa, Andreas S Schulz, and Nicolás E Stier-Moses. "On the inefficiency of equilibria in congestion games". In: *International Conference on Integer Programming and Combinatorial Optimization*. Springer. 2005, pp. 167–181.

[AB06] Michele Aghassi and Dimitris Bertsimas. "Robust game theory". In: *Mathematical programming* 107.1 (2006), pp. 231–273.

[BKK06] Eli Ben-Sasson, Ehud Kalai, and Adam Kalai. "An approach to bounded rationality". In: *Advances in neural information processing systems* 19 (2006).

[Bis06] C.M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer New York, 2006. ISBN: 9781493938438.

[CS06] Vincent Conitzer and Tuomas Sandholm. "Computing the optimal strategy to commit to". In: *Proceedings of the 7th ACM conference on Electronic commerce*. 2006, pp. 82–90.

[FY06] Dean Foster and Hobart Peyton Young. "Regret testing: Learning to play Nash equilibrium without knowing you have an opponent". In: *Theoretical Economics* 1.3 (2006), pp. 341–367.

[Ghn+06] Rawan Ghnemat, Saleh Oqeili, Cyrille Bertelle, and Gérard HE Duchamp. "Automata-based adaptive behavior for economic modelling using game theory". In: *Emergent properties in natural and artificial dynamical systems*. Springer, 2006, pp. 171–183.

[HM06] Sergiu Hart and Andreu Mas-Colell. "Stochastic uncoupled dynamics and Nash equilibrium". In: *Games and economic behavior* 57.2 (2006), pp. 286–303.

[Kiw06] K.C. Kiwiel. *Methods of Descent for Nondifferentiable Optimization*. Lecture Notes in Mathematics. Springer Berlin Heidelberg, 2006. ISBN: 9783540395096.

[Lay07] S.R. Lay. *Convex Sets and Their Applications*. Dover Books on Mathematics Series. Dover Publications, 2007. ISBN: 9780486458038.

[Nis+07] N. Nisan, T. Roughgarden, E. Tardos, and V.V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007. ISBN: 9780521872829.

[SL07] Gilles Stoltz and Gábor Lugosi. "Learning correlated equilibria in games with compact sets of strategies". In: *Games and Economic Behavior* 59.1 (2007), pp. 187–208.

[ARV08] Heiner Ackermann, Heiko Röglin, and Berthold Vöcking. "On the impact of combinatorial structure on congestion games". In: *Journal of the ACM (JACM)* 55.6 (2008), pp. 1–22.

[AS08] Heiner Ackermann and Alexander Skopalik. "Complexity of pure Nash equilibria in player-specific network congestion games". In: *Internet Mathematics* 5.4 (2008), pp. 323–342.

[Par+08] Praveen Paruchuri, Jonathan P Pearce, Janusz Marecki, Milind Tambe, Fernando Ordonez, and Sarit Kraus. "Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games". In: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*. 2008, pp. 895–902.

[Aga+09] Alekh Agarwal, Martin J Wainwright, Peter Bartlett, and Pradeep Ravikumar. "Information-theoretic lower bounds on the oracle complexity of convex optimization". In: *Advances in Neural Information Processing Systems* 22 (2009).

[BGA+09] Nicola Basilico, Nicola Gatti, Francesco Amigoni, et al. "Leader-follower strategies for robotic patrolling in environments with arbitrary topologies". In: *Proceedings of the International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS)*. 2009, pp. 57–64.

[BT09] Amir Beck and Marc Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". In: *SIAM journal on imaging sciences* 2.1 (2009), pp. 183–202.

[CDT09] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. "Settling the complexity of computing two-player Nash equilibria". In: *Journal of the ACM (JACM)* 56.3 (2009), pp. 1–57.

[CK09] Raymond Chiong and Michael Kirley. "Co-evolutionary learning in the N-player iterated prisoner's dilemma with a structured environment". In: *Australian Conference on Artificial Life*. Springer. 2009, pp. 32–42.

[For09] Lance Fortnow. "Program Equilibria and Discounted Computation Time". In: *Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge*. TARK '09. California: Association for Computing Machinery, 2009, pp. 128–133. ISBN: 9781605585604.

[KBV09] Yehuda Koren, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems". In: *Computer* 42.8 (2009), pp. 30–37.

[HL10] Jason D Hartline and Brendan Lucier. "Bayesian algorithmic mechanism design". In: *Proceedings of the forty-second ACM symposium on Theory of computing*. 2010, pp. 301–310.

[KTM10] Christopher Kiekintveld, Milind Tambe, and Janusz Marecki. "Robust Bayesian methods for Stackelberg security games". In: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*. 2010, pp. 1467–1468.

[Sta+10] H. von Stackelberg, D. Bazin, R. Hill, and L. Urch. *Market Structure and Equilibrium*. Springer Berlin Heidelberg, 2010. ISBN: 9783642125867.

[CP11] Antonin Chambolle and Thomas Pock. "A first-order primal-dual algorithm for convex problems with applications to imaging". In: *Journal of mathematical imaging and vision* 40 (2011), pp. 120–145.

[Bab12] Yakov Babichenko. "Completely uncoupled dynamics and Nash equilibria". In: *Games and Economic Behavior* 76.1 (2012), pp. 1–14.

[Con12] Vincent Conitzer. "Computing game-theoretic solutions and applications to security". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 26. 1. 2012, pp. 2106–2112.

[NVD12] Ann Nowé, Peter Vrancx, and Yann-Michaël De Hauwere. "Game theory and multi-agent reinforcement learning". In: *Reinforcement Learning: State-of-the-Art* (2012), pp. 441–470.

[Sip12] M. Sipser. *Introduction to the Theory of Computation*. Cengage Learning, 2012. ISBN: 9781133187790.

[Aho+13] A.V. Aho, M.S. Lam, R. Sethi, and J.D. Ullman. *Compilers: Principles, Techniques, and Tools: Pearson New International Edition*. Pearson Education, 2013. ISBN: 9781292037233.

[AB13] C.D. Aliprantis and K.C. Border. *Infinite Dimensional Analysis: A Hitchhiker's Guide*. Springer Berlin Heidelberg, 2013. ISBN: 9783662039618.

[ASR13] Sally Almanasra, Khaled Suwais, and Muhammad Rafie. "The applications of automata in game theory". In: *Intelligent Technologies and Techniques for Pervasive Computing*. IGI Global, 2013, pp. 204–217.

[CQ13] Frank E Curtis and Xiaocun Que. "An adaptive gradient sampling algorithm for non-smooth optimization". In: *Optimization Methods and Software* 28.6 (2013), pp. 1302–1324.

[Dev13] L. Devroye. *Non-Uniform Random Variate Generation*. Springer New York, 2013. ISBN: 9781461386452.

[FLM13] Jianqing Fan, Yuan Liao, and Martina Mincheva. "Large covariance estimation by thresholding principal orthogonal complements". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 75.4 (2013), pp. 603–680.

[Gro+13] Michael Großhans, Christoph Sawade, Michael Brückner, and Tobias Scheffer. "Bayesian games for adversarial regression problems". In: *International conference on machine learning*. PMLR. 2013, pp. 55–63.

[KIK13] Christopher Kiekintveld, Towhidul Islam, and Vladik Kreinovich. "Security games with interval uncertainty". In: *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*. AAMAS '13. St. Paul, MN, USA: International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 231–238. ISBN: 9781450319935.

[MZS13] M. Maschler, S. Zamir, and E. Solan. *Game Theory*. Cambridge University Press, 2013. ISBN: 9781107005488.

[BKM14] A. Bagirov, N. Karmitsa, and M.M. Mäkelä. *Introduction to Nonsmooth Optimization: Theory, Practice and Software*. Springer International Publishing, 2014. ISBN: 9783319081144.

[Ber+14] Oded Berger-Tal, Jonathan Nathan, Ehud Meron, and David Saltz. "The exploration-exploitation dilemma: a multidisciplinary framework". In: *PloS one* 9.4 (2014), e95693.

[HMU14] J.E. Hopcroft, R. Motwani, and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Always learning. Pearson Education, 2014. ISBN: 9781292039053.

[IN14] Anatoli Iouditski and Yuri Nesterov. "Primal-dual subgradient methods for minimizing uniformly convex functions". In: *arXiv preprint arXiv:1401.1792* (2014).

[Meh14] Ruta Mehta. "Constant rank bimatrix games are PPAD-hard". In: *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*. 2014, pp. 545–554.

[Ngu+14] Thanh Nguyen, Amulya Yadav, Bo An, Milind Tambe, and Craig Boutilier. "Regret-based optimization and preference elicitation for Stackelberg security games with uncertainty". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 28. 1. 2014.

[PB+14] Neal Parikh, Stephen Boyd, et al. "Proximal algorithms". In: *Foundations and trends® in Optimization* 1.3 (2014), pp. 127–239.

[Bub+15] Sébastien Bubeck et al. "Convex optimization: Algorithms and complexity". In: *Foundations and Trends® in Machine Learning* 8.3-4 (2015), pp. 231–357.

[HP15] Joseph Y Halpern and Rafael Pass. "Algorithmic rationality: Game theory with costly computation". In: *Journal of Economic Theory* 156 (2015), pp. 246–268.

[NS15] Yu Nesterov and Vladimir Shikhman. "Quasi-monotone subgradient methods for nonsmooth convex minimization". In: *Journal of Optimization Theory and Applications* 165.3 (2015), pp. 917–940.

[CJ16] Hau Chan and Albert Xin Jiang. "Congestion Games with Polytopal Strategy Spaces." In: *IJCAI*. 2016, pp. 165–171.

[DB16] Steven Diamond and Stephen Boyd. "CVXPY: A Python-embedded modeling language for convex optimization". In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5.

[MKW16] Marko M Mäkelä, Napsu Karmitsa, and Outi Wilppu. "Proximal bundle method for nonsmooth and nonconvex multiobjective optimization". In: *Mathematical modeling and optimization of complex structures* (2016), pp. 191–204.

[PW16] Chayne Planiden and Xianfu Wang. "Most Convex Functions Have Unique Minimizers". In: *Journal of Convex Analysis* 23.3 (2016), pp. 877–892.

[Rou16] T. Roughgarden. *Twenty Lectures on Algorithmic Game Theory*. Cambridge University Press, 2016. ISBN: 9781107172661.

[AO17] Zeyuan Allen-Zhu and Lorenzo Orecchia. "Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent". In: *Proceedings of the 8th Innovations in Theoretical Computer Science*. ITCS '17. Full version available at http://arxiv.org/abs/1407.1537. 2017.

[Bec17] A. Beck. *First-Order Methods in Optimization*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2017. ISBN: 9781611974980.

[CMG17] Andrea Celli, Alberto Marchesi, and Nicola Gatti. "On the Complexity of Nash Equilibrium Reoptimization". In: *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*. Ed. by Gal Elidan, Kristian Kersting, and Alexander Ihler. AUAI Press, 2017.

[HSS17] Elias Salomão Helou, Sandra A Santos, and Lucas EA Simões. "On the local convergence analysis of the gradient sampling method for finite max-functions". In: *Journal of Optimization Theory and Applications* 175 (2017), pp. 137–157.

[Far+18] Gabriele Farina, Alberto Marchesi, Christian Kroer, Nicola Gatti, and Tuomas Sandholm. "Trembling-hand perfection in extensive-form games with commitment". In: *International Joint Conferences on Artificial Intelligence Organization*. 2018.

[KFS18] Christian Kroer, Gabriele Farina, and Tuomas Sandholm. "Robust Stackelberg equilibria in extensive-form games and extension to limited lookahead". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[NIT18] Natalia Nikitina, Evgeny Ivashko, and Andrei Tchernykh. "Congestion game scheduling for virtual drug screening optimization". In: *Journal of computer-aided molecular design* 32.2 (2018), pp. 363–374.

[Sub18] P.V. Subrahmanyam. *Elementary Fixed Point Theorems*. Forum for Interdisciplinary Mathematics. Springer Singapore, 2018. ISBN: 9789811331572 9789811331589.

[Wea18] N. Weaver. *Lipschitz Algebras (Second Edition)*. World Scientific Publishing Company, 2018. ISBN: 9789814740654.

[Che19] AV Chernov. "On some approaches to find Nash equilibrium in concave games". In: *Automation and Remote Control* 80 (2019), pp. 964–988.

[RSH19] Aseem Rastogi, Nikhil Swamy, and Michael Hicks. "WYS*: A DSL for Verified Secure Multi-party Computations". In: *International Conference on Principles of Security and Trust*. Springer. 2019, pp. 99–122.

[VG19] Mehrdad Valizadeh and Amin Gohari. "Playing games with bounded entropy". In: *Games and Economic Behavior* 115 (2019), pp. 363–380.

[Bag+20] A.M. Bagirov, M. Gaudioso, N. Karmitsa, M.M. Mäkelä, and S. Taheri. *Numerical Nonsmooth Optimization: State of the Art Algorithms*. Springer International Publishing, 2020. ISBN: 9783030349103.

[Bur+20] James V Burke, Frank E Curtis, Adrian S Lewis, Michael L Overton, and Lucas EA Simões. "Gradient sampling methods for nonsmooth optimization". In: *Numerical nonsmooth optimization: State of the art algorithms* (2020), pp. 201–225.

[Chr+20] George Christodoulou, Martin Gairing, Yiannis Giannakopoulos, Diogo Poças, and Clara Waldmann. "Existence and Complexity of Approximate Equilibria in Weighted Congestion Games". In: *ICALP*. 2020.

[Kar20] Napsu Karmitsa. "Limited memory bundle method and its variations for large-scale nonsmooth optimization". In: *Numerical Nonsmooth Optimization: State of the Art Algorithms* (2020), pp. 167–199.

[LWT20] Shuting Le, Yuhu Wu, and Mitsuru Toyoda. "A congestion game framework for service chain composition in NFV with function benefit". In: *Information Sciences* 514 (2020), pp. 512–522.

[LJJ20] Tianyi Lin, Chi Jin, and Michael I Jordan. "Near-optimal algorithms for minimax optimization". In: *Conference on Learning Theory*. PMLR. 2020, pp. 2738–2779.

[Ped20] Niels C. Pedersen. *Summary of GS-441524 Treatment of Feline Infectious Peritonitis*. 2020. URL: https://ccah.vetmed.ucdavis.edu/sites/g/files/dgvnsk4586/files/inline-files/Summary%20of%20GS-441524%20treatment%20v5.pdf (visited on 07/19/2024).

[ZS20] Brian Zhang and Tuomas Sandholm. "Sparsified linear programming for zero-sum equilibrium finding". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 11256–11267.

[ZW20] Fenghui Zhang and Michael Mao Wang. "Stochastic congestion game for load balancing in mobile-edge computing". In: *IEEE Internet of Things Journal* 8.2 (2020), pp. 778–790.

[Che+21] Xu Chen, Wei Feng, Yantian Luo, Meng Shen, Ning Ge, and Xianbin Wang. "Defending against link flooding attacks in internet of things: A bayesian game approach". In: *IEEE Internet of Things Journal* 9.1 (2021), pp. 117–128.

[Cou21] Patrick Cousot. *Principles of Abstract Interpretation*. MIT Press, 2021. ISBN: 9780262044905.

[Das21] Constantinos Daskalakis. "Non-concave games: A challenge for game theory's next 100 years". In: *Nobel symposium" One Hundred Years of Game Theory: Future Applications and Challenges*. 2021.

[DFG21] Constantinos Costis Daskalakis, Maxwell Fishelson, and Noah Golowich. "Near-Optimal No-Regret Learning in General Games". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan. 2021.

[GC21] Paul Goulart and Yuwen Chen. *Clarabel: A Library for Optimization and Control*. 2021. URL: https://oxfordcontrol.github.io/ClarabelDocs/stable.

[Han+21] Zhao Han, Shahzaib Qazi, Michael Werner, Keerthikumara Devarajegowda, and Wolfgang Ecker. "On Self-Verifying DSL Generation for Embedded Systems Automation". In: *MBMV 2021; 24th Workshop*. VDE. 2021, pp. 1–7.

[Mor+21] Dustin Morrill, Ryan D'Orazio, Reca Sarfati, Marc Lanctot, James R Wright, Amy R Greenwald, and Michael Bowling. "Hindsight and sequential rationality of correlated play". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 6. 2021, pp. 5584–5594.

[NT21] Julian Nickerl and Jacobo Torán. "Pure Nash Equilibria in a Generalization of Congestion Games Allowing Resource Failures". In: *International Symposium on Algorithmic Game Theory*. Springer. 2021, pp. 186–201.

[Say+21] Muhammed Sayin, Kaiqing Zhang, David Leslie, Tamer Basar, and Asuman Ozdaglar. "Decentralized Q-learning in zero-sum Markov games". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 18320–18334.

[XHZ21] Guangzeng Xie, Yuze Han, and Zhihua Zhang. *DIPPA: An improved Method for Bilinear Saddle Point Problems*. 2021. arXiv: 2103.08270 [cs.LG].

[Far+22a] Gabriele Farina, Ioannis Anagnostides, Haipeng Luo, Chung-Wei Lee, Christian Kroer, and Tuomas Sandholm. "Near-optimal no-regret learning dynamics for general convex games". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 39076–39089.

[Far+22b] Gabriele Farina, Andrea Celli, Alberto Marchesi, and Nicola Gatti. "Simple uncoupled no-regret learning dynamics for extensive-form correlated equilibrium". In: *Journal of the ACM* 69.6 (2022), pp. 1–41.

[FS22] Gabriele Farina and Tuomas Sandholm. "Fast payoff matrix sparsification techniques for structured extensive-form games". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 5. 2022, pp. 4999–5007.

[YJR22] Yichen Yang, Kai Jia, and Martin Rinard. "On the Impact of Player Capability on Congestion Games". In: *Algorithmic Game Theory*. Ed. by Panagiotis Kanellopoulos, Maria Kyropoulou, and Alexandros Voudouris. Cham: Springer International Publishing, 2022, pp. 311–328. ISBN: 978-3-031-15714-1.

[DG23] Mateo Díaz and Benjamin Grimmer. "Optimal convergence rates for the proximal bundle method". In: *SIAM Journal on Optimization* 33.2 (2023), pp. 424–454.

[Far23] Gabriele Farina. *Lecture 12: Introduction to Extensive-Form Games*. Slides of 6.S890: Topics in Multiagent Learning. Massachusetts Institute of Technology, 2023. URL: https://web.archive.org/web/20240415194734/https://web.mit.edu/~gfarina/www/6S890/lecture12.pdf.

[Mil23] Johannes Milz. *MPBNGCInterface.jl: Julia package interfacing the Fortran code MPBNGC*. 2023. URL: https://github.com/milzj/MPBNGCInterface.jl.

[OR23] Edan Orzech and Martin Rinard. "Randomness Requirements and Asymmetries in Nash Equilibria". In: *arXiv preprint arXiv:2312.17364* (2023).

[PVZ23] Christos Papadimitriou, Emmanouil-Vasileios Vlatakis-Gkaragkounis, and Manolis Zampetakis. "The Computational Complexity of Multi-Player Concave Games and Kakutani Fixed Points". In: *Proceedings of the 24th ACM Conference on Economics and Computation*. EC '23. London, United Kingdom: Association for Computing Machinery, 2023, p. 1045. ISBN: 9798400701047.

[Sch+23] Roland Schwan, Yuning Jiang, Daniel Kuhn, and Colin N. Jones. *PIQP: A Proximal Interior-Point Quadratic Programming Solver*. 2023. eprint: arXiv:2304.00290.

[WB23] Andrzej Wąsowski and Thorsten Berger. *Domain-Specific Languages: Effective modeling, automation, and reuse*. Springer, 2023. ISBN: 9783031236693.

[Zwi+23] Katharina Zwicklbauer, Daniela Krentz, Michèle Bergmann, Sandra Felten, Roswitha Dorsch, Andrea Fischer, Regina Hofmann-Lehmann, Marina L Meli, Andrea M Spiri, Martin Alberer, et al. "Long-term follow-up of cats in complete remission after treatment of feline infectious peritonitis with oral GS-441524". In: *Journal of Feline Medicine and Surgery* 25.8 (2023), p. 1098612X231183250.

[ZC23] Dan Zwillinger and Paul San Clemente. "Game theory analysis when playing the wrong game". In: *arXiv preprint arXiv:2307.10257* (2023).

[JR24a] Kai Jia and Martin Rinard. *Limited-perception games*. 2024. arXiv: 2405.16735.

[JR24b] Kai Jia and Martin Rinard. *TRAFS: A Nonsmooth Convex Optimization Algorithm with $\mathcal{O}\left(\frac{1}{\epsilon}\right)$ Iteration Complexity*. 2024. arXiv: 2311.06205 [math.OC].

[MMT24] Richard D. McKelvey, Andrew M. McLennan, and Theodore L. Turocy. *Gambit: Software Tools for Game Theory*. Version 16.1.1. 2024. URL: http://www.gambit-project.org.