POLITECNICO MILANO 1863

POLITECNICO DI MILANO
DIPARTIMENTO DI ELETTRONICA, INFORMAZIONE E BIOINGEGNERIA
DOCTORAL PROGRAM IN INFORMATION TECHNOLOGY

LEADERSHIP GAMES: MULTIPLE FOLLOWERS, MULTIPLE LEADERS, AND PERFECTION

Doctoral Dissertation of:
Alberto Marchesi

Supervisor:
Prof. Nicola Gatti

Tutor:
Prof. Cesare Alippi

The Chair of the Doctoral Program:
Prof. Barbara Pernici

2019 – Cycle XXXII

Abstract

Over the last years *algorithmic game theory* has received growing interest in AI, as it allows to tackle complex real-world scenarios involving multiple artificial agents engaged in a competitive interaction. These settings call for rational agents endowed with the capability of reasoning *strategically*, *i.e.*, taking into account not only how their actions affect the external environment, but also their impact on the behavior of other agents. This is achieved by exploiting ideas from game theory, and, in particular, *equilibrium* concepts that prescribe the agents how to behave strategically. Thus, the challenge faced by the researchers working in algorithmic game theory is to design scalable computational tools that enable the adoption of such equilibrium notions in real-world problems.

In this thesis, we study the computational properties of a specific game-theoretic model known as the *Stackelberg paradigm*. In a Stackelberg game, there are some players who act as *leaders* with the ability to commit to a strategy beforehand, whereas the other players are *followers* who decide how to play after observing the commitment. Recently, Stackelberg games and the corresponding Stackelberg equilibria have received considerable attention from the algorithmic game theory community, since they have been successfully applied in many real-world settings, such as, *e.g.*, in the security domain, toll-setting problems, and network routing. Nevertheless, the majority of the computational works on Stackelberg games study the case in which there is one leader and one follower, focusing, in most of the cases, on instances enjoying very specific structures, such as security games. A comprehensive study of general Stackelberg games with

(possibly) multiple leaders and followers is still lacking.

In this thesis, we make substantial steps towards filling this gap. In particular, in the first part of the work, we address the largely unexplored problem of computing Stackelberg equilibria in games with a *single leader* and *multiple followers*, focusing on the case in which the latter play a *Nash equilibrium* after observing the leader's commitment. We analyze different classes of games, from general *normal-form* Stackelberg games to games with a compact representation, namely, Stackelberg *polymatrix* and *congestion* games. Then, in the second part of the thesis, we study Stackelberg games with *multiple leaders*, proposing a new way to apply the Stackelberg paradigm in such settings. Our idea is to let the leaders decide whether they want to participate in the commitment or defect from it by becoming followers. This is orchestrated by a suitably defined agreement protocol, which allows us to introduce interesting properties for the commitments. Finally, in the last part of the thesis, we focus on Stackelberg games with a sequential structure, addressing, for the first time in such setting, the problem of equilibrium refinement. This problem has been widely investigated for the Nash equilibrium, as it is well-known that refinements can amend some of its weaknesses, such as sub-optimality off the equilibrium path. In this work, we show that such issues also arise in Stackelberg settings, and, thus, we introduce and study Stackelberg equilibrium refinements based on the idea of *trembling-hand perfection* so as to solve them.

Contents

CHAPTER 1

Introduction

Since *artificial intelligence* (AI) was officially founded during the workshop held at Dartmouth in the summer of 1956, most of the research in the field has focused on how to design artificial agents endowed with *rational* behaviors. Nowadays, one of the biggest challenges in AI research is to build rational agents that are not only able to interact with an external environment, but they can also handle more complex interactions involving different actors, such as other artificial agents and human beings.

Over the last two decades, *algorithmic game theory* has received an increasing interest from the AI community, since it allows to tackle complex real-world scenarios where multiple artificial agents are engaged in a competitive interaction. These settings call for rational agents with the ability to reason *strategically*, taking into account not only how their actions affect the external environment, but also how they influence the behaviors of the other agents. These capabilities are achieved by exploiting ideas from economics, specifically microeconomic models called *games*, and their corresponding solutions, usually known as *equilibria*, which have been introduced and studied by game theorists during the last century (see, among others, the widely acclaimed notion of equilibrium defined by Nash (1951), worth him a Nobel prize in economics). The challenge faced by the researchers working in algorithmic game theory is to design scalable computational tools that can deal with these mathematical notions of equilibrium, enabling their adoption for the solution of real-world problems.

The recent advances in the development of equilibrium-finding techniques have lead to the successful application of game-theoretic models in real-world settings. For instance, game theory has been extensively adopted in security domains, with the goal of devising protection strategies which are robust against strategic attackers (Tambe, 2011). Other application domains are found in the *Internet*, where interactions involving multiple strategic agents naturally arise, given the intrinsic distributed nature of the network. One examples is, among others, the problem of designing auction mechanisms for web advertising (Gatti et al., 2015; Farina and Gatti, 2017b). Moreover, great achievements have been made towards the development of artificial agents capable of beating human professional in large two-player zero-sum recreational games like Chess (Campbell et al., 2002), Go (Silver et al., 2016), and Poker (Brown and Sandholm, 2018, 2019).

Despite the great attention devoted to algorithmic game theory in the last years, the majority of the works in the literature study (relatively) simple settings involving only two players with opposite objectives, *i.e.*, two-player zero-sum games. In such models, there is a clear and well-established definition of solution, in which each player aims to maximize her utility given that the opponent acts so as to minimize it. In zero-sum games, this definition corresponds to that of *Nash equilibrium*. Thus, considerable efforts have been devoted to studying the problem of computing (possibly approximate) Nash equilibria in such settings. Instead, more complex games where there are more than two players and/or arbitrary, *i.e.*, general-sum, utilities are widely unexplored. In such scenarios, there is no clear definition of solution to a game, as this strongly depends on the specific application that one wish to represent. As a result, many solution concepts other than the Nash equilibrium have been introduced and studied. However, there is still a lot of work to be done on the computational side, as the algorithmic works on multi-player general-sum games are only few.

In this work, we study settings beyond two-player zero-sum games, focusing on a particular game paradigm which leads to the definition of what is known in the literature as the *Stackelberg equilibrium*.

1.1 The Stackelberg Paradigm

The Stackelberg paradigm was originally introduced by von Stackelberg in 1934 to model economic situations where a firm (the *leader*) moves first and, then, another firm (the *follower*) moves second by reacting to the first firm's move (Von Stackelberg, 1934). Recently, this paradigm was brought to new attention by the work of Von Stengel and Zamir (2010), who study a variant of the original Stackelberg paradigm in which the leader commits to a (possibly randomized, *i.e.*, mixed) strategy beforehand, while the follower decides how to play after observing the leader's strategy. In general settings involving multiple players, a *Stackelberg game* is characterized by a group of players who act as leaders with the ability to commit to (possibly mixed) strategies beforehand, whereas the other players are followers who observe the commitment and decide how to play thereafter.

Over the last years, Stackelberg games and their corresponding Stackelberg equilibria have received growing attention in the AI literature, where the computational problem of finding such equilibria in often referred to as the problem of *computing optimal strategies to commit to* (Conitzer and Sandholm, 2006). This surge of interest was motivated by the successful applications of Stackelberg games in many interesting real-world settings. In particular, among the others, the security domain is the most explored one, and, in it, different game models have been introduced, usually referred to as *security games* (Paruchuri et al., 2008; Kiekintveld et al., 2009; An et al., 2011; Tambe, 2011). In such models, there is a defender that has to protect some valuable targets from an attacker, who can wait while observing the defender's protection strategy before deciding where, when and how to attack. This scenario naturally fits into the Stackelberg model, where the defender is the leader and the attacker is the follower. Other interesting applications are found in *toll-setting games*, where the leader is a central authority which collects tolls from the users of a network who, acting as followers, decide on how to best travel through the network so as to minimize their cost after observing the pricing strategy chosen by the authority (Labbé et al., 1998; Labbé and Violin, 2016). Besides the security domain and toll-setting games, applications of Stackelberg games can be found in, among others, interdiction games (Caprara et al., 2016; Matuschke et al., 2017), network routing (Amaldi et al., 2013), inspection games (Avenhaus et al., 1991), and mechanism design (Sandholm, 2002).

Despite the attention that Stackelberg games received from the AI literature, most of the works related to them focus, with some exceptions (see, *e.g.*, (Von Stengel and Zamir, 2010; Conitzer and Korzhyk, 2011; Gan et al., 2018)), on particular game settings that involve only two players (*i.e.*, one leader and one follower) and enjoy specific structures, as it is the case in security games. It is worth pointing out two works that study general Stackelberg games with a single leader and multiple followers; specifically, Von Stengel and Zamir (2010) study the case in which the followers play a Nash equilibrium given the leader's commitment, whereas Conitzer and Korzhyk (2011) address the case where they play a *correlated equilibrium*. We refer the reader to Chapter 3 for a complete survey of the state of tha art on Stackelberg equilibrium computation.

Let us also notice that, while some works (see, *e.g.*, (Letchford and Conitzer, 2010; Bošanský and Cermak, 2015; Cermak et al., 2016)) address the computation of Stackelberg equilibria in games with a sequential (*i.e.*, tree-form) structure, none of them investigates refinements of such equilibria. This is surprising as refinements have been extensively studied for the Nash equilibrium, since it is well-known that classical (unrefined) solution concepts may lead to a sub-optimal behavior off the equilibrium path in games with a sequential structure (see (Van Damme, 1987; Farina et al., 2018a) for some references on the topic).

1.2 Original Contributions

The goal of this thesis is to advance the state of the art on equilibrium computation in general Stackelberg games. In particular, we follow three directions. First, we study the problem of finding Stackelberg equilibria in general Stackelberg games with a single leader and multiple followers. Then, we address Stackelberg games with multiple leaders, proposing a novel way to apply the Stackelberg paradigm in such settings. Finally, we tackle, for the first time, the problem of defining (and computing) equilibrium refinements in Stackelberg games with a sequential structure.

In the rest of this section, we survey all the original contributions that we provide in this thesis. For an easy reference, Table 1.1 shows the contributions related to Stackelberg games with a single leader, summarizing the computational complexity and the algorithmic aspects of the problems we study, with focus on normal-form, extensive-form, Bayesian, and polymatrix games. The table also shows, for comparison, other state-of-the-art results, including those about single-leader single-follower Stackelberg games (our original contributions are those without a reference). Our contributions on Stackelberg congestion games are instead detailed in Table 1.2. The reader can refer to Chapter 3 for additional details on state-of-the-art results reported in the tables.

1.2.1 Stackelberg Games with Multiple Followers

In the first part of the thesis, we address Stackelberg games with a *single leader* and *multiple followers*. Following Von Stengel and Zamir (2010),

		strong Stackelberg(-Nash) equilibrium		weak Stackelberg(-Nash) equilibrium	
Followers' strategies		Pure	Mixed	Pure	Mixed
Normal-form Stackelberg games					
$n = 2$	Complexity	P (Conitzer and Sandholm, 2006)		P (Von Stengel and Zamir, 2010)	
	Algorithm	multi-LP (Conitzer and Sandholm, 2006)		multi-LP (Von Stengel and Zamir, 2010)	
$n = 3$	Complexity	P	NP-hard, $\notin$ Poly-APX (Basilico et al., 2019)	NP-hard	NP-hard, $\notin$ Poly-APX (Basilico et al., 2019)
	Algorithm	multi-LP	spatial branch-and-bound (Basilico et al., 2017b)	multi-lex-MILP	–
$n \geq 4$	Complexity	P	NP-hard, $\notin$ Poly-APX (Basilico et al., 2019)	NP-hard, $\notin$ Poly-APX	NP-hard, $\notin$ Poly-APX (Basilico et al., 2019)
	Algorithm	multi-LP	spatial branch-and-bound (Basilico et al., 2017b)	multi-lex-MILP	–
Bayesian Stackelberg games					
$n = 2$	Complexity	NP-hard (Conitzer and Sandholm, 2006), Poly-APX-complete (Letchford et al., 2009)		NP-hard, Poly-APX-complete	
	Algorithm	MILP (Paruchuri et al., 2008)		multi-LP	
Extensive-form Stackelberg games					
$n = 2$	Complexity	NP-hard (Letchford and Conitzer, 2010)		NP-hard	
	Algorithm	MILP (Bošanský and Cermak, 2015; Cermak et al., 2016)		multi-LP (Von Stengel and Zamir, 2010)	
Stackelberg polymatrix games					
$n = 3$	Complexity	P	NP-hard, $\notin$ Poly-APX (Basilico et al., 2019)	NP-hard	NP-hard, $\notin$ Poly-APX (Basilico et al., 2019)
	Algorithm	multi-LP	spatial branch-and-bound (Basilico et al., 2017b, 2019)	multi-lex-MILP	–
$n \geq 4$ (fixed)	Complexity	P	NP-hard, $\notin$ Poly-APX (Basilico et al., 2019)	NP-hard, $\notin$ Poly-APX	NP-hard, $\notin$ Poly-APX (Basilico et al., 2019)
	Algorithm	multi-LP	spatial branch-and-bound (Basilico et al., 2017b, 2019)	multi-lex-MILP	–
$n \geq 4$ (free)	Complexity	NP-hard, $\notin$ Poly-APX	NP-hard, $\notin$ Poly-APX (Basilico et al., 2019)	NP-hard, $\notin$ Poly-APX	NP-hard, $\notin$ Poly-APX (Basilico et al., 2019)
	Algorithm	multi-LP	spatial branch-and-bound (Basilico et al., 2017b)	multi-lex-MILP	–

Table 1.1: *Summary of the results on the computation of Stackelberg equilibria in normal-form Stackelberg games, Bayesian Stackelberg games, extensive-form Stackelberg games, and Stackelberg polymatrix games. The state-of-the-art results are those with related references.*

we study settings in which, after observing the leader's commitment, the followers play a Nash equilibrium in the resulting game. We refer to this solution as *Stackelberg-Nash equilibrium*. We focus on the case in which the followers are restricted to pure (*i.e.*, non-mixed) strategies, as the general problem with followers playing mixed strategies is already known to be computationally intractable (Basilico et al., 2017a). As we will see, this restriction leads to interesting computational complexity results. Moreover, this is without loss of generality in games always admitting pure-strategy Nash equilibria, as it is the case for congestion games (Rosenthal, 1973).

We study the problem of computing Stackelberg-Nash equilibria, focusing on two cases: the one in which the followers break ties in favor of the leader (what is usually referred to as a *strong* equilibrium), and the case where they break ties against the leader (leading to a *weak* equilibrium).

We analyze three different classes of games, namely, *normal-form* games, *polymatrix* games, and *congestion* games.

Norma-Form Stackelberg Games

After briefly pointing out that a strong Stackelberg-Nash equilibrium (with followers restricted to pure strategies) can be computed efficiently (in polynomial time) by solving multiple *linear programs* (LPs), we entirely devote the remainder of our analysis to the weak case (with, again, followers restricted to pure strategies). In terms of computational complexity, we show that, differently from the strong case, in the weak one the equilibrium-finding problem is NP-hard with two or more followers, while, when the number of followers is three or more, the problem cannot be approximated in polynomial time to within any polynomial multiplicative factor unless P = NP (*i.e.*, in formal terms, it is *not* in the class Poly-APX unless P = NP). To establish these two results, we introduce two reductions, one from Independent Set and the other one from 3-SAT.

After analyzing the complexity of the problem, we focus on its algorithmic aspects. First, we formulate the problem as a *bilevel programming problem*. We then show how to recast it as a single-level *quadratically constrained quadratic program* (QCQP), which we show to be impractical to solve due to admitting a supremum, but not a maximum. We then introduce a restriction based on a *mixed-integer linear program* (MILP) which, while forsaking optimality, always admits an optimal (restricted) solution. Next, we propose an exact algorithm to compute the value of the supremum of the problem based on an enumeration scheme which, at each iteration, solves a *lexicographic* MILP (lex-MILP) where the two objective functions are optimized in sequence. Subsequently, we embed the enumerative algorithm within a branch-and-bound scheme, obtaining an algorithm which is, in practice, much faster. We also extend the algorithms so that, for cases where the supremum is not a maximum, they return a strategy by which the leader can obtain a utility within an additive loss α with respect to the supremum, for any $\alpha > 0$. To conclude, we experimentally evaluate the scalability of our methods over a testbed of randomly generated instances.

Stackelberg Polymatrix Games

We identify two classes of Stackelberg polymatrix games that allow to characterize the complexity of computing Stackelberg-Nash equilibria (with followers restricted to pure strategies). The key property of these games is that, once fixed the number of players, computing a strong or weak equilibrium presents the same complexity, namely polynomial (again assuming that the followers play pure strategies). These games are of practical interest in security problems. Moreover, they are equivalent to *Bayesian* Stackelberg games with one leader and one follower, where the latter may be of different types. Our first class is equivalent to games with *interdependent types*, while the second one is equivalent to games with *independent types* (*i.e.*, the leader's utility is independent of the follower's type). Thus, every result that holds for a game class also holds for its equivalent class.

We investigate whether the problem keeps being easy when the number of players is *not* fixed. We show that it is NP-hard to compute a weak Stackelberg-Nash equilibrium, and we provide an exact (exponential-time) algorithm (conversely, to compute a strong equilibrium, one can adapt the algorithm provided in (Conitzer and Sandholm, 2006) for Bayesian games, by means of our mapping). We also prove that, in all the instances where the weak Stackelberg-Nash equilibrium is a supremum but not a maximum, an α-approximation of the supremum can be found in polynomial time (also in the number of players) for any given additive loss $\alpha > 0$. As for approximation complexity, we show that the problem is Poly-APX-complete. This also shows that, in Bayesian Stackelberg games with uncertainty over the follower, computing a weak Stackelberg-Nash equilibrium is as hard as finding a strong one (Letchford et al., 2009).

Next, we investigate whether, in general polymatrix games with followers restricted to play pure strategies, the problem admits polynomial-time approximation algorithms. We provide a negative answer, showing that in the strong case the problem is *not* in Poly-APX if the number of players is non-fixed, unless $\mathsf{P} = \mathsf{NP}$. We also prove that the same inapproximability result holds for the weak case, even with a fixed number of players.

strong Stackelberg-Nash equilibrium				
		Leader's commitment	Pure	Mixed
Identical action spaces (symmetric games)	Monotonic costs	Complexity	P	P
		Algorithm	Greedy	Greedy
	Generic costs	Complexity	P	NP-hard, $\notin$ Poly-APX
		Algorithm	Dynamic Programming	MILP
Different action spaces	Monotonic costs	Complexity	NP-hard, $\notin$ Poly-APX	NP-hard, $\notin$ Poly-APX
		Algorithm	MILP	MILP
	Generic costs	Complexity	NP-hard, $\notin$ Poly-APX	NP-hard, $\notin$ Poly-APX
		Algorithm	MILP	MILP
weak Stackelberg-Nash equilibrium				
		Leader's commitment	Pure	Mixed
Identical action spaces (symmetric games)	Monotonic costs	Complexity	P	P
		Algorithm	Greedy	Greedy
	Generic costs	Complexity	P	NP-hard, $\notin$ Poly-APX
		Algorithm	Dynamic Programming	multi-lex-MILP
Different action spaces	Monotonic costs	Complexity	NP-hard, $\notin$ Poly-APX	NP-hard, $\notin$ Poly-APX
		Algorithm	multi-lex-MILP	multi-lex-MILP
	Generic costs	Complexity	NP-hard, $\notin$ Poly-APX	NP-hard, $\notin$ Poly-APX
		Algorithm	multi-lex-MILP	multi-lex-MILP

Table 1.2: *Summary of the results on the computation of Stackelberg equilibria in Stackelberg singleton congestion games with a single leader.*

Stackelberg Congestion Games

We provide a comprehensive study of the computational complexity of finding Stackelberg-Nash equilibria in congestion games. These are games with a large number of players that compete for the use of some shared resources, where the cost of each resource is a function of the number of players using that resource, *i.e.*, its *congestion*. Notice that, in such setting, assuming that the followers play a pure-strategy Nash equilibrium is without loss of generality, as congestion games always admit one (Rosenthal, 1973).

First, we focus on games with *singleton* actions, *i.e.*, where each player selects only one resource at a time. We draw a complete picture of the computational complexity of the problem of finding equilibria in Stackelberg singleton congestion games, with pure or mixed-strategy commitments, and considering the cases of finding either a strong equilibrium or a weak one. Interestingly, we identify two features which allow for thoroughly characterizing hard and easy game instances. The first one concerns the relationship among the action spaces of the players, with two possibilities: the one where the players are *symmetric* as they have identical action spaces and therefore they share the same set of resources, and the one where their action spaces may differ. The second feature is related to the shape of the players' cost functions. Two cases are possible: the one where these functions are *monotonically increasing* in the resource congestion and the one in which they may be not.

In particular, we show that, in games where the players' action spaces can be different, computing a (strong or weak) Stackelberg-Nash equilibrium is *not* in Poly-APX unless $\mathsf{P} = \mathsf{NP}$ even when the players' cost functions are monotonic, the leader has only one action available, and her costs are equal to the followers'. This result also holds if we restrict the leader to pure-strategy commitments, given that the leader has only one action available. For symmetric games where the players have identical action spaces, we show that the complexity of computing an equilibrium depends on the nature of the players' cost functions. For the case where the players' costs are generic (monotonic or not) functions of the resource congestion, we prove that the problem is *not* in Poly-APX unless $\mathsf{P} = \mathsf{NP}$. On the other hand, we show that, in symmetric games, the problem of computing a strong or weak Stackelberg-Nash equilibrium can be solved in polynomial time when the cost functions are monotonic by proposing an algorithm for it. We also consider the case where the leader is restricted to pure-strategy commitments, providing a polynomial-time algorithm for its solution which applies even to symmetric games with generic cost functions. This algorithm is based on a polynomial-time dynamic programming algorithm available in the literature for computing a socially optimal Nash equilibria in non-Stackelberg singleton congestion games with identical action spaces, which we improve and extend to solve our problem.

Then, we switch the attention to games beyond singleton ones. We show that having actions made of only one resource is necessary to have efficient (polynomial-time) algorithms. Indeed, we prove that finding a strong Stackelberg-Nash equilibrium is NP-hard and *not* in Poly-APX unless $\mathsf{P} = \mathsf{NP}$, even if players' actions contain only two resources, costs are monotonic, and players are symmetric. We also introduce and study singleton congestion games in which the players are partitioned into *classes*, with followers of the same class sharing the same set of actions. These are a generalization of singleton games with symmetric players, capturing the common case in which users can be split into (usually few) different classes, such as, *e.g.*, users with different priorities. For these games, we provide a dynamic programming algorithm that computes a strong Stackelberg-Nash equilibrium in polynomial time, when the number of classes is fixed and the leader is restricted to play pure strategies. On the other hand, we prove that, if the leader is allowed to play mixed strategies, then the problem becomes NP-hard even with only four classes and monotonic costs.

Finally, for all the settings we study, we design MILP formulations for computing a strong Stackelberg-Nash equilibrium, and we experimentally evaluate them on a testbed containing both randomly generated game instances and worst-case instances based on our hardness reductions.

1.2.2 Stackelberg Games with Multiple Leaders

In the second part of the thesis, we focus our attention on games with multiple leaders, providing a new way to apply the Stackelberg paradigm to any finite (underlying) game. Our approach extends the idea of commitment to *correlated strategies* in settings involving multiple leaders and followers, generalizing the work of Conitzer and Korzhyk (2011). The crucial component of our framework is that a leader can decide whether to participate in the commitment or to defect from it by becoming a follower. This induces a preliminary *agreement stage* that takes place before the underlying game is played, where the leaders decide, in turn, whether to opt out from the commitment or not. We model this stage as a sequential game, whose size is factorial in the number of players. Our goal is to identify commitments guaranteeing some desirable properties that we define on the agreement stage. The first one requires that the leaders do not have any incentive to become followers. It comes in two flavors, called *stability* and *perfect stability*, which are related to, respectively, Nash and subgame perfect equilibria of the sequential game representing the agreement stage. The second property is also defined in two flavors, namely *efficiency* and *perfect efficiency*, both enforcing Pareto optimality with respect to the leaders' utility functions, though at different levels of the agreement stage.

We introduce three solution concepts, which we generally call *Stackelberg correlated equilibria*. They differ depending on the properties they call for. Specifically, (simple) Stackelberg correlated equilibria, Stackelberg correlated equilibria *with perfect agreement*, and Stackelberg correlated equilibria *with perfect agreement and perfect efficiency* require, respectively, stability and efficiency, perfect stability and efficiency, and both perfect stability and perfect efficiency.

First, we investigate the game theoretic properties of our solution concepts. We show that Stackelberg correlated equilibria with or without perfect agreement are guaranteed to exist in any game, while Stackelberg correlated equilibria with perfect agreement and perfect efficiency may not. Moreover, we compare the former with other solution concepts, both Stackelberg and non-Stackelberg ones.

Then, we switch the attention to the computational complexity perspective. We show that, provided a suitably defined *stability oracle* is solvable in polynomial time, a Stackelberg correlated equilibrium optimizing some linear function of leaders' utilities (such as the leaders' social welfare) can be computed in polynomial time, even in the number of players. The same holds for finding *a* Stackelberg correlated equilibrium with perfect agreement, while we prove that computing an optimal one is an intractable problem. Nevertheless, in the latter case, we provide an (exponential in the game size) upper bound on the necessary number of queries to the oracle.

In conclusion, we study which classes of games admit a polynomial-time stability oracle, focusing on *succinct games of polynomial type* (Papadimitriou and Roughgarden, 2008). We show that the problem solved by our oracle is strictly connected with the *weighted deviation-adjusted social welfare problem* introduced by Jiang and Leyton-Brown (2011). As a result, we get that our oracle is solvable in polynomial time in all the game classes where the same holds for the problem of finding an optimal correlated equilibrium.

1.2.3 Trembling-Hand Perfection in Stackelberg Games

In the last part of the thesis, we study Stackelberg games with a sequential structure, usually referred to as *extensive-form* Stackelberg games. In particular, we show that classical Stackelberg equilibria may prescribe the players to play sub-optimally off the equilibrium path, as it is the case for the Nash equilibrium. Thus, in order to amend these weaknesses, we propose a way to refine Stackelberg equilibria thorough trembling-hand perfection, which is based on the idea that each player might play each action with low-but-non-zero probabilities, usually called *trembles* (Selten, 1975).

We show that for every perturbation scheme (*i.e.*, any possible way of introducing trembles), the set of limit points of Stackelberg equilibria for perturbed games with vanishing perturbations is always a nonempty subset of the Stackelberg equilibria of the non-perturbed game. This does not hold when focusing only on strong (or weak) equilibria: for a given game, the set of strong Stackelberg equilibria (or weak Stackelberg equilibria) in the non-perturbed game may be disjoint from the set of limit points of strong Stackelberg equilibria (or weak Stackelberg equilibria) in the perturbed game. We resort to the perturbation schemes used for *quasi-perfect* equilibria (Van Damme, 1984) and *extensive-form* perfect equilibria (Selten, 1975) to define their Stackelberg counterpart—and their strong and weak versions—as refinements of the Stackelberg equilibrium.

Next, we focus on quasi perfection. We formally define the *quasi-perfect Stackelberg equilibrium* refinement game theoretically in the same axiomatic fashion as the quasi-perfect equilibrium was defined for non-Stackelberg games (Van Damme, 1984). Thus, our definition is based on a set of properties of the players' strategies, and it cannot be directly used to search for a quasi-perfect Stackelberg equilibrium. Subsequently, we define a class of perturbation schemes for the sequence form such that any limit point of a sequence of Stackelberg equilibria in perturbed games with vanishing perturbation is a quasi-perfect Stackelberg equilibrium. This class of perturbation schemes strictly includes those used to find a quasi-perfect equilibrium by Miltersen and Sørensen (2010). Then, we extend the algorithm by Cermak et al. (2016) to the case of quasi-perfect Stackelberg equilibrium computation. We derive the corresponding mathematical program for computing a *Stackelberg extensive-form correlated equilibrium* when a perturbation scheme is introduced and we discuss how the individual steps of the algorithm change. In particular, the implementation of our algorithm is much more involved, requiring the combination of branch-and-bound techniques with arbitrary-precision arithmetic to deal with small perturbations. This does not allow a direct application of off-the-shelf solvers. Finally, we experimentally evaluate the scalability of our algorithm.

In conclusion, we also study the computational complexity of finding Stackelberg equilibrium refinements, showing that the problem of deciding the existence of a Stackelberg equilibrium—refined or not—that gives the leader expected value at least ν is NP-hard.

1.3 Structure of the Work

In this section, we describe the structure of the thesis. Before presenting our results, we introduce the main concepts related to algorithmic game theory, with particular emphasis on Stackelberg games. Specifically:

- Chapter 2 introduces, in the first part, the formal definition of game, describing the different game representations which are studied in the thesis. Then, the second part of the chapter defines two classical (non-Stackelberg) equilibrium concepts that are relevant for the rest of the work, namely, Nash and correlated equilibria.

- Chapter 3 surveys the main state-of-the-art results on Stackelberg games and the computation of Stackelberg equilibria. These results represent the groundings upon which we build our original contributions.

Part I: Stackelberg Games with Multiple Followers

Our contributions are organized as follows:

- Chapter 4 provides our computational results on Stackelberg-Nash equilibria (with follower restricted to pure strategies) in normal-form Stackelberg games. A preliminary version of the results provided in this chapter appeared in (Coniglio et al., 2017), while a complete and extended version is in (Coniglio et al., 2019).

- Chapter 5 addresses the problem of computing Stackelberg-Nash equilibria in Stackelberg polymatrix games, also pointing out which results can be directly extended to the Bayesian setting. The results in this chapter appeared in (De Nittis et al., 2018a) (see (De Nittis et al., 2018b) for an extended version).

- Chapter 6 focuses on the problem of finding Stackelberg-Nash equilibria in Stackelberg congestion games. The results related to singleton games appeared in (Marchesi et al., 2018a) and its extended version (Castiglioni et al., 2019c). Instead, all the other results are provided by (Marchesi et al., 2019a) (see (Marchesi et al., 2019b) for an extended version of the latter).

- Chapter 7 reports the experimental evaluation of the algorithms developed for Stackelberg games with multiple followers. All these results are taken from the papers related to the previous chapters, namely (Coniglio et al., 2019; De Nittis et al., 2018a; Castiglioni et al., 2019c; Marchesi et al., 2019a).

Part II: Stackelberg Games with Multiple Leaders

The results provided in this part of the thesis appeared in (Castiglioni et al., 2019a) (see Castiglioni et al. (2019b) for an extended version). Our contributions are organized as follows:

- Chapter 8 introduces our model for multi-leader Stackelberg games, studying the game theoretic properties of the related solution concepts, in terms of existence and relation with other equilibrium notions.

- Chapter 9 analyses the computational complexity of the problem of computing equilibria in our multi-leader Stackelberg games.

Part III: Trembling-Hand Perfection in Stackelberg Games

Our contributions are organized as follows:

- Chapter 10 studies trembling-hand perfection in extensive-form Stackelberg games, so as to refine the classical notion of Stackelberg equilibrium. The results in this chapter appeared in (Farina et al., 2018b).
- Chapter 11 focuses on a particular type of refinement, known as quasi-perfection. The results in this chapter appeared in (Marchesi et al., 2019c) (see (Marchesi et al., 2018b) for an extended version).

Finally, Chapter 12 concludes the thesis by drawing some overall observations and pointing out possible directions for future research.

CHAPTER 2

Games and Equilibria

In this chapter, we provide a brief introduction to the theory of games and their equilibria, surveying the basic concepts needed in the rest of this work, with particular emphasis on computational results.

Section 2.1 starts introducing the general picture of finite games and their most common representations, namely, the *normal form* and the *extensive form*. In Section 2.2, we also introduce some *succinct* game representations that allow to compactly encode games with a specific structure, focusing on *polymatrix* and *congestion* games. Finally, Section 2.3 shades the light on what it means to solve a game by defining classical *equilibrium concepts*, such as *Nash* and *correlated* equilibria.

For a comprehensive treatment of the subject, we refer the reader to the books by Shoham and Leyton-Brown (2008) and Nisan et al. (2007).

2.1 Games and How to Represent Them

Games are powerful mathematical tools that provide rigorous models of complex strategic interactions involving multiple *rational agents* (or *players*). Such interactions arise in many different real-world settings, such as, *e.g.*, cybersecurity problems (Tambe, 2011), auctions for web advertising (Gatti et al., 2015), and, more naturally, common recreational games like Chess (Campbell et al., 2002), Go (Silver et al., 2016), and Poker (Brown and Sandholm, 2018, 2019). The three fundamental ingredients defining a game are: the participating *players*, the *strategies* they are allowed to play, and their preferences over the possible game outcomes, which are usually quantified by means of their *utilities*. Formally:

Definition 2.1 (Finite Game)**.** *A* (finite) game Γ *is a tuple* (N, S, u)*, where:*

- $N := \{1, \dots, n\}$ *is a finite set of players;*
- $S := \times_{p \in N} S_p$ *is a set of* (pure) strategy profiles *(or* outcomes*), with* S_p *denoting a finite set of player* p*'s* (pure) strategies*;*
- $u := \{u_p\}_{p \in N}$ *is a set of players'* utility functions*, with* $u_p : S \to \mathbb{R}$ *defining player* p*'s utility over strategy profiles* $s = (s_1, \dots, s_n) \in S$*.*

In general, players are also allowed to randomize their play according to some probability distribution. Formally, for every player $p \in N$, we let x_p be a player p's *mixed strategy*, *i.e.*, a probability distribution defined over her set of strategies S_p, with $x_p(s_p)$ denoting the probability of playing $s_p \in S_p$. Moreover, $\mathcal{X}_p := \Delta(S_p)$ denotes the set of player p's mixed strategies, while $x = (x_1, \dots, x_n) \in \times_{p \in N} \mathcal{X}_p$ is a *mixed strategy profile* specifying a mixed strategy $x_p \in \mathcal{X}_p$ for each player $p \in N$. With an overload of notation, we use $u_p(x)$ to denote the player p's *expected utility* when the mixed strategy profile x is played, *i.e.*, $u_p(x) := \sum_{s \in S} u_p(s) \prod_{q \in N} x_q(s_q)$.

For the ease of presentation, we also introduce the following notation. Given a strategy profile $s \in S$, we let $s_{-p} \in S_{-p} := \times_{q \in N \setminus \{p\}} S_q$ be the partial profile obtained by dropping player p's strategy s_p from s, so that $s = (s_p, s_{-p})$. Similarly, given a mixed strategy profile $x \in \times_{p \in N} \mathcal{X}_p$, we let $x_{-p} \in \times_{q \in N \setminus \{p\}} \mathcal{X}_q$ be such that $x = (x_p, x_{-p})$.

While a game may admit different equivalent representations, the most natural one for that game depends on its specific structure. In the rest of this section, we introduce the two most common game representations: the *normal form* and the *extensive form*.

2.1.1 Normal-Form Representation

The normal form is a tabular representation in which each player's utility function is specified by a matrix, where each entry defines the player's utility for some combination of players' actions. Formally:

Definition 2.2 (Normal-Form Game)**.** *A* normal-form game Γ *is a tuple* (N, A, U)*, where:*

- $N \coloneqq \{1, \dots, n\}$ *is a finite set of players;*
- $A \coloneqq \bigtimes_{p \in N} A_p$ *is a set of* action profiles*, with A_p denoting a finite set of player p's actions, of cardinality $m_p \coloneqq |A_p|$;*
- $U \coloneqq \{U_p\}_{p \in N}$ *is a set of matrices, with $U_p \in \mathbb{Q}^{m_1 \times \dots \times m_n}$ representing a player p's (multidimensional)* utility *(or* payoff*)* matrix*, in which each component $U_p^{a_1 \dots a_n}$ corresponds to the utility of player p when all the players play the action profile $a = (a_1, \dots, a_n) \in A$.*

Any finite game can be represented in normal form by letting $A = S$ and $U_p^{a_1 \dots a_n} = u_p(a)$ for every $p \in N$ and $a = (a_1, \dots, a_n) \in A$.

For the ease of presentation and when no ambiguity arises, we will often write U_p^a in place of $U_p^{a_1 \dots a_n}$. Moreover, given an action profile $a \in A$, we define $a_{-p} \in A_{-p} \coloneqq \bigtimes_{q \in N \setminus \{p\}} A_q$ so that $a = (a_p, a_{-p})$, with $U_p^{a_{-p}, a_p}$ denoting the component of U_p corresponding to the action profile a.

Using matrix notation, we represent a player p's mixed strategy (or *strategy*, for short) using a vector $x_p \in [0, 1]^{m_p}$ such that $\sum_{a_p \in A_p} x_p^{a_p} = 1$, where each component $x_p^{a_p}$ of x_p is the probability by which player p plays action $a_p \in A_p$. Moreover, $\Delta_p \coloneqq \{x_p \in [0, 1]^{m_p} : \sum_{a_p \in A_p} x_p^{a_p} = 1\}$ denotes the set of player p's strategies, corresponding to the standard $(m_p - 1)$–simplex in $\mathbb{R}^{m_p}$. As customary when working with normal-form games, a strategy is said *pure* when only one action is played with positive probability, *i.e.*, when $x_p \in \{0, 1\}^{m_p}$, and *mixed* otherwise. Mixed strategy profiles (or *strategy profiles*, for short) are defined as in general finite games. Given a strategy profile $x = (x_1, \dots, x_n) \in \bigtimes_{p \in N} \Delta_p$, the expected utility of player $p \in N$ is the n-th-degree polynomial $\sum_{a \in A} U_p^a x_1^{a_1} x_2^{a_2} \dots x_n^{a_n}$.

Bayesian Normal-Form Games

In some scenarios, the players are uncertain about the preferences of their opponents (*i.e.*, their utilities). We can model these situations using *Bayesian* games. In normal form, we have the following formal definition.

Definition 2.3 (Bayesian Normal-Form Game)**.** *A* Bayesian (normal-form) game Γ *is a tuple $(N, \Theta, \Omega, A, U)$, where:*

- $N \coloneqq \{1, \dots, n\}$ *is a finite set of players;*
- $\Theta \coloneqq \bigtimes_{p \in N} \Theta_p$ *is a set of* type profiles*, with Θ_p denoting a finite set of player p's types;*
- $\Omega \in \Delta(\Theta)$ *is a probability distribution over the set of type profiles;*

- $A \coloneqq \bigtimes_{p \in N} \bigtimes_{\theta_p \in \Theta_p} A_{p,\theta_p}$ *is a set of* action profiles*, with* A_{p,θ_p} *denoting a finite set of actions for player* p*'s type* θ_p*;*

- $U \coloneqq \{U_{p,\theta}\}_{p \in N, \theta \in \Theta}$ *is a set of matrices, with* $U_{p,\theta} \in \mathbb{Q}^{|A_{1,\theta_1}| \times \ldots \times |A_{n,\theta_n}|}$ *representing a player* p*'s* utility matrix *when the players' type profile is* $\theta = (\theta_1, \ldots, \theta_n)$*, in which each component* $U_{p,\theta}^{a_1 \ldots a_n}$ *corresponds to the utility of player* p *when the players' type profile is* θ *and all the players play the action profile* $a = (a_1, \ldots, a_n) \in \bigtimes_{p \in N} A_{p,\theta_p}$.

Intuitively, before the game starts, a type profile $\theta = (\theta_1, \ldots, \theta_n) \in \Theta$ is drawn according to the probability distribution Ω, and, then, each player is informed about her type θ_p as specified by θ, while she keeps to be uncertain about the other players' types. As a result, the players do not have complete knowledge of θ, and, in turn, of their utilities.

Let us remark that, in a Bayesian game, a player's pure strategy specifies an action for each type of that player, while, as usual, mixed strategies are defined as probability distributions over pure strategies. Thus, using the formalism of general finite games, we have $S = A$ and, for every $p \in N$ and $a = (a_1, \ldots, a_n) \in A$, $u_p(a) = \sum_{\theta \in \Theta} \Omega(\theta) U_{p,\theta}^{a_1 \ldots a_n}$. This also shows how any Bayesian game can be equivalently represented as a non-Bayesian normal-form game, in which each player has a number of actions exponentially larger (in the number of types) than in the original game.

2.1.2 Extensive-Form Representation

The extensive form allows to represent games where the players play *sequentially*, and it is usually specified by defining a *game tree*. Formally:

Definition 2.4 (Extensive-Form Game)**.** *An* extensive-form game with imperfect information Γ *is a tuple* $(N, H, Z, A, \rho, \chi, \pi_c, u, \mathcal{I})$ *in which:*

- $N \coloneqq \{1, \ldots, n\}$ *is a finite set of players;*

- $H \coloneqq \bigcup_{p \in N} H_p \cup H_c$ *is a finite set of nonterminal nodes, where* H_c *is a set of chance nodes, while* H_p *is a set of player* p*'s decision nodes;*

- Z *is the set of terminal nodes;*

- $A \coloneqq \bigcup_{p \in N} A_p \cup A_c$ *is a finite set of actions, where* A_c *contains chance moves, while* A_p *is a set of player* p*'s actions;*

- $\rho : H \to 2^A$ *is an action function that assigns to each nonterminal node a set of available actions;*

- $\chi : H \times A \to H \cup Z$ *is a successor function that defines the node reached when an action is performed in a nonterminal node;*
- $\pi_c : H \cup Z \to [0, 1]$ *assigns each node with its probability of being reached given chance moves;*
- $u := \{u_p\}_{p \in N}$ *is a set of players'* payoff functions, *where* $u_p : Z \to \mathbb{R}$ *specifies player* p*'s payoffs (or utilities) in each terminal node;*
- $\mathcal{I} := \{\mathcal{I}_p\}_{p \in N}$ *is an* information partition, *where* $\mathcal{I}_p$ *defines a partition of* H_p *into* information sets, *which are groups of nodes that are indistinguishable by player* p*.*

Let us remark that, for every player $p \in N$ and information set $I \in \mathcal{I}_p$, it must be the case that $\rho(h) = \rho(h') := A(I)$ for any pair of nodes $h, h' \in I$, otherwise player p would be able to distinguish them. As usual, w.l.o.g., we assume that each action $a \in A$ belongs to only one set $A(I)$.

We focus on extensive-form games with *perfect recall*, *i.e.*, games in which no player forgets what she did or knew in the past. Formally, for every $p \in N$ and information set $I \in \mathcal{I}_p$, we require that all nodes belonging to I share the same player p's actions on their paths from the root node.

Note that any extensive-form game can also be represented using the formalism of general finite games. In particular, a player p's strategy $s_p \in S_p$ defines a collection of actions, one per information set $I \in \mathcal{I}_p$, thus specifying player p's behavior at every decision node of the game tree. As a result, any extensive-form game admits an equivalent normal-form representation, which, however, may be exponentially larger than the original game tree.

Strategies in Extensive-Form Games

A straightforward way of defining mixed strategies in an extensive-form game is to identify them as the strategies of the corresponding, exponentially sized normal form. Fortunately, under the perfect recall assumption, it is sufficient to restrict the attention to *behavioral strategies* (Kuhn, 1953), which define, for every player $p \in N$ and information set $I \in \mathcal{I}_p$, a probability distribution over the actions $A(I)$. For $p \in N$, let $\pi_p \in \Pi_p$ be a player p's behavioral strategy, with π_{pa} denoting the probability of playing action $a \in A_p$. As for the other types of strategies, we let $\pi \in \bigtimes_{p \in N} \Pi_p$ be a *behavioral strategy profile*. Moreover, overloading notation, we use u_p as if it were defined over behavioral strategies instead of terminal nodes, with $u_p(\pi)$ being player p's expected utility when $\pi \in \bigtimes_{p \in N} \Pi_p$ is played.

While behavioral strategies require space polynomial in the size of the game tree, players' expected utilities $u_p(\pi)$ depend non-linearly on the single strategy components π_{pa}, and, thus, they are *not* amenable to efficient computational techniques. However, assuming perfect recall, behavioral strategies admit an equivalent, computationally efficient representation, which is based on what is called the *sequence form* of an extensive-form game (Von Stengel, 1996; Romanovskii, 1962).

In the sequence form, every node $h \in H \cup Z$ defines a *sequence* $\sigma_p(h)$ for player $p \in N$, which is the ordered set of player p's actions on the path from the root of the game tree to h. Let Σ_p be the set of player p's sequences. As usual, we let $\sigma_\varnothing \in \Sigma_p$ be a fictitious element representing the empty sequence. In perfect-recall games, given an information set $I \in \mathcal{I}_p$, for any pair of nodes $h, h' \in I$ it holds $\sigma_p(h) = \sigma_p(h') := \sigma_p(I)$. Given $\sigma_p \in \Sigma_p$ and $a \in A(I)$ with $I \in \mathcal{I}_p : \sigma_p = \sigma_p(I)$, we denote as $\sigma_p a$ the *extended* sequence obtained by appending a to σ_p. Moreover, for any pair of sequences $\sigma_p, \sigma'_p \in \Sigma_p$, we write $\sigma'_p \sqsubseteq \sigma_p$ whenever σ'_p is a *prefix* of σ_p, *i.e.*, σ_p can be obtained by extending σ'_p with a finite number of actions. Given $\sigma_p \in \Sigma_p$, we also denote with $I_p(\sigma_p)$ the information set $I \in \mathcal{I}_p$ such that $\sigma_p = \sigma_p(I)a$ for some action $a \in A(I)$.

A sequence-form strategy is called *realization plan*, and it assigns each sequence with its probability of being played. For $p \in N$, we let $r_p \in R_p$ be a player p's realization plan. In order to be well-defined, a realization plan r_p must be such that $r_p(\sigma_\varnothing) = 1$ and, for every $I \in \mathcal{I}_p$:

$$r_p(\sigma_p(I)) = \sum_{a \in A(I)} r_p(\sigma_p(I)a).$$

A *realization plan profile* $r \in \bigtimes_{p \in N} R_p$ is defined as usual.

Finally, letting $\Sigma := \bigtimes_{p \in N} \Sigma_p$ be the set of *sequence profiles* $\sigma = (\sigma_1, \ldots, \sigma_n)$, overloading notation, $u_p : \Sigma \mapsto \mathbb{R}$ is the player p's utility function in the sequence form, defined as follows:

$$u_p(\sigma) := \sum_{h \in Z \,:\, \sigma(h) = \sigma} u_i(h) \pi_c(h),$$

where, for $h \in H \cup Z$, we let $\sigma(h)$ be the sequence profile defined by the sequences $\sigma_p(h)$. Moreover, we also use u_p as if it were defined over realization plans. Formally, $u_p(r) := \sum_{\sigma \in \Sigma} u_p(\sigma) \prod_{q \in N} r_q(\sigma_q)$, where r_q is the player q's realization plan in the profile r.

The sequence form is usually expressed with matrix notation, as follows. Player p's utility function is a $|\Sigma_1| \times \ldots \times |\Sigma_n|$ matrix U_p whose entries are the utilities $u_p(\sigma)$, for $\sigma \in \Sigma$. The constraints defining $r_p \in R_p$ are expressed as $F_p r_p = f_p$, where: F_p is a $(|\mathcal{I}_p| + 1) \times |\Sigma_p|$ (multidimensional) matrix, $f_p \in \mathbb{R}^{|\mathcal{I}_p|+1}$, and, overloading notation, $r_p \in \mathbb{R}^{|\Sigma_p|}$ is a vector representing r_p. Specifically, introducing a fictitious information set $I_\varnothing$, the entry of F_p indexed by $(I_\varnothing, \sigma_\varnothing)$ is 1, and, for $I \in \mathcal{I}_p$ and $\sigma_p \in \Sigma_p$, the entry indexed by (I, σ_p) is -1 if $\sigma_p = \sigma_p(I)$, while it is 1 if $\sigma_p = \sigma_p(I)a$ for some $a \in A(I)$. F_p is zero everywhere else. Moreover, $f_p^\top = (1\ 0 \cdots 0)$. Finally, given $r \in \bigtimes_{p \in N} R_p$, we can write $u_p(r) = U_p \prod_{q \in N} r_q$, where the products involving the matrix U_p and the vectors r_q representing players' realization plans are defined in such a way that the result is a scalar.

In perfect-recall games, behavioral strategies and realization plans are equally expressive (Von Stengel, 1996). Given $r_p \in R_p$, we obtain an equivalent $\pi_p \in \Pi_p$ by setting, for all $I \in \mathcal{I}_p$ and $a \in A(I)$, $\pi_{pa} = \frac{r_p(\sigma_p(I)a)}{r_p(\sigma_p(I))}$ when $r_p(\sigma_p(I)) > 0$, while π_{pa} can be any otherwise. Similarly, $\pi_p \in \Pi_p$ has an equivalent $r_p \in R_p$ with $r_p(\sigma_p) = \prod_{a \in \sigma_p} \pi_{pa}$ for all $\sigma_p \in \Sigma_p$. [1]

In conclusion, let us remark that the constraints defining players' realization plans are linear, and, in two-player games, the same holds for expected utilities. As a result, the sequence form allows for the development of efficient computational tools for two-player extensive-form games.

2.2 Succinct Games

In this section, we introduce some game representations that allow to compactly represent finite games with many players and strategies by leveraging the specific structure of such games. In particular, we focus on *succinct games*, as formally defined by Papadimitriou and Roughgarden (2008).

Definition 2.5 (Succinct Game)**.** *A* succinct game of polynomial type *is a finite game $\Gamma = (N, S, u)$ such that:*

- *the number of players n and the cardinalities $|S_p|$ of the players' strategy sets are polynomially bounded in the size of the game;*
- *there exists a polynomial-time algorithm that, given as input a player $p \in N$ and a strategy profile $s \in S$, returns $u_p(s)$.*

There are many classes of games that can be defined as succinct games of polynomial type, such as, *e.g.*, *graphical games* (Kearns et al., 2001, 2013), *polymatrix games* (Janovskaja, 1968; Howson Jr, 1972; Eaves, 1973), *anonymous games* (Blonski, 2000), *congestion games* (Rosenthal, 1973),

[1] Here, the equivalence is in terms of probabilities that the strategies induce on terminal nodes, *i.e.*, what is usually known in the literature as *realization equivalence*.

facility location games (Chun et al., 2004), *network design games* (Anshelevich et al., 2008), *local-effect games* (Leyton-Brown and Tennenholtz, 2003), and *scheduling games* (Fotakis et al., 2009).

Next, we provide further details on polymatrix and congestion games.

2.2.1 Polymatrix Games

Polymatrix games capture settings in which the players interact *pairwise*. Specifically, in a polymatrix game, each player competes in a two-player normal-form game with every opponent, adopting a common strategy in all such games. Then, a player's overall utility is given by the sum of the utilities perceived in the pairwise games. Formally:

Definition 2.6 (Polymatrix Game)**.** *A* polymatrix game Γ *is defined as a tuple* $(N, \mathcal{A}, U)$*, where:*

- $N := \{1, \dots, n\}$ *is a finite set of players;*
- $\mathcal{A} := \{A_p\}_{p \in N}$ *contains players'* actions sets*, with* A_p *denoting a finite set of player* p*'s actions, of cardinality* $m_p := |A_p|$*;*
- $U := \{U_{p,q}\}_{p \neq q \in N}$ *is a set of matrices, with* $U_{p,q} \in \mathbb{Q}^{m_p \times m_q}$ *representing a player* p*'s* utility *(or* payoff*) matrix, in which each component* $U_{p,q}^{a_p a_q}$ *corresponds to the utility of player* p *when playing against player* q*, with their actions being* $a_p \in A_p$ *and* $a_q \in A_q$*, respectively.*

When working with polymatrix games, we define action profiles $a = (a_1, \dots, a_n) \in A := \times_{p \in N} A_p$, mixed strategies $x_p \in \Delta_p$, and strategy profiles $x = (x_1, \dots, x_n) \in \times_{p \in N} \Delta_p$ as in normal-form games, using the same notation and conventions. Furthermore, given a strategy profile $x = (x_1, \dots, x_n)$, the expected utility of player $p \in N$ is given by the sum of the expected utilities resulting from each two-player normal-form game involving p, *i.e.*, the polynomial $\sum_{q \neq p \in N} \sum_{a_p \in A_p} \sum_{a_q \in A_q} U_{p,q}^{a_p a_q} x_p^{a_p} x_q^{a_q}$.

Clearly, we can cast polymatrix games as general finite games by letting $S = A$ and $u_p(a) = \sum_{q \neq p \in N} U_{p,q}^{a_p a_q}$ for every $p \in N$ and $a = (a_1, \dots, a_n) \in A$. Moreover, notice that they are succinct games of polynomial type since the size of a game instance $(N, \mathcal{A}, U)$ is $O(n^2 m^2)$, where n is the number of players and, w.l.o.g., we assumed $m_p = m$ for all $p \in N$.

2.2.2 Congestion Games

In a congestion game, the players compete for the use of a set of shared resources, with each player choosing a subset of such resources. Then, a player incurs a cost (*i.e.*, the opposite of utility) equal to the sum of the costs for the selected resources, where each resource cost depends on the number of players using it (called the resource *congestion*). Formally:

Definition 2.7 (Congestion Game)**.** *A* congestion game Γ *is defined as a tuple* $(N, R, \mathcal{A}, c)$*, where:*

- $N \coloneqq \{1, \dots, n\}$ *is a finite set of players;*
- $R \coloneqq \{1, \dots, r\}$ *is a finite set of resources;*
- $\mathcal{A} \coloneqq \{A_p\}_{p \in N}$ *defines players'* actions sets*, with* $A_p \subseteq 2^R$ *denoting a finite set of player* p*'s actions, in which each action* $a_p \in A_p$ *is a non-empty subset of resources, i.e.,* $a_p \subseteq R$*;*
- $c \coloneqq \{c_i\}_{i \in R}$ *is a finite set of* resource cost functions*, with* $c_i : \mathbb{N} \to \mathbb{Q}$ *defining the cost of resource* i *as a function of its congestion.*

We assume that $c_i(0) = 0$ for every resource $i \in R$. Given an action profile $a = (a_1, \dots, a_n) \in A \coloneqq \times_{p \in N} A_p$, we let $c_p(a) \coloneqq \sum_{i \in a_p} c_i(\nu_i^a)$ be player p's cost when the game is played according to the actions defined by a, with $\nu_a^i \coloneqq |\{q \in N \mid i \in a_q\}|$ denoting the congestion induced by a on $i \in R$, *i.e.*, the number of players choosing resource i.

When working with congestion games, we adopt the same notation and conventions as for normal-form games, defining mixed strategies $x_p \in \Delta_p$ and strategy profiles $x = (x_1, \dots, x_n) \in \times_{p \in N} \Delta_p$. Moreover, with an abuse of notation, given a strategy $x_p \in \Delta_p$, we let $x_p^i \coloneqq \sum_{a_p \ni i} x_p^{a_p}$ be the probability of selecting resource $i \in R$ when x_p is played.

In the literature, many classes of congestion games have been studied, depending on the specific structure of the players' action sets. For instance, one possibility is that the players play a congestion game on a graph, with their actions being either paths from a source to a destination (Fabrikant et al., 2004) or spanning trees (Werneck et al., 2000), or, as studied by Ackermann et al. (2008), the players' action sets may be represented as matroids. In this work, we extensively analyze *singleton congestion games* (Ieong et al., 2005), where the players' actions are required to be singletons, *i.e.*, $|a_p| = 1$ for every $p \in N$ and $a_p \in A_p$. Formally:

Definition 2.8 (Singleton Congestion Game)**.** *A* singleton congestion game *is a congestion game* $\Gamma = (N, R, \mathcal{A}, c)$ *in which* $A_p \subseteq R$ *for all* $p \in N$*.*

When working with singleton games, we use mixed strategies $x_p \in \Delta_p$ as if they were directly defined over resources in A_p, with x_p^i denoting the probability by which player p selects resource $i \in A_p$.

2.3 Classical Equilibrium Concepts

Since the birth of game theory, researchers have put much of their effort trying to come up with a universal definition of "optimal" solution for a game. Indeed, while in optimization it is clear that the best solution for a model is one maximizing (or minimizing) the given objective function, in game theory no such clear definition exists, as games involve multiple players having their own objectives. In game theory, the most suitable definition of solution is the one of *equilibrium*, *i.e.*, a stable situation from which no player wants to leave. Unfortunately, it turns out that no unique definition of equilibrium exists, and, thus, over the last decades many possible equilibrium concepts have been introduced, differing on the specific classes of games they refer to and the assumptions they make.

In this section, we survey the two most widely adopted equilibrium concepts in finite games: the *Nash equilibrium* (Nash, 1951) and the *correlated equilibrium* (Aumann, 1974). Next, in the following Chapter 3, we introduce and extensively describe the *Stackelberg equilibrium* (Von Stackelberg, 1934), which is the main focus on this work.

2.3.1 Nash Equilibrium

The Nash equilibrium has been the most acclaimed and studied equilibrium concept in the literature, since it was originally introduced by Nash (1951). The idea of Nash equilibrium is strikingly simple: the players in a game are at an equilibrium if none of them has an incentive to *unilaterally* deviate from the currently played strategy, given that the others do not deviate either. It turns out that if we consider equilibria in which the players are allowed to play general mixed strategies, then any finite game admits at least one Nash equilibrium. Formally:

Definition 2.9 (Nash Equilibrium). *A mixed strategy profile $x \in \bigtimes_{p \in N} \mathcal{X}_p$ is a* Nash equilibrium (NE) *of a finite game Γ if, for every player $p \in N$ and mixed strategy $x'_p \in \mathcal{X}_p$, the following holds:*

$$u_p(x) \geq u_p(x'_p, x_{-p}). \tag{2.1}$$

Theorem 2.1 (Nash (1951)). *Every finite game admits at least one Nash equilibrium in mixed strategies.*

Notice that it is also possible to define NEs on pure strategy profiles $s \in S$, and these are usually referred to as *pure-strategy* NEs (or, for short, *pure* NEs). Formally:

Definition 2.10 (Pure Nash Equilibrium)**.** *A pure strategy profile $s \in S$ is a* pure NE *of a finite game Γ if, for every player $p \in N$ and pure strategy $s'_p \in S_p$, the following holds:*

$$u_p(s) \geq u_p(s'_p, s_{-p}). \tag{2.2}$$

Let us observe that *not* all finite games admit a pure NE. Nevertheless, there are some specific classes of games that always admit at least one pure NE, such as, *e.g.*, congestion games (Rosenthal, 1973).

Due to the proliferation of NEs, most of the early works in algorithmic game theory has focused on the problem of computing them. Indeed, one of the most important results on the characterization of the computational complexity of finding equilibria is that of Daskalakis et al. (2009), who prove that computing an NE is a PPAD-complete problem. Moreover, other works further investigate the complexity of finding NEs, studying, *e.g.*, approximate NEs (Braverman et al., 2014; Deligkas et al., 2016), the problem of computing social-welfare-maximizing NEs (Conitzer and Sandholm, 2008), and the complexity of finding equilibria in games with a specific structure (Fabrikant et al., 2004).

Refinements of the Nash Equilibrium in Extensive-Form Games

In the specific context of extensive-form games, considerable attention has been devoted to the definition of refinements of the NE. The reason is that, in games with a sequential structure, an NE may prescribe the players suboptimal actions off the equilibrium path, *i.e.*, at those decision points which are never reached if the players play as the equilibrium prescribes.

In order to refine the NE concept, several approaches have been investigated. Among them, *trembling-hand perfection* (introduced by Selten (1975)) received the attention of the majority of the works on equilibrium refinements in the literature. The main idea behind this approach is to introduce the possibility that the players may *tremble*, *i.e.*, play each action with a minimum low-but-non-zero probability. As a result, any information set of the game is reached with positive probability, which ensures that the resulting equilibria prescribe to play optimally everywhere.

Among the plethora of NE refinements based on trembling-hand perfection (see (Van Damme, 1987) for details), the *quasi-perfect equilibrium*, proposed by Van Damme (1984), plays a central role, and it is considered one of the most attractive NE refinement concepts, as argued, *e.g.*, by Mertens (1995). The rationale behind the quasi-perfect equilibrium concept is that every player, in every information set, plays her best response to perturbed, *i.e.*, subject to trembles, strategies of the opponents. Another well-known NE refinement is the *extensive-form perfect equilibrium* introduced by Selten (1975), where, differently from the quasi-perfect equilibrium, the players also take into account their own trembles (in addition to those of the opponents). Notice that addressing the opponents' mistakes only (which is the core idea of quasi perfection) is generally considered as a reasonable assumption, and it also excludes some unreasonable players' strategies (Mertens, 1995). Finally, yet another possibility is to apply trembles to the normal-form representation of the extensive-form game, resulting in the definition of *normal-form perfect equilibrium*. Differently from a quasi-perfect equilibrium, a normal-form perfect equilibrium does not guarantee that the strategies of the players are sequentially rational, and, additionally, quasi-perfection implies normal-form perfection.

Computation of NE refinements has received extensive attention in the literature. In the two-player case, Miltersen and Sørensen (2010) provide algorithms for finding a quasi-perfect equilibrium, while Farina and Gatti (2017a) for finding an extensive-form perfect equilibrium. In particular, Miltersen and Sørensen (2010) show that a strict subset of the quasi-perfect equilibria can be found when the sequence form is subject to a specific perturbation, while Farina and Gatti (2017a) do the same for the extensive-form perfect equilibrium. Iterative algorithms for such perturbed games in the zero-sum extensive-form perfect equilibrium setting were introduced by Kroer et al. (2017) and Farina et al. (2017).

2.3.2 Correlated Equilibrium

The correlated equilibrium assumes that there is an external mediator (usually referred to as the *correlation device*) that privately communicates how to play to the players, who must not have any incentive to deviate from the recommendations. Formally, given any finite game $\Gamma = (N, S, u)$, we let $\mathcal{X} := \Delta(S)$ be the set of *correlated distributions* defined over strategy profiles in S, *i.e.*, each $x \in \mathcal{X}$ satisfies $\sum_{s \in S} x(s) = 1$ and $x(s) \geq 0$ for all $s \in S$. Moreover, overloading notation, we let $u_p(x) := \sum_{s \in S} x(s) u_p(s)$ be player p's expected utility in $x \in \mathcal{X}$.

Definition 2.11 (Correlated Equilibrium). *A correlated distribution $x \in \mathcal{X}$ is a* correlated equilibrium (CE) *of a finite game Γ if, for every player $p \in N$ and pair of strategies $s_p, s'_p \in S_p$, the following holds:*

$$\sum_{s_{-p} \in S_{-p}} x(s_p, s_{-p}) \left(u_p(s_p, s_{-p}) - u_p(s'_p, s_{-p}) \right) \geq 0. \tag{2.3}$$

We can interpret a CE as follows: a correlation device draws some strategy profile $s = (s_1, \dots, s_n) \in S$ from the publicly known correlated distribution x, and, then, it privately communicates each recommendation s_p to player p. Then, the distribution is an equilibrium if no player has an incentive to deviate from the recommendation, as made formal by the *incentive constraints* of Equation (2.3).

The *coarse correlated equilibrium* weakens the original version by only enforcing protection against *a priori* defections, *i.e.*, before the recommendations are revealed to the players (Moulin and Vial, 1978).

Definition 2.12 (Coarse Correlated Equilibrium)**.** *A correlated distribution $x \in \mathcal{X}$ is a* coarse correlated equilibrium (CCE) *of a finite game Γ if, for every player $p \in N$ and strategy $s'_p \in S_p$, the following constraint holds:*

$$\sum_{s \in S} x(s) \left(u_p(s) - u_p(s'_p, s_{-p}) \right) \geq 0. \tag{2.4}$$

From the computational perspective, CEs (and their coarse variant) enjoy some nice properties. For instance, in the basic setting of normal-form games, the problem of computing a CE (or a CCE) can be formulated as an LP, whose size is polynomial in the size of the normal-form game (Shoham and Leyton-Brown, 2008). Moreover, CEs and CCEs can be approximated efficiently by letting the players play iteratively by means of regret minimizing procedures (Hart and Mas-Colell, 2000; Cesa-Bianchi and Lugosi, 2006), which have been shown to converge to an equilibrium in a sub-linear (in the size of the game) number of iterations in many classes of games beyond the normal form (see, *e.g.*, (Hartline et al., 2015; Celli et al., 2019)).

CHAPTER 3

Stackelberg Games and Equilibria

In this chapter, we introduce the core subject of this work: *Stackelberg games* and their corresponding equilibria, called *Stackelberg equilibria*. [1]

Section 3.1 defines Stackelberg games in general, introducing the notation we adopt in the rest of the work. Then, the rest of the chapter surveys the main state-of-the-art computational results on Stackelberg games and equilibria. In particular, Section 3.2 presents the results about the computation of Stackelberg equilibria in single-leader single-follower games, while Sections 3.3 and 3.4 do the same for single-leader multi-follower and multi-leader games, respectively. Finally, Section 3.5 addresses some related works studying variations of the Stackelberg paradigm that are different from the classical one addressed in this work.

3.1 Stackelberg Games

Any finite game has a Stackelberg counterpart where some of the players are *leaders* and the others are *followers*. The former have the ability to commit to a course of play beforehand, while the latter decide how to

[1] In some works, Stackelberg games are also called *leadership* (or *leader-follower*) *games*, while their equilibria are named *leader-follower equilibria*. Here, we use the term Stackelberg since it is the most widely adopted.

play after observing the commitment (Von Stackelberg, 1934; Conitzer and Sandholm, 2006; Von Stengel and Zamir, 2010).

Definition 3.1 (Stackelberg Game)**.** *Given a finite game* $\Gamma = (N, S, u)$*, a* Stackelberg game (SG) *is a tuple* (Γ, L, F) *where* L *and* F *are the sets of* leaders *and* followers*, respectively, with* $N = L \cup F$*.*

When focusing on SGs with a single leader and multiple (*i.e.*, more than one) followers, we adopt the convention that player n is the leader, and we let F be the set of followers, *i.e.*, $N = F \cup \{n\}$. Furthermore, for SGs with a single leader and a single follower, we do not explicitly refer to the sets L and F, as we always assume that player 2 is the leader and player 1 is the follower. For the ease of presentation, we use ℓ and f to denote the leader and the follower, respectively, *i.e.*, $N = \{\ell, f\}$. In both cases, when no ambiguity arises, we simply refer to a SG with its underlying non-Stackelberg finite game $\Gamma = (N, S, u)$.

3.2 Single-Leader Single-Follower Stackelberg Games

In single-leader single-follower SGs, the leader first commits to a mixed strategy, and, then, the follower best responds to the commitment (Conitzer and Sandholm, 2006; Von Stengel and Zamir, 2010). Given a leader's mixed strategy $x_\ell \in \mathcal{X}_\ell$, we define $\mathsf{BR}(x_\ell) := \arg\max_{x_f \in \mathcal{X}_f} u_f(x_\ell, x_f)$ as the set of follower's best responses to x_ℓ. If there are multiple best responses to the same strategy, in order to determine an optimal commitment the leader needs to make an assumption about the follower's tie-breaking scheme. A *follower response function* specifies how the follower responds to any possible mixed-strategy commitment of the leader. Formally:

Definition 3.2. *A* follower response function *is a function* $\tau : \mathcal{X}_\ell \to \mathcal{X}_f$ *such that* $\tau(x_\ell) \in \mathsf{BR}(x_\ell)$ *for every leader's mixed strategy* $x_\ell \in \mathcal{X}_\ell$*.*

Definition 3.3 (Stackelberg Equilibrium)**.** *In an SG with a single leader and a single follower* Γ*, given a follower response function* τ*, a* τ*-Stackelberg equilibrium, if it exists, is a mixed strategy profile* $(x_\ell, \tau(x_\ell)) \in \mathcal{X}_\ell \times \mathcal{X}_f$*:*

$$x_\ell \in \arg\max_{x_\ell \in \mathcal{X}_\ell} u_\ell(x_\ell, \tau(x_\ell)). \tag{3.1}$$

Moreover, a mixed strategy profile $(x_\ell, x_f) \in \mathcal{X}_\ell \times \mathcal{X}_f$ *is a* Stackelberg equilibrium (SE) *of* Γ *if there exists a follower response function* τ *such that* (x_ℓ, x_f) *is a* τ*-Stackelberg equilibrium.*

Intuitively, an SE prescribes the leader to play a utility-maximizing strategy, under the assumption that the follower best responds according to some response function. An SG may admit many SEs depending on how the follower is assumed to break ties. Two notable examples are the *strong* SE and the *weak* SE, where the follower is assumed to break ties either in favor or against the leader. [2] Formally, we let:

- $\tau^{\mathsf{s}} : \mathcal{X}_\ell \to \mathcal{X}_f$ be a follower response function in which the follower always breaks ties in favor of the leader, *i.e.*, for all $x_\ell \in \mathcal{X}_\ell$, it holds $\tau^{\mathsf{s}}(x_\ell) \in \arg\max_{x_f \in \mathsf{BR}(x_\ell)} u_\ell(x_\ell, x_f)$;
- $\tau^{\mathsf{w}} : \mathcal{X}_\ell \to \mathcal{X}_f$ be a follower response function that always prescribes her to break ties against the leader, *i.e.*, for all $x_\ell \in \mathcal{X}_\ell$, it holds $\tau^{\mathsf{w}}(x_\ell) \in \arg\min_{x_f \in \mathsf{BR}(x_\ell)} u_\ell(x_\ell, x_f)$.

Then, the following definitions hold.

Definition 3.4 (Strong Stackelberg Equilibrium)**.** *A* strong Stackelberg equilibrium (SSE) *is a τ^{s}-Stackelberg equilibrium.*

Definition 3.5 (Weak Stackelberg Equilibrium)**.** *A* weak Stackelberg equilibrium (WSE) *is a τ^{w}-Stackelberg equilibrium.*

Strong and weak SEs define the extreme values for the utility that the leader could get when playing an SE. Moreover, while an SSE is always guaranteed to exist (Von Stengel and Zamir, 2010), a WSE may not, since the function $u_\ell(x_\ell, \tau^{\mathsf{w}}(x_\ell))$ does not in general admit a maximum over $\mathcal{X}_\ell$. When a WSE does not exist, it is customary to look at the supremum attained by $u_\ell(x_\ell, \tau^{\mathsf{w}}(x_\ell))$ over $\mathcal{X}_\ell$ (Von Stengel and Zamir, 2010) (see also the example in the proof of Proposition 4.1).

3.2.1 Computing SEs in Single-Leader Single-Follower SGs

The problem of computing an SE [3] is known to be easy in two-player (*i.e.*, single-leader single-follower) normal-form SGs in both the strong and the weak setting, as shown in, respectively, (Conitzer and Sandholm, 2006) and (Von Stengel and Zamir, 2010). The key insight for efficiently solving the problem (in both settings) is that we can restrict the attention, w.l.o.g., to pure-strategy follower's best responses, since $u_\ell(x_\ell, x_f)$ and $u_f(x_\ell, x_f)$ are

[2] The terms *strong* and *weak* were originally introduced by Breton et al. (1988), who first studied these two variants of the SE. In the literature, sometimes the terms *optimistic* and *pessimistic* are used in place of strong and weak (see, *e.g.*, (Coniglio et al., 2017; De Nittis et al., 2018a)).

[3] In the literature related to AI, the problem of computing an SE is also referred to as *computing an optimal strategy to commit to* (see (Conitzer and Sandholm, 2006) and other following related works).

linear functions in x_ℓ and x_f. In the strong case, an SE can be found in polynomial time by solving an LP for each action of the (single) follower (the algorithm is, thus, a multi-LP). Each LP maximizes the expected utility of the leader subject to a set of constraints imposing that the given follower's action is a best response (Conitzer and Sandholm, 2006). As shown in Conitzer and Korzhyk (2011), all these LPs can be encoded into a single LP—a slight variation of the LP that is used to compute a CE. [4] Some works study the equilibrium-finding problem in structured games where the action space is combinatorial. See (Basilico et al., 2017c) for more references.

For what concerns the weak case, Von Stengel and Zamir (2010) study the problem of computing the supremum of the leader's expected utility $u_\ell(x_\ell, \tau^{\mathsf{w}}(x_\ell))$. They show that, for the latter, it suffices to consider the follower's actions which constitute a best response to a full-dimensional region of the leader's strategy space. The multi-LP algorithm the authors propose solves two LPs per action of the follower, one to verify whether the best-response region for that action is full-dimensional (so to discard it if full-dimensionality does not hold) and a second one to compute the best leader's strategy within that best-response region. The algorithm runs in polynomial time. While the authors limit their analysis to computing the supremum of the leader's utility $u_\ell(x_\ell, \tau^{\mathsf{w}}(x_\ell))$, we remark that such value does not always translate into a strategy that the leader can play as, in the general case where the leader's utility does not admit a maximum, there is no leader's strategy giving her a utility equal to the supremum. In such cases, one should rather look for a strategy providing the leader with an expected utility which approximates the value of the supremum. This aspect, which is not addressed in (Von Stengel and Zamir, 2010), will be tackled on the multi-follower case by our work.

Besides normal-form SGs, the literature has devoted considerable attention to two-player Bayesian SGs where the follower can be of different types, mainly due to their relevance in security games. In this setting, it is known that finding an SSE is Poly-APX-complete (Letchford et al., 2009) and that an equilibrium can be found with an MILP (Paruchuri et al., 2008).

Over the last years, the Stackelberg paradigm has also been applied to two-player extensive-form SGs. In particular, Letchford et al. (2009) prove that finding an SSE is NP-hard even in games without nature. Works such as (Bošanský and Cermak, 2015; Cermak et al., 2016; Bošanskỳ et al., 2017) address the problem of computing an SSE in extensive-form games, providing worst-case exponential-time algorithms based on MILPs. In the

[4] In this case, the leader and the follower play correlated strategies under rationality constraints imposed on the follower only, maximizing the leader's expected utility.

context of extensive-form games, attempts have also been made towards the refinement of SEs. In particular, Kroer et al. (2018) introduce the idea of a *robust* SE, where an optimal commitment is found against a worst-case follower's utility model.

3.3 Single-Leader Multi-Follower Stackelberg Games

In SGs with multiple (*i.e.*, at least two) followers, many definitions of SE are possible depending on *how* the followers are assumed to play after observing the leader's mixed-strategy commitment. Here, we present two notable cases, which correspond to assuming that the followers play an NE and a CE, respectively.

3.3.1 Stackelberg-Nash Equilibria

A reasonable choice, which was first introduced and studied by Von Stengel and Zamir (2010), is that the followers play *non-cooperatively* after observing the leader's commitment, thus reaching an NE in the game obtained after fixing the leader's strategy. By letting $\mathcal{X}_F := \times_{p \in F} \mathcal{X}_p$ be the set of all the followers' mixed strategy profiles, we denote with $\mathcal{E}(x_n) \subseteq \mathcal{X}_F$ the set of NEs in the followers' game resulting from the leader's mixed strategy $x_n \in \mathcal{X}_n$. Then, we have the following formal definitions. [5]

Definition 3.6 (Strong Stackelberg-Nash Equilibrium). *In an SG with a single leader and multiple followers* Γ*, a* strong Stackelberg-Nash equilibrium (SSNE) *is a mixed strategy profile* $x = (x_n, x_{-n}) \in \times_{p \in N} \mathcal{X}_p$ *such that:*

$$(x_n, x_{-n}) \in \arg\max_{x_n \in \mathcal{X}_n} \max_{x_{-n} \in \mathcal{E}(x_n)} u_n(x_n, x_{-n}). \tag{3.2}$$

Definition 3.7 (Weak Stackelberg-Nash Equilibrium). *In an SG with a single leader and multiple followers* Γ*, a* weak Stackelberg-Nash equilibrium (WSNE)*, if it exists, is a profile* $x = (x_n, x_{-n}) \in \times_{p \in N} \mathcal{X}_p$ *such that:*

$$(x_n, x_{-n}) \in \arg\max_{x_n \in \mathcal{X}_n} \min_{x_{-n} \in \mathcal{E}(x_n)} u_n(x_n, x_{-n}). \tag{3.3}$$

As for the single-follower case, an SSNE is always guaranteed to exist, while a WSNE may not (Von Stengel and Zamir, 2010). Moreover, notice that, in the basic setting of single-leader single-follower SGs, SSNEs and WSNEs reduce to SSEs and WSEs, respectively.

[5] In this work, when working with single-leader multi-follower SGs, we are only concerned with the two specific cases of strong and weak SEs. However, note that it is possible to define generic SEs even in this setting, as we did in Definition 3.3 for single-leader single-follower SGs. In order to do so, it is sufficient to generalize the concept of response function to the general setting with multiple followers.

We remark that the adoption of either the strong or the weak setting does not correspond to assuming that the followers could necessarily *agree* on a specific equilibrium in a practical application. Rather, by computing an SSNE and a WSNE the leader becomes aware of the largest and smallest utility she can get without having to make any assumptions on which equilibrium the followers would *actually* end up playing if the game resulting from the leader's commitment were to admit more than a single one. What is more, while an SSNE accounts for the best case for the leader, a WSNE accounts for the worst case. In this sense, the computation of a WSNE is paramount in realistic scenarios as, differently from an SSNE, it is robust, guaranteeing the leader a lower bound on the maximum utility she would get independently of how the followers would break ties among multiple equilibria. As we will see, though, this degree of robustness comes at a high computational cost, as computing a WSNE is a much harder task than computing its strong counterpart.

The problem of computing SNEs has already been investigated in n-player normal-form games with $n \geq 3$ (*i.e.*, one leader and at least two followers). In this case, it is known that finding an S/WSNE is not in Poly-APX unless $\mathsf{P} = \mathsf{NP}$ even when there are only two followers (*i.e.*, with $n = 3$) (Basilico et al., 2016, 2017a). As for algorithms, Basilico et al. (2016, 2017a, 2019) show how to formulate the problem of finding an SSNE in n-player normal-form games as a *nonlinear* and *nonconvex* mathematical program, which they solve via spatial branch-and-bound techniques.

3.3.2 Commitment to Correlated Strategies

Some works address single-leader multi-follower SGs in which the followers do not play an NE. [6] In particular, Conitzer and Korzhyk (2011) study what they call *optimal correlated strategies to commit to*, where the leader commits to a utility-maximizing correlated distribution satisfying the incentive constraints (Equation (2.3)) for the followers only. By letting $\mathcal{X}_P^{\mathrm{CE}} \subseteq \mathcal{X}$ be the set of correlated distributions that satisfy the incentive constraints of Equation (2.3) only for a subset of players $P \subseteq N$, we can state the following formal definition:

Definition 3.8 (Optimal Correlated Strategy to Commit to)**.** *In an SG with a single leader and multiple followers* Γ*,* $x \in \mathcal{X}$ *is an* optimal correlated strategy to commit to *if it maximizes the leader's utility* $u_n(x)$ *over* $\mathcal{X}_F^{\mathrm{CE}}$*.*

[6] Here, we only discuss the idea of commitment to correlated strategies introduced by Conitzer and Korzhyk (2011) (see Definition 3.8). Some works address different models, such as, *e.g.*, a setting in which the followers play in a hierarchical fashion, where finding an equilibrium is NP-hard Conitzer and Sandholm (2006).

Conitzer and Korzhyk (2011) show that finding an optimal correlated strategy to commit to is easy in n-player normal-form games, as the problem can be cast as an LP, which is a variation of the LP for finding a CE.

Notice that optimal correlated strategies to commit to coincide with SSEs in the basic setting of single-leader single-follower SGs, as the follower's incentive constraints (Equation (2.3)) reduce to the best response conditions. Thus, the two models provide the same leader's utility in single-leader single-follower SGs. However, optimal correlated strategies to commit to may be strictly better in SGs with two or more followers (see (Conitzer and Korzhyk, 2011) for an example).

3.4 Multi-Leader Stackelberg Games

Settings including multiple leaders are widely unexplored in the literature. In spite of this, many real-world applications naturally involve more than one player with competitive advantages, playing the role of leader. Some scenarios are, *e.g.*, network platforms with premium (prioritized) users, markets where a group of firms forms a price-determining dominant cartel (Diamantoudi, 2005), and political elections in which some candidates choose policy positions in advance of challengers (Anderson and Glomm, 1992). In the second part of this work, we fill this gap by proposing a novel model that can be applied to any SG with multiple leaders and followers. In the following, we briefly discuss some results related to ours.

Restricted to the security context, there are some works addressing games with multiple uncoordinated defenders (leaders) (Lou and Vorobeychik, 2015; Laszka et al., 2016; Lou et al., 2017; Gan et al., 2018). The common point that unifies all these work is that they enforce Nash-like constraints on the leaders' strategies. However, the resulting models suffer from two major drawbacks: *(i)* an exact equilibrium may not exist, and *(ii)* they strongly rely on problem-specific structures arising in security problems.

The operations research literature provides further works on multi-leader-follower settings, under the name of *mathematical programs with equilibrium constraints* (Luo et al., 1996). They assume that both leaders and followers are subject to Nash constraints, with the latter playing in the game resulting from the leaders' strategies (Pang and Fukushima, 2005; Leyffer and Munson, 2010; Kulkarni and Shanbhag, 2014). Furthermore, other works from the same field focus on oligopoly models with multiple leaders (Sherali, 1984; DeMiguel and Xu, 2009; Sinha et al., 2014). All these works considerably depart from ours, as they use fundamentally different models and lack thorough game theoretic and computational studies.

3.5 Other Stackelberg Paradigms

When studying SGs, we apply the Stackelberg paradigm to finite games following the approach of Von Stengel and Zamir (2010) and Conitzer and Sandholm (2006), *i.e.*, we treat the leader as a special player who seeks for an optimal (in terms of her utility) *strategy to commit to*. In the literature, there are a number of works that apply the Stackelberg paradigm following different approaches. This is the case, *e.g.*, in congestion games. Even if these works address settings that are are substantially different from ours, it is worth discussing how their results relate to our work.

There are some works, such as, *e.g.*, (Roughgarden, 2004; Swamy, 2007; Sharma and Williamson, 2009; Fotakis, 2010), which study congestion games where the leader is an authority whose objective is to minimize the inefficacy (in terms of followers' social welfare) of the NE reached by the followers (*i.e.*, minimize the price of anarchy). This setting is fundamentally different form ours, as we assume that the leader looks for a strategy to commit to that minimizes her own cost, while she is not concerned with the maximization of followers' social welfare. Let us remark that our approach leads to what is usually called SE, while the Stackelberg strategies analyzed in these works are not SEs according to the classical definitions (Conitzer and Sandholm, 2006; Von Stengel and Zamir, 2010).

Moreover, there are other works, such as, *e.g.*, (Leme et al., 2012; de Jong and Uetz, 2014; Correa et al., 2015), which apply the Stackelberg paradigm to congestion games following yet another approach. They assume that the players play sequentially in a predefined order, reaching a *subgame perfect equilibrium* in the extensive-form extension of the original congestion game where each player plays after observing the actions performed by the preceding players. This is different from our setting in two fundamental ways: *(i)* we assume that the followers play simultaneously, rather than sequentially; and *(ii)* these works study the inefficiency (in terms of followers' social welfare) of subgame perfect equilibria, rather than the computational problem of finding an optimal leader's strategy. Furthermore, we remark that an SE is a subgame perfect equilibrium of a particular extensive-form extension of the original congestion game, known as mixed extension (Von Stengel and Zamir, 2010). In this extended game, the leader first commits to a mixed strategy (having a continuum of actions), and, then, the followers observe it and play simultaneously, reaching an NE. This is different from the extensive-form extension studied in the work by Leme et al. (2012) and its follow-ups, where only pure-strategy commitments are possible and the followers play sequentially.

Part I

Stackelberg Games with Multiple Followers

CHAPTER 4

Computing Stackelberg-Nash Equilibria in Normal-Form Stackelberg Games

In this chapter, we study the problem of finding SNEs in n-player normal-form SGs. We focus on the case where the followers are restricted to pure strategies. This restriction is motivated by two reasons. First, while the unrestricted problem is already hard with two followers (as shown by Basilico et al. (2016, 2017a)), it is not known whether the restriction to followers playing pure strategies makes the problem easier or not. Secondly, many games admit pure-strategy NEs, among which potential games (Monderer and Shapley, 1996), toll-setting problems (Labbé and Violin, 2016), and congestion games (Rosenthal, 1973) (see also Chapter 6).

We start, in Section 4.1, introducing the rigorous definitions of the problems analyzed in this chapter. First, we briefly address the *strong* version of the problem (showing that it can be solved in polynomial time), and, then, we formally define its *weak* variant, on which we focus our attention entirely in the rest of the chapter. In particular, Section 4.2 studies the computational complexity of finding WSNEs in n-player normal-form SGs (with followers restricted to pure strategies). Then, Sections 4.3 and 4.4 focus on the algorithmic aspects of the problem.

4.1 The Problem and Its Formulation

In this section, we formally define the problem analyzed in this chapter and show some preliminary results related to it. For the ease of notation, we define $A_F := A_{-n} = \times_{p \in F} A_p$ as the set of followers' action profiles, *i.e.*, the set of all the collections of followers' actions. We also assume, unless otherwise stated, $m_p = m$ for every player $p \in N$, where m denotes the number of actions available to each player. This is without loss of generality, as one can always introduce additional actions with a utility small enough to guarantee that they would never be played, thus obtaining a game where each player has the same number of actions.

We tackle the problem of computing an SNE in n-player normal-form SGs where the followers play a pure NE once they have observed the leader's mixed-strategy commitment. We refer to a *strong Stackelberg-pure-Nash equilibrium* (SSPNE) when the followers play a pure NE which maximizes the leader's utility, and to a *weak Stackelberg-pure-Nash equilibrium* (WSPNE) when they seek a pure NE minimizing the leader's utility.

First, we briefly discuss SSPNEs, showing that an equilibrium can be computed in polynomial time (in the size of the n-player normal-form game given as input). Then, we formally define WSPNEs, which are the focus of our attention in the rest of this chapter.

4.1.1 The Strong Case

An SSPNE can be found by solving the following *bilevel programming problem*:

$$\begin{aligned}
\max_{x_n, x_{-n}} \quad & \sum_{a \in A} U_n^a x_1^{a_1} x_2^{a_2} \dots x_n^{a_n} \\
\text{s.t.} \quad & x_n \in \Delta_n \\
& x_p \in \arg\max_{x_p} \quad \sum_{a \in A} U_p^a x_1^{a_1} x_2^{a_2} \dots x_n^{a_n} \qquad \forall p \in F \\
& \qquad\quad \text{s.t.} \qquad x_p \in \Delta_p \cap \{0,1\}^{m_p}.
\end{aligned} \tag{4.1}$$

Note that, due to the integrality constraints on x_p for all $p \in F$, each follower can play a single action with probability 1. By imposing the $\arg\max$ constraint for each $p \in F$, the formulation guarantees that each follower plays a best-response action a_p, thus guaranteeing that the action profile $a_{-n} = (a_1, \dots, a_{n-1})$ with, for all $a_p \in A_p$, $a_p = 1$ if and only if $x_p^{a_p} = 1$, be an NE for the given x_n. It is crucial to note that the maximization in the upper level is carried out not only w.r.t. x_n, but also w.r.t. x_{-n}. This way, if the followers' game admits multiple NEs for the chosen x_n, optimal solutions to Problem (4.1) are then guaranteed to contain followers' action profiles which maximize the leader's utility.

The following theorem shows that computing an SSPNE is easy.

Theorem 4.1. *In an n-player normal-form SG, an SSPNE can be computed in polynomial time by solving a multi-LP.*

Proof. It suffices to enumerate, in $O(m^{n-1})$, all the followers' action profiles $a_{-n} \in A_F$ and, for each of them, solve an LP to: *(i)* check whether there is a strategy vector x_n for the leader for which the action profile a_{-n} is an NE and *(ii)* find, among all such strategy vectors x_n, one which maximizes the leader's utility. The action profile a_{-n} which, with the corresponding x_n, yields the largest expected utility for the leader is an SSPNE. Given a followers' action profile a_{-n}, *(i)* and *(ii)* can be carried out in polynomial time by solving the following LP, where the second constraint guarantees that $a_{-n} = (a_1, \ldots, a_{n-1})$ is a pure NE in the followers' game for any of its solutions x_n:

$$
\begin{aligned}
\max_{x_n} \quad & \sum_{a_n \in A_n} U_n^{a_{-n}, a_n} x_n^{a_n} \\
\text{s.t.} \quad & \sum_{a_n \in A_n} U_p^{a_{-n}, a_n} x_n^{a_n} \geq \sum_{a_n \in A_n} U_p^{a_1 \ldots a'_p \ldots a_{n-1} a_n} x_n^{a_n} \quad \forall p \in F, a'_p \in A_p \\
& x_n \in \Delta_n .
\end{aligned}
$$

As the size of an instance of the problem is bounded from below by m^n, one can enumerate over the set of the followers' action profiles (of cardinality m^{n-1}) in polynomial time. The polynomiality of the overall algorithm follows due to linear programming being solvable in polynomial time. □

4.1.2 The Weak Case

The computation of a WSPNE amounts to solving the following *bilevel problem*:

$$
\begin{aligned}
\sup_{x_n} \min_{x_{-n}} \quad & \sum_{a \in A} U_n^a x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n} \\
\text{s.t.} \quad & x_n \in \Delta_n \\
& x_p \in \operatorname*{arg\,max}_{x_p} \sum_{a \in A} U_p^a x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n} \quad \forall p \in F \\
& \qquad \text{s.t.} \quad x_p \in \Delta_p \cap \{0,1\}^{m_p} .
\end{aligned}
\tag{4.2}
$$

There are two differences between this problem and Problem (4.1): the presence of the $\min$ operator in the objective function and the fact that Problem (4.2) calls for a $\sup$ rather than for a $\max$. The former guarantees that, in the presence of many pure NEs in the followers' game for the chosen x_n, one which minimizes the leader's utility is selected. The $\sup$ operator is introduced because, as illustrated in Subsection 4.1.3, Problem (4.2) does not admit a maximum in the general case.

Throughout the paper, we will compactly refer to Problem (4.2) as

$$\sup_{x_n \in \Delta_n} f(x_n),$$

where f is the leader's utility in the weak case, defined as a function of x_n. Since a pure NE may not exist for every leader's strategy x_n, we define $\sup_{x_n \in \Delta_n} f(x_n) = -\infty$ whenever there is no x_n such that the resulting followers' game admits a pure NE. Note that f is always bounded from above when assuming bounded payoffs and, thus, $\sup_{x_n \in \Delta_n} f(x_n) < \infty$.

4.1.3 Some Preliminary Results

First, we provide an example showing that Problem (4.2) may not admit a maximum. This is provided in the proof of the following proposition.

Proposition 4.1. *In an n-player normal-form SG, Problem* (4.2) *may not admit a* $\max$ *even if the followers' game admits a pure NE for every leader's mixed strategy $x_n \in \Delta_n$.*

Proof. Consider a game with $n = 3$, $A_1 = \{a_1^1, a_1^2\}$, $A_2 = \{a_2^1, a_2^2\}$, $A_3 = \{a_3^1, a_3^2\}$. The matrices reported below are the utility matrices for, respectively, the case where the leader plays action a_3^1, action a_3^2, or the strategy vector $x_3 = (1 - \rho, \rho)$ for some $\rho \in [0, 1]$ (the third matrix is the convex combination of the first two with weights x_3):

	a_2^1	a_2^2
a_1^1	1,1,0	2,2,5
a_1^2	$\frac{1}{2}$,$\frac{1}{2}$,1	1,1,0

a_3^1

	a_2^1	a_2^2
a_1^1	0,0,0	2,2,10
a_1^2	$\frac{1}{2}$,$\frac{1}{2}$,1	0,0,0

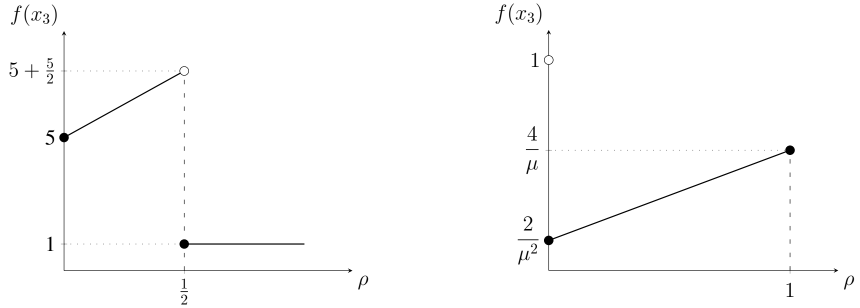
a_3^2

	a_2^1	a_2^2
a_1^1	$1-\rho$,$1-\rho$,0	2,2,$5+5\rho$
a_1^2	$\frac{1}{2}$,$\frac{1}{2}$,1	$1-\rho$,$1-\rho$,0

$x_3 = (1 - \rho, \rho)$

It is easy to verify that (a_1^1, a_2^2, a_3^2) is the unique SSPNE (as it achieves the largest leader's payoff, no mixed strategy would yield a better utility).

In an WSPNE, the leader induces the followers' game in the third matrix by playing $x_3 = (1 - \rho, \rho)$. For $\rho < \frac{1}{2}$, (a_1^1, a_2^2) is the unique NE, giving



(a) *Leader's utility in the normal-form SG in the proof of Proposition 4.1, showing that Problem* (4.2) *may not admit a maximum.*

(b) *Leader's utility in the normal-form SG in the proof of Proposition 4.2, plotted as a function of ρ, where $x_3 = (1-\rho, \rho)$.*

Figure 4.1: *Leader's utility in the normal-form SGs used for Propositions 4.1 and 4.2.*

the leader a utility of $5 + 5\rho$. For $\rho \geq \frac{1}{2}$, there are two NEs, (a_1^1, a_2^2) and (a_1^2, a_2^1), with a utility of, respectively, $5 + 5\rho$ and 1. Since in a WSPNE the latter is selected, we conclude that the leader's utility is equal to $5 + 5\rho$ for $\rho < \frac{1}{2}$ and to 1 for $\rho \geq \frac{1}{2}$ (see Figure 4.1a for an illustration). Thus, Problem (4.2) admits a supremum of value $5 + \frac{5}{2}$, but not a maximum. $\square$

We remark that the result in Proposition 4.1 is in line with a similar result shown in (Von Stengel and Zamir, 2010) for the single-leader single-follower case, as well as with those which hold for general pessimistic bilevel problems (Zemkoho, 2016).

The relevance of computing a weak SPNE is highlighted by the following proposition:

Proposition 4.2. *In an n-player normal-form SG with payoffs in $[0, 1]$, the leader's utility in a WSPNE can be arbitrarily worse than that in an SSPNE. Moreover, the utility obtained by perturbing the leader's strategy in a neighborhood of an SSPNE can be arbitrarily worse than that one in a WSPNE.*

Proof. Consider the following normal-form SG where $n = 3$, $A_1 = \{a_1^1, a_1^2\}$, $A_2 = \{a_2^1, a_2^2\}$, $A_3 = \{a_3^1, a_3^2\}$, parametrized by $\mu > 4$:

a_3^1:

	a_2^1	a_2^2
a_1^1	0,0,0	0,0,1
a_1^2	1,1,$\frac{2}{\mu^2}$	0,0,0

a_3^2:

	a_2^1	a_2^2
a_1^1	$\frac{1}{2}$,$\frac{1}{2}$,0	0,0,0
a_1^2	1,1,$\frac{4}{\mu}$	$\frac{1}{2}$,$\frac{1}{2}$,0

$x_3 = (1-\rho, \rho)$:

	a_2^1	a_2^2
a_1^1	$\frac{\rho}{2}$,$\frac{\rho}{2}$,0	0,0,$1-\rho$
a_1^2	1,1,$\frac{2+(4\mu-2)\rho}{\mu^2}$	$\frac{\rho}{2}$,$\frac{\rho}{2}$,0

Let $x_3 = (1-\rho, \rho)$. The followers' game admits the NE (a_1^2, a_2^1) for all values of ρ (with leader's utility $\frac{2+(4\mu-2)\rho}{\mu^2}$) and the NE (a_1^1, a_2^2) for $\rho = 0$ (with leader's utility 1). Therefore, the game admits a unique SSPNE achieved at $\rho = 0$ (utility 1), and a unique WSPNE achieved at $\rho = 1$ (utility $\frac{4}{\mu}$). See Figure 4.1b for an illustration of the leader's utility function.

To show the first part of the claim, it suffices to observe that the ratio between the leader's utility in an SSPNE and that one in a WSPNE, which is equal to $\frac{\mu}{4}$, becomes arbitrarily large when letting $\mu \to \infty$.

As to the second part of the claim, after perturbing the value that x_3 takes in the unique SSPNE by any arbitrarily small $\epsilon > 0$ (*i.e.*, $x_3 = (1-\epsilon, \epsilon)$) we obtain a leader's utility of $\frac{2+4\mu\epsilon}{\mu^2}$, whose ratio w.r.t. the utility of $\frac{4}{\mu}$ in the unique WSPNE becomes again arbitrarily large for $\mu \to \infty$. $\square$

4.2 Computational Complexity of Finding WSPNEs

In this section, we focus on the problem of computing a WSPNE for n-player normal-form SGs. In Subsection 4.2.1, we show that the problem is NP-hard for $n \geq 3$ (*i.e.*, with at least two followers). Moreover, in Subsection 4.2.2 we prove that for $n \geq 4$ (*i.e.*, for games with at least three followers) the problem is inapproximable, being *not* in Poly-APX unless P = NP, *i.e.*, it cannot be approximated, in polynomial time, to within any polynomial multiplicative factor. We introduce two reductions, a non approximation-preserving one which is valid for $n \geq 3$ and another one only valid for $n \geq 4$ but approximation-preserving.

In decision form, the problem of computing a WSPNE reads:

Definition 4.1 (WSPNE). *Given an n-player normal-form SG with $n \geq 3$ players and a finite number K, is there a WSPNE in which the leader achieves a utility greater than or equal to K?*

In Section 4.2.1, we show that WSPNE is NP-complete by polynomially reducing to it Independent Set (one of Karp's original 21 NP-complete problems (Karp, 1972)), which, in decision form, reads as follows:

Definition 4.2 (IND-SET). *Given an undirected graph $\mathcal{G} := (V, E)$ and an integer $J \leq |V|$, does $\mathcal{G}$ contain an* independent set *(a subset of vertices $V' \subseteq V : \forall u, v \in V'$, $\{u, v\} \notin E$) of size greater than or equal to J?*

In Subsection 4.2.2, we the inapproximability result for the case with at least three followers by means of a polynomial reduction from 3-SAT (another of Karp's 21 NP-complete problems (Karp, 1972)). 3-SAT reads:

Definition 4.3 (3-SAT). *Given a collection C of clauses defined on a finite set V of Boolean variables, with $|\phi| = 3$ for every $\phi \in C$, is there a truth assignment for V which satisfies all the clauses in C?*

4.2.1 NP-Completeness

Before presenting our reduction, we introduce the following class of games:

Definition 4.4. *Given two numbers $b, c \in \mathbb{Q}$ with $1 > c > b > 0$ and an integer $k \geq 1$, let $\Gamma_b^c(k)$ be a class of normal-form games with three players ($n = 3$), the first two having $k+1$ actions each with action sets $A_1 = A_2 = A = \{1, ..., k, \chi\}$ and the third one having k actions with action set $A_3 = A \setminus \{\chi\}$, such that, for every third player's action $a_3 \in A \setminus \{\chi\}$, the other players play a game where:*

- *the payoffs on the main diagonal (where both players play the same action) satisfy $U_1^{a_3 a_3 a_3} = U_2^{a_3 a_3 a_3} = 1, U_1^{\chi\chi a_3} = c, U_2^{\chi\chi a_3} = b$ and, for every $a_1 \in A \setminus \{a_3, \chi\}$, $U_1^{a_1 a_1 a_3} = U_2^{a_1 a_1 a_3} = 0$;*
- *for every $a_1, a_2 \in A \setminus \{\chi\}$ with $a_1 \neq a_2$, $U_1^{a_1 a_2 a_3} = U_2^{a_1 a_2 a_3} = b$;*
- *for every $a_2 \in A \setminus \{\chi\}$, $U_1^{\chi a_2 a_3} = c$ and $U_2^{\chi a_2 a_3} = 0$;*
- *for every $a_1 \in A \setminus \{\chi\}$, $U_1^{a_1 \chi a_3} = 1$ and $U_2^{a_1 \chi a_3} = 0$.*

No restrictions are imposed on the third player's payoffs.

See Figure 4.2 for an illustration of one such game $\Gamma_b^c(k)$ with $k = 3$, parametric in b and c. The special feature of $\Gamma_b^c(k)$ games is that, no matter which mixed strategy the third player (the leader) commits to, with the exception of (χ, χ) only the diagonal outcomes can be pure NEs in the resulting followers' game. Moreover, for every subset of diagonal outcomes there is a leader's strategy such that this subset precisely corresponds to the set of all pure NEs in the followers' game. Formally:

Proposition 4.3. *A $\Gamma_b^c(k)$ game with $c \leq \frac{1}{k}$ admits, for all $\mathcal{D} \subseteq \{(a_1, a_1) : a_1 \in A \setminus \{\chi\}\}$ with $\mathcal{D} \neq \varnothing$, a leader's strategy $x_3 \in \Delta_3$ such that the outcomes $(a_1, a_1) \in \mathcal{D}$ are exactly the pure NEs in the followers' game.*

Proof. First, observe that the followers' payoffs that are not on the main diagonal are independent of the leader's strategy x_3. Thus, any outcome (a_1, a_2) with $a_1, a_2 \in A \setminus \{\chi\}$ and $a_1 \neq a_2$ cannot be an NE, as the first follower would deviate by playing action χ so to obtain a utility $c > b$. Analogously, any outcome (χ, a_2) with $a_2 \in A \setminus \{\chi\}$ cannot be an NE because the second follower would deviate by playing χ (since $b > 0$).

	1	2	3	χ
1	1,1,0	b,b,0	b,b,0	1,0,0
2	b,b,0	0,0,$\frac{-1-c}{c}$	b,b,0	1,0,0
3	b,b,0	b,b,0	0,0,1	1,0,0
χ	c,0,0	c,0,0	c,0,0	c,b,0

1

	1	2	3	χ
1	0,0,$\frac{-1-c}{c}$	b,b,0	b,b,0	1,0,0
2	b,b,0	1,1,0	b,b,0	1,0,0
3	b,b,0	b,b,0	0,0,$\frac{-1-c}{c}$	1,0,0
χ	c,0,0	c,0,0	c,0,0	c,b,0

2

	1	2	3	χ
1	0,0,1	b,b,0	b,b,0	1,0,0
2	b,b,0	0,0,$\frac{-1-c}{c}$	b,b,0	1,0,0
3	b,b,0	b,b,0	1,1,0	1,0,0
χ	c,0,0	c,0,0	c,0,0	c,b,0

3

Figure 4.2: *A $\Gamma_b^c(k)$ game with $k = 3$. The third player (the leader) selects a matrix, while the first and the second players (the followers) select rows and columns, respectively. The third player's payoffs are defined starting from the graph in Figure 4.4, as explained in the proof of Theorem 4.2.*

The same holds for any outcome (a_1, χ) with $a_1 \in A \setminus \{\chi\}$, since the second follower would be better off playing another action (as $b > 0$). The last outcome on the diagonal, (χ, χ), cannot be an NE either, as the first follower would deviate from it (as she would get c in it, while she can obtain $1 > c$ by deviating).

As a result, the only outcomes which can be pure NEs are those in $\{(a_1, a_1) : a_1 \in A \setminus \{\chi\}\}$. When the leader plays a pure strategy $a_3 \in A \setminus \{\chi\}$, the unique pure NE in the followers' game is (a_3, a_3) as, due to providing the followers with their maximum payoff, they would not deviate from it. Outcomes (a_1, a_1) with $a_1 \in A \setminus \{\chi, a_3\}$ are not NEs as, with them, the first follower would get $0 < c$. In general, if the leader plays an arbitrary mixed strategy $x_3 \in \Delta_3$ the resulting followers' game is such that the payoffs in (a_3, a_3) with $a_3 \in A \setminus \{\chi\}$ are $(x_3^{a_3}, x_3^{a_3})$. Noticing that (a_3, a_3) is an equilibrium if and only if $x_3^{a_3} \geq c$ (as, otherwise, the first follower would deviate by playing action χ), we conclude that the set of pure NEs in the followers' game is defined as follows: $\{(a_3, a_3) : x_3^{a_3} \geq c\}$.

In order to guarantee that, for every possible $\mathcal{D} \subseteq \{(a_1, a_1) : a_1 \in A \setminus \{\chi\}\}$ with $\mathcal{D} \neq \varnothing$, there is a leader's strategy such that $\mathcal{D}$ contains all the pure NEs of the followers' game, we must properly choose the value of c. Choosing $c \leq \frac{1}{k}$ suffices, as, for any set $\mathcal{D}$, the leader's strategy $x_3 \in \Delta_3$ such that $x_3^{a_3} = \frac{1}{|\mathcal{D}|}$ for every $a_3 \in A \setminus \{\chi\}$ with $(a_3, a_3) \in \mathcal{D}$ induces a followers' game in which all the outcomes in $\mathcal{D}$ are NEs. □

Notice that the followers' game always admits a pure NE for any leader's commitment x_3 in a $\Gamma_b^c(k)$ game with $c \leq \frac{1}{k}$. As shown in Figure 4.3 for $k = 3$, the leader's strategy space Δ_3 is partitioned into $2^k - 1$ regions, each corresponding to a subset of $\{(a_1, a_1) : a_1 \in A \setminus \{\chi\}\}$ containing those diagonal outcomes which are the only pure NEs in the followers' game. Hence, in a $\Gamma_b^c(k)$ game with $c \leq \frac{1}{k}$ the number of combinations of outcomes which may constitute the set of pure NEs in the followers' game is exponential in r, and, thus, in the size of the game instance.

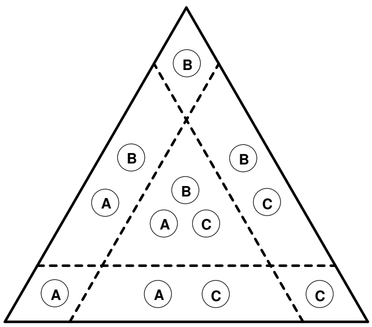


Figure 4.3: *A $\Gamma_b^c(k)$ game with $k = 3$ and $c \leq \frac{1}{k}$. The leader's strategy space Δ_3 is partitioned into $2^k - 1$ regions, one per subset of $\{(a_1, a_1) : a_1 \in A \setminus \{\chi\}\}$ (the three NEs in the followers' game, $(1, 1)$, $(2, 2)$, and $(3, 3)$, are labeled* A, B, C*).*

Relying on Proposition 4.3, we can establish the following result:

Theorem 4.2. WSPNE *is strongly* NP*-complete even for $n = 3$.*

Proof. For the sake of clarity, we split the proof over multiple steps.

Mapping. Given an instance of IND-SET, *i.e.*, an undirected graph $\mathcal{G} = (V, E)$ and a positive integer J, we construct a special instance $\Gamma(\mathcal{G})$ of WSPNE of class $\Gamma_b^c(k)$ as follows. Assuming an arbitrary labeling of the vertices $V := \{v_1, v_2, ..., v_{|V|}\}$, let $\Gamma(\mathcal{G})$ be an instance of $\Gamma_b^c(k)$ with $k := |V|$, $c < \frac{1}{r}$ and $0 < b < c < 1$, where each action $a_1 \in A \setminus \{\chi\}$

is associated with a vertex $v_{a_1} \in V$. In compliance with Definition 4.4, in which no constraints are specified for the leader payoffs, we define:

- for any pair of vertices $v_{a_1}, v_{a_2} \in V$: $U_3^{a_1 a_1 a_2} = U_3^{a_2 a_2 a_1} = \frac{-1-c}{c}$ if $\{v_{a_1}, v_{a_2}\} \in E$, and $U_3^{a_1 a_1 a_2} = U_3^{a_2 a_2 a_1} = 1$ otherwise;
- for every $a_3 \in A \setminus \{\chi\}$: $U_3^{a_3 a_3 a_3} = 0$ and $U_3^{\chi\chi a_3} = 0$;
- for every $a_3 \in A \setminus \{\chi\}$ and for every $a_1, a_2 \in A$ with $a_1 \neq a_2$: $U_3^{a_1 a_2 a_3} = U_3^{a_2 a_1 a_3} = 0$.

As an example, Figure 4.4 illustrates an instance of IND-SET from which the game depicted in Figure 4.2 is obtained by applying our reduction. Finally, let $K := \frac{J-1}{J}$. Notice that this transformation can be carried out in time polynomial in the number of vertices $|V|$. W.l.o.g., we assume that the graph $\mathcal{G}$ contains no isolated vertices. Indeed, it is always possible to remove all the isolated vertices from $\mathcal{G}$ (in polynomial time), solve the problem on the residual graph, and, then, add the isolated vertices back to the independent set that has been found, still obtaining an independent set.

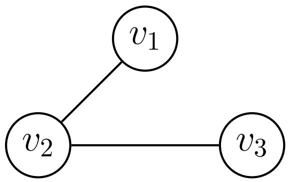


Figure 4.4: *Undirected graph* $\mathcal{G} = (V, E)$, $V = \{v_1, v_2, v_3\}$, $E = \{\{v_1, v_2\}, \{v_2, v_3\}\}$.

If. We show that, if the graph $\mathcal{G}$ contains an independent set of size greater than or equal to J, then $\Gamma(\mathcal{G})$ admits a WSPNE with leader's utility greater than or equal to K. Let V^* be an independent set with $|V^*| = J$. Consider the case in which outcomes (a_1, a_1), with $v_{a_1} \in V^*$, are the only pure NEs in the followers' game, and assume that the leader's strategy x_3 is such that $x_3^{a_3} = \frac{1}{|V^*|}$ if $v_{a_3} \in V^*$ and $x_3^{a_3} = 0$ otherwise. Since, by construction, $U_3^{a_1 a_1 a_3} = 1$ for all $a_3 \in A \setminus \{\chi, a_1\}$, the leader's utility at an equilibrium (a_1, a_1) is:

$$\sum_{a_3 \in A \setminus \{\chi\}} U_3^{a_1 a_1 a_3} x_3^{a_3} = \sum_{a_3 \in A \setminus \{\chi, a_1\}} U_3^{a_1 a_1 a_3} x_3^{a_3} = \sum_{a_3 \in A \setminus \{\chi, a_1\}} x_3^{a_3} = \frac{|V^*| - 1}{|V^*|} = K.$$

Only if. We show that, if $\Gamma(\mathcal{G})$ admits a WSPNE with leader's utility greater than or equal to K, then $\mathcal{G}$ contains an independent set of size greater than or equal to J. Due to Proposition 4.3, at any WSPNE the leader plays a strategy $\bar{x}_3$ inducing a set of pure NEs in the followers' game corresponding to $\mathcal{D}^* = \{(a_3, a_3) : \bar{x}_3^{a_3} \geq c\}$. We now show that the leader would never play two actions $a_1, a_2 \in A \setminus \{\chi\}$ and $\{v_{a_1}, v_{a_2}\} \in E$ with probability greater than or equal to c in a WSPNE. By contradiction, assume that the leader's equilibrium strategy $\bar{x}_3$ is such that $\bar{x}_3^{a_1}, \bar{x}_3^{a_2} \geq c$. When the followers play the equilibrium (a_1, a_1) (the same holds for (a_2, a_2)), the leader's utility is:

$$\sum_{a_3 \in A \setminus \{\chi\}} U_3^{a_1 a_1 a_3} \bar{x}_3^{a_3} = \sum_{a_3 \in A \setminus \{\chi, a_1, a_2\}} U_3^{a_1 a_1 a_3} \bar{x}_3^{a_3} + \bar{x}_3^{a_2} \frac{-1-c}{c}.$$

In the right-hand side, the first term is < 1 (as the leader's payoffs are ≤ 1 and $\sum_{a_3 \in A \setminus \{\chi, a_1, a_2\}} \bar{x}_3^{a_3} = 1 - \bar{x}_3^{a_1} - \bar{x}_3^{a_2} < 1$, since $\bar{x}_3^{a_1}, \bar{x}_3^{a_2} \geq c$). The second term is less than or equal to $c \frac{-1-c}{c} = -1 - c$ (as $\bar{x}_3^{a_2} \geq c$), which is strictly less than -1. It follows that, since (a_1, a_1) (or, equivalently, (a_2, a_2)) always provides the leader with a negative utility, she would never play $\bar{x}_3$ in an equilibrium. This is because, by playing a pure strategy she would obtain a utility of at least zero (as the followers' game admits a unique pure NE giving her a zero payoff when she plays a pure strategy). As a result, we have $U_3^{a_3 a_3 a_3} = 0$ for every action a_3 such that $\bar{x}_3^{a_3} \geq c$ and $U_3^{a_1 a_1 a_3} = 1$ for every other action a_1 such that $\bar{x}_3^{a_1} \geq c$ (since v_{a_1} and v_{a_3} are not connected by an edge).

Note that, in any equilibrium $(a_1, a_1) \in \mathcal{D}^*$, the leader's utility is:

$$\sum_{a_3 \in A \setminus \{\chi\}} U_3^{a_1 a_1 a_3} \bar{x}_3^{a_3} = \sum_{a_3 \in A \setminus \{\chi, a_1\} : \bar{x}_3^{a_3} \geq c} U_3^{a_1 a_1 a_3} \bar{x}_3^{a_3} + \sum_{a_3 \in A \setminus \{\chi\} : \bar{x}_3^{a_3} < c} U_3^{a_1 a_1 a_3} \bar{x}_3^{a_3},$$

where, in the first summation in the right-hand side, each payoff $U_3^{a_1 a_1 a_3}$ is equal to 1 (as $\bar{x}_3^{a_1} \geq c$ and $\bar{x}_3^{a_3} \geq c$). We show that the same holds for each payoff $U_3^{a_1 a_1 a_3}$ appearing in the second summation. By contradiction, assume that there exists an action $a_3 \in A \setminus \{\chi\}$ such that $\bar{x}_3^{a_3} < c$ and $U_3^{a_1 a_1 a_3} = \frac{-1-c}{c}$ for some equilibrium $(a_1, a_1) \in \mathcal{D}^*$. By shifting all the probability that $\bar{x}_3$ places on a_3 to actions a_1 such that $(a_1, a_1) \in \mathcal{D}^*$ (so that $\bar{x}_3^{a_3} = 0$), we obtain a new leader's strategy which induces the same set $\mathcal{D}^*$ of pure NEs in the followers' game. Moreover, the leader's utility in any equilibrium $(a_1, a_1) \in \mathcal{D}^*$ strictly increases if $U_3^{a_1 a_1 a_3} = \frac{-1-c}{c}$, while it stays the same when $U_3^{a_1 a_1 a_3} = 1$. This contradicts the fact that $\bar{x}_3$ is a WSPNE. Thus, all the actions $a_3 \in A \setminus \{\chi\}$ such that $\bar{x}_3^{a_3} < c$ satisfy $U_3^{a_1 a_1 a_3} = 1$ for every equilibrium $(a_1, a_1) \in \mathcal{D}^*$.

As a result, the leader's utility at an equilibrium $(a_3, a_3) \in \mathcal{D}^*$ is $1 - \bar{x}_3^{a_3}$. Since in a WSPNE the leader maximizes her utility in the worst NE, her best choice is to select an $\bar{x}_3$ such that all NEs yield the same utility, that is: $\bar{x}_3^{a_1} = \bar{x}_3^{a_2}$ for every a_1, a_2 with $(a_1, a_1), (a_2, a_2) \in \mathcal{D}^*$. This results in the leader playing all actions a_3 such that $(a_3, a_3) \in \mathcal{D}^*$ with the same probability $\bar{x}_3^{a_3} = \frac{1}{|\mathcal{D}^*|}$, obtaining a utility of $\frac{|\mathcal{D}^*|-1}{|\mathcal{D}^*|} = K$. Therefore, the vertices in the set $\{v_{a_3} : (a_3, a_3) \in \mathcal{D}^*\}$ form an independent set of $\mathcal{G}$ of size $|\mathcal{D}^*| = J$. The reduction is, thus, complete.

NP **membership.** Given a triple (a_1, a_2, x_3) which is encoded with a number of bits that is polynomial w.r.t. the size of the game, we can verify in polynomial time whether (a_1, a_2) is an NE in the followers' game induced by x_3 and whether, when playing (a_1, a_2, x_3), the leader's utility is at least as large as K. The existence of such a triple follows as a consequence of the correctness of either of the two equilibrium-finding algorithms that we propose in Section 4.4—we refer the reader to Section 4.4.2 for a discussion on this. Therefore, we deduce that WSPNE belongs to NP. Moreover, since in the game of the reduction the players' payoffs are encoded with a polynomial number of bits and due to IND-SET being strongly NP-complete, WSPNE is strongly NP-complete. □

4.2.2 Inapproximability

We show now that the search problem of computing a WSPNE is not only NP-hard (due to its decision version, WSPNE, being NP-complete), but it is also difficult to approximate. Since the reduction from IND-SET in Theorem 4.2 is not approximation-preserving, we propose a new one based on 3-SAT (see Definition 4.3). We remark that, differently from our previous reduction (which holds for any number of followers greater than or equal to two), this one requires at least three followers.

In the following, given a literal l (an occurrence of a variable, possibly negated), we define $v(l)$ as its corresponding variable. For a generic clause $\phi = l_1 \vee l_2 \vee l_3$, we denote the ordered set of possible truth assignments to the variables, namely, $x = v(l_1), y = v(l_2)$, and $z = v(l_3)$, by

$$L_\phi = \{xyz, xy\bar{z}, x\bar{y}z, x\bar{y}\bar{z}, \bar{x}yz, \bar{x}y\bar{z}, \bar{x}\bar{y}z, \bar{x}\bar{y}\bar{z}\},$$

where, in each truth assignment, a variable is set to 1 if positive and to 0 if negative. Given a generic 3-SAT instance, we build a corresponding normal-form SG as detailed in the following definition.

Definition 4.5. *Given a* 3-SAT *instance where* $C := \{\phi_1, \dots, \phi_{|C|}\}$ *is a collection of clauses and* $V := \{v_1, \dots, v_{|V|}\}$ *is a set of Boolean variables, and some* $\epsilon \in (0, 1)$*, let* $\Gamma_\epsilon(C, V)$ *be a normal-form SG with four players*

($n = 4$) defined as follows. The fourth player has an action for each variable in V plus an additional one, i.e., $A_4 = \{1, \dots, |V|\} \cup \{w\}$. Each action $a_4 \in \{1, \dots, |V|\}$ is associated with variable v_{a_4}. The other players share the same set of actions A, with $A = A_1 = A_2 = A_3 = \{\varphi_{ca} \mid c \in \{1, \dots, |C|\}, a \in \{1, \dots, 8\}\} \cup \{\chi\}$, where each action φ_{ca} is associated with one of the eight possible assignments of truth to the variables in clause ϕ_c, so that φ_{ca} corresponds to the a-th assignment in the ordered set L_{ϕ_c}. For each player $p \in \{1, 2, 3\}$, we define her utilities as follows:

- *for each $a_4 \in A_4 \setminus \{w\}$ and for each $a_1 \in A \setminus \{\chi\}$ with $a_1 = \varphi_{ca} = l_1 l_2 l_3$, $U_p^{a_1 a_1 a_1 a_4} = 1$ if $v(l_p) = v_{a_4}$ and l_p is a positive literal or $v(l_p) \neq v_{a_4}$ and l_p is negative;*
- *for each $a_4 \in A_4 \setminus \{w\}$ and for each $a_1 \in A \setminus \{\chi\}$ with $a_1 = \varphi_{ca} = l_1 l_2 l_3$, $U_p^{a_1 a_1 a_1 a_4} = 0$ if $v(l_p) = v_{a_4}$ and l_p is a negative literal or $v(l_p) \neq v_{a_4}$ and l_p is positive;*
- *for each $a_1 \in A \setminus \{\chi\}$ with $a_1 = \varphi_{ca} = l_1 l_2 l_3$, $U_p^{a_1 a_1 a_1 w} = 0$ if l_p is a positive literal, while $U_p^{a_1 a_1 a_1 w} = 1$ otherwise;*
- *for each $a_4 \in A_4$ and for each $a_1, a_2, a_3 \in A \setminus \{\chi\}$ such that $a_1 \neq a_2 \vee a_2 \neq a_3 \vee a_1 \neq a_3$, $U_p^{a_1 a_2 a_3 a_4} = \frac{1}{|V|+2}$;*
- *for each $a_4 \in A_4$, $a_3 \in A \setminus \{\chi\}$, and $a_2 \in A \setminus \{\chi\}$ with $a_2 = \varphi_{ca} = l_1 l_2 l_3$, $U_1^{\chi a_2 a_3 a_4} = \frac{1}{|V|+1}$ if l_1 is a positive literal, whereas $U_1^{\chi a_2 a_3 a_4} = \frac{|V|}{|V|+1}$ if l_1 is negative, while $U_2^{\chi a_2 a_3 a_4} = U_3^{\chi a_2 a_3 a_4} = 0$;*
- *for each $a_4 \in A_4$, $a_3 \in A \setminus \{\chi\}$, and $a_1 \in A \setminus \{\chi\}$ with $a_1 = \varphi_{ca} = l_1 l_2 l_3$, $U_2^{a_1 \chi a_3 a_4} = \frac{1}{|V|+1}$ if l_2 is a positive literal, whereas $U_2^{a_1 \chi a_3 a_4} = \frac{|V|}{|V|+1}$ if l_2 is negative, while $U_1^{a_1 \chi a_3 a_4} = 1$ and $U_3^{a_1 \chi a_3 a_4} = 0$;*
- *for each $a_4 \in A_4$, $a_1 \in A \setminus \{\chi\}$, and $a_2 \in A \setminus \{\chi\}$ with $a_2 = \varphi_{ca} = l_1 l_2 l_3$, $U_3^{a_1 a_2 \chi a_4} = \frac{1}{|V|+1}$ if l_3 is a positive literal, whereas $U_3^{a_1 a_2 \chi a_4} = \frac{|V|}{|V|+1}$ if l_3 is negative, while $U_1^{a_1 a_2 \chi a_4} = 0$ and $U_2^{a_1 a_2 \chi a_4} = 1$;*
- *for each $a_4 \in A_4$, $U_1^{a_1 \chi \chi a_4} = U_3^{a_1 \chi \chi a_4} = 1$ and $U_2^{a_1 \chi \chi a_4} = 0$, for all $a_1 \in A \setminus \{\chi\}$;*
- *for each $a_4 \in A_4$, $U_1^{\chi a_2 \chi a_4} = 1$ and $U_2^{\chi a_2 \chi a_4} = U_3^{\chi a_2 \chi a_4} = 0$, for all $a_2 \in A \setminus \{\chi\}$;*
- *for each $a_4 \in A_4$, $U_1^{\chi \chi a_3 a_4} = U_3^{\chi \chi a_3 a_4} = 0$ and $U_2^{\chi \chi a_3 a_4} = 1$, for all $a_3 \in A$.*

The payoff matrix of the fourth player is so defined:

- *for each $a_4 \in A_4$ and for each $a_1 \in A \setminus \{\chi\}$ with $a_1 = \varphi_{ca} = l_1 l_2 l_3$, $U_4^{a_1 a_1 a_1 a_4} = \epsilon$ if the truth assignment identified by φ_{ca} makes ϕ_c false (i.e., whenever, for each $p \in \{1, 2, 3\}$, the clause ϕ_c contains the negation of l_p), while $U_4^{a_1 a_1 a_1 a_4} = 1$ otherwise;*
- *for each $a_4 \in A_4$ and for each $a_1, a_2, a_3 \in A$ such that $a_1 \neq a_2 \vee a_2 \neq a_3 \vee a_1 \neq a_3$, with the addition of the triple (χ, χ, χ), $U_4^{a_1 a_2 a_3 a_4} = 0$.*

Games adhering to Definition 4.5 have some interesting properties, which we formally state in the following Propositions 4.4 and 4.5.

First, we give a characterization of the strategy space of the leader in terms of the set of pure NEs in the followers' game. In particular, given a game $\Gamma_\epsilon(C, V)$, the leader's strategy space Δ_4 is partitioned according to the boundaries $x_4^{a_4} = \frac{1}{|V|+1}$, for $a_4 \in A_4 \setminus \{w\}$, by which Δ_4 is split into $2^{|V|}$ regions, each corresponding to a possible truth assignment to the variables in V. Specifically, in the assignment corresponding to a region, variable v_{a_4} takes value 1 if $x_4^{a_4} \geq \frac{1}{|V|+1}$, while it takes value 0 if $x_4^{a_4} \leq \frac{1}{|V|+1}$. Moreover, for each $a_1 \in A \setminus \{\chi\}$ and $a_1 = \varphi_{ca}$ an outcome (a_1, a_1, a_1) is an NE in the followers' game only in the regions of the leader's strategy space whose corresponding truth assignment is compatible with the one represented by φ_{ca}. For instance, if $\varphi_{ca} = \bar{v}_1 v_2 v_3$ the corresponding outcome is an NE only if $x_4^1 \leq \frac{1}{|V|+1}$, $x_4^2 \geq \frac{1}{|V|+1}$, and $x_4^3 \geq \frac{1}{|V|+1}$ (with no further restrictions on the other probabilities). Formally, we can claim the following:

Proposition 4.4. *Given a game $\Gamma_\epsilon(C, V)$ and an action $a_1 \in A \setminus \{\chi\}$ with $a_1 = \varphi_{ca} = l_1 l_2 l_3$, the outcome (a_1, a_1, a_1) is an NE of the followers' game whenever the leader commits to a strategy $x_4 \in \Delta_4$ such that:*

- $x_4^{a_4} \geq \frac{1}{|V|+1}$ *if $v(l_p) = v_{a_4}$ and l_p is positive, for some $p \in \{1, 2, 3\}$;*
- $x_4^{a_4} \leq \frac{1}{|V|+1}$ *if $v(l_p) = v_{a_4}$ and l_p is negative, for some $p \in \{1, 2, 3\}$;*
- $x_4^{a_4}$ *can be any if $v(l_p) \neq v_{a_4}$ for each $p \in \{1, 2, 3\}$.*

All the other outcomes of the followers' game, i.e., those belonging to the set $\{(a_1, a_2, a_3) : a_1, a_2, a_3 \in A$ with $a_1 \neq a_2 \vee a_2 \neq a_3 \vee a_1 \neq a_3\} \cup \{(\chi, \chi, \chi)\}$, cannot be NEs for any of the leader's commitments.

Proof. Observe that, the followers' payoffs do not depend on the leader's strategy x_4 in the outcomes not in $\{(a_1, a_1, a_1) : a_1 \in A \setminus \{\chi\}\}$. Thus, for every $a_1, a_2, a_3 \in A \setminus \{\chi\}$ such that $a_1 \neq a_2 \vee a_2 \neq a_3 \vee a_1 \neq a_3$ the outcome (a_1, a_2, a_3) cannot be an NE as the first follower would deviate by playing action χ, obtaining a utility at least as large as $\frac{1}{|V|+1}$, instead of $\frac{1}{|V|+2}$. Also, for all $a_2, a_3 \in A \setminus \{\chi\}$ the outcome (χ, a_2, a_3) is not an NE since the second follower would be better off playing χ (as she gets $1 > 0$). Analogously, for all $a_1, a_3 \in A \setminus \{\chi\}$ the outcome (a_1, χ, a_3) cannot be an NE as the third follower would deviate to χ (getting a utility of $1 > 0$). For all $a_3 \in A$, a similar argument also applies to the outcome (χ, χ, a_3) as the first follower would have an incentive to deviate by playing any action different from χ (note that (χ, χ, χ), whose payoffs are defined in the last item of Definition 4.5, is included). Moreover, for all $a_1 \in A \setminus \{\chi\}$ the outcome (a_1, χ, χ) is not an NE as the second follower would deviate to any other action (getting a utility of 1). For all $a_1, a_2 \in A \setminus \{\chi\}$, the same holds for the outcome (a_1, a_2, χ), where the first follower would deviate and play action χ, and for the outcome (χ, a_2, χ) where, for all $a_2 \in \setminus\{\chi\}$, the second follower would deviate and play χ.

Therefore, the only outcomes which can be NEs in the followers' game are those in $\{(a_1, a_1, a_1) : a_1 \in A \setminus \{\chi\}\}$. Assume that the leader commits to an arbitrary mixed strategy $x_4 \in \Delta_4$. For each $a_1 \in A \setminus \{\chi\}$ with $a_1 = \varphi_{ca} = l_1 l_2 l_3$ and for each $p \in \{1, 2, 3\}$, the outcome (a_1, a_1, a_1) provides follower p with a utility of u_p such that:

- $u_p = x_4^{a_4}$ if $v(l_p) = v_{a_4}$ and l_p is a positive literal;
- $u_p = 1 - x_4^{a_4}$ if $v(l_p) = v_{a_4}$ and l_p is a negative literal;

The outcome (a_1, a_1, a_1) is an NE if the following conditions hold:

- $u_p \geq \frac{1}{|V|+1}$ for each $p \in \{1, 2, 3\}$ such that l_p is positive, as otherwise follower p would deviate and play χ;
- $u_p \geq \frac{|V|}{|V|+1}$ for each $p \in \{1, 2, 3\}$ such that l_p is negative, as otherwise follower p would deviate and play χ;

These conditions together with the definition of u_p prove the claim. $\square$

The characterization of the leader's strategy space given in Proposition 4.4 establishes the relationship between the leader's utility in a WSPNE of a game $\Gamma_\epsilon(C, V)$ and the feasibility of the corresponding 3-SAT instance. We highlight it in the following proposition.

Proposition 4.5. *Given a game $\Gamma_\epsilon(C, V)$, the leader's utility in a WSPNE is 1 if and only if the corresponding* 3-SAT *instance is feasible, and it is equal to ϵ otherwise.*

Proof. The result follows form Proposition 4.4. If the 3-SAT instance is a *yes* instance (*i.e.*, if it is feasible), there exists then a strategy $x_4 \in \Delta_4$ such that all the NEs of the resulting followers' game provide the leader with a utility of 1. This is because there is a region corresponding to a truth assignment which satisfies all the clauses. On the other hand, if the 3-SAT instance is a *no* instance (*i.e.*, if it is not satisfiable), then in each region of the leader's strategy space there exits an NE for the followers' game which provides the leader with a utility of ϵ. Therefore, the followers would always play such equilibrium due to the assumption of pessimism. □

We are now ready to state the result.

Theorem 4.3. *With $n \geq 4$ and unless* P = NP*, the problem of computing a WSPNE in an n-player normal-form SG is not in Poly-*APX.

Proof. Given a generic 3-SAT instance, let us build its corresponding game $\Gamma_\epsilon(C, V)$ according to Definition 4.5. This construction can be done in polynomial time as $|A_4| = |V| + 1$ and $|A| = |A_1| = |A_2| = |A_3| = 8|C| + 1$ are polynomials in $|V|$ and $|C|$, and, therefore, the number of outcomes in $\Gamma_\epsilon(C, V)$ is polynomial in $|V|$ and $|C|$. Furthermore, let us select $\epsilon \in \left(0, \frac{1}{2^{|V|}}\right)$ (the polynomiality of the reduction is preserved as $\frac{1}{2^{|V|}}$ is representable in binary encoding with a polynomial number of bits).

By contradiction, let us assume that there exists a polynomial-time approximation algorithm $\mathcal{A}$ capable of constructing a solution to the problem of computing a WSPNE with a multiplicative approximation factor $\frac{1}{\text{poly}(I)}$, where $\text{poly}(I)$ is any polynomial function of the size I of the normal-form game given as input. By Proposition 4.5, it follows that, when applied to $\Gamma_\epsilon(C, V)$, $\mathcal{A}$ would return an approximate solution with value greater than or equal to $1 \cdot \frac{1}{\text{poly}(I)} > \frac{1}{2^{|V|}}$ (for a sufficiently large $|V|$) if and only if the 3-SAT instance is feasible. When the 3-SAT instance is not satisfiable, $\mathcal{A}$ would return a solution with value at most $\frac{1}{2^{|V|}}$. Since this would provide us with a solution to 3-SAT in polynomial time, we conclude that the problem of computing a WSPNE in an n-player normal-form SG cannot be approximated in polynomial time to within any polynomial multiplicative factor unless P = NP. □

4.3 Single-Level Reformulation and Restriction

We propose a single-level reformulation of the problem of computing a WSPNE admitting a supremum but, in general, not a maximum, and a corresponding restriction which always admits optimal (restricted) solutions.

For notational simplicity, we consider the case with $n = 3$ players. Although notationally more involved, the generalization to $n \geq 3$ is straightforward. With only two followers, Problem (4.2), *i.e.*, the bilevel programming formulation we gave in Section 4.1, reads:

$$\begin{aligned}
\sup_{x_3} \min_{x_1, x_2} \quad & \sum_{a_1 \in A_1} \sum_{a_2 \in A_2} \sum_{a_3 \in A_3} U_3^{a_1 a_2 a_3} x_1^{a_1} x_2^{a_2} x_3^{a_3} \\
\text{s.t.} \quad & x_3 \in \Delta_3 \\
& x_1 \in \arg\max_{x_1} \sum_{a_1 \in A_1} \sum_{a_2 \in A_2} \sum_{a_3 \in A_3} U_1^{a_1 a_2 a_3} x_1^{a_1} x_2^{a_2} x_3^{a_3} \\
& \qquad \text{s.t.} \quad x_1 \in \Delta_1 \cap \{0,1\}^m \\
& x_2 \in \arg\max_{x_2} \sum_{a_1 \in A_1} \sum_{a_2 \in A_2} \sum_{a_3 \in A_3} U_2^{a_1 a_2 a_3} x_1^{a_1} x_2^{a_2} x_3^{a_3} \\
& \qquad \text{s.t.} \quad x_2 \in \Delta_2 \cap \{0,1\}^m.
\end{aligned} \tag{4.3}$$

4.3.1 Single-Level Reformulation

In order to cast Problem (4.3) into a single-level problem, we first introduce the following reformulation of the followers' problem:

Lemma 4.1. *The following MILP, parametric in x_3, is an exact reformulation of the followers' problem of finding a pure NE which minimizes the leader's utility given a leader's strategy $x_3 \in \Delta_3$:*

$$\min_{y} \sum_{a_1 \in A_1} \sum_{a_2 \in A_2} y^{a_1 a_2} \sum_{a_3 \in A_3} U_3^{a_1 a_2 a_3} x_3^{a_3} \tag{4.4a}$$

$$\text{s.t.} \sum_{a_1 \in A_1} \sum_{a_2 \in A_2} y^{a_1 a_2} = 1 \tag{4.4b}$$

$$y^{a_1 a_2} \sum_{a_3 \in A_3} (U_1^{a_1 a_2 a_3} - U_1^{a_1' a_2 a_3}) x_3^{a_3} \geq 0 \quad \forall a_1, a_1' \in A_1, a_2 \in A_2 \tag{4.4c}$$

$$y^{a_1 a_2} \sum_{a_3 \in A_3} (U_2^{a_1 a_2 a_3} - U_2^{a_1 a_2' a_3}) x_3^{a_3} \geq 0 \quad \forall a_1 \in A_1, a_2, a_2' \in A_2 \tag{4.4d}$$

$$y^{a_1 a_2} \in \{0,1\} \qquad \forall a_1 \in A_1, a_2 \in A_2. \tag{4.4e}$$

Proof. Note that, in Problem (4.3), a solution to the followers' problem satisfies $x_1^{a_1} = x_2^{a_2} = 1$ for some $(a_1, a_2) \in A_1 \times A_2$ and $x_1^{a_1'} = x_2^{a_2'} = 0$ for all $(a_1', a_2') \neq (a_1, a_2)$. Problem (4.4) encodes this in terms of the variable $y^{a_1 a_2}$ by imposing $y^{a_1 a_2} = 1$ if an only if (a_1, a_2) is an NE minimizing the leader's utility. Due to Constraints (4.4b) and (4.4e), $y^{a_1 a_2}$ is equal to 1 for one and only one pair (a_1, a_2). Due to Constraints (4.4c) and (4.4d), for all (a_1, a_2) such that $y^{a_1 a_2} = 1$ there can be no action $a_1' \in A_1$ (resp., $a_2' \in A_2$) by which the first follower (resp., the second follower) could obtain a better payoff when assuming that the other would play action a_2 (resp., action a_1). This guarantees that (a_1, a_2) be an NE. Also note that Constraints (4.4c) and (4.4d) boil down to the tautology $0 \geq 0$ for any $(a_1, a_2) \in A_1 \times A_2$ with $y^{a_1 a_2} = 0$. By minimizing the objective function (which corresponds to the leader's utility), a pure NE with the desired properties is found. $\square$

To arrive at a single-level reformulation of Problem (4.3), we rely on linear programming duality to restate Problem (4.4) in terms of optimality conditions which do not employ the $\min$ operator. First:

Lemma 4.2. *The linear programming relaxation of Problem* (4.4) *always admits an optimal integer solution.*

Proof. Let us focus on Constraints (4.4c) and analyze , for all $(a_1, a_2) \in A_1 \times A_2$ and $a_1' \in A_1$, the coefficient $\sum_{a_3 \in A_3} (U_1^{a_1 a_2 a_3} - U_1^{a_1' a_2 a_3}) x_3^{a_3}$ which multiplies $y^{a_1 a_2}$. The coefficient is equal to the *regret* the first player would suffer from by not playing action a_1'. If equal to 0, we have the tautology $0 \geq 0$. If the regret is positive, after dividing by $\sum_{a_3 \in A_3} (U_1^{a_1 a_2 a_3} - U_1^{a_1' a_2 a_3}) x_3^{a_3}$ both sides of the constraint we obtain $y^{a_1 a_2} \geq 0$, which is subsumed by the nonnegativity of $y^{a_1 a_2}$. If the regret is negative, after diving both sides of the constraint again by $\sum_{a_3 \in A_3} (U_1^{a_1 a_2 a_3} - U_1^{a_1' a_2 a_3}) x_3^{a_3}$ we obtain $y^{a_1 a_2} \leq 0$, which implies $y^{a_1 a_2} = 0$. A similar reasoning applies to Constraints (4.4d). Let us now define $\mathcal{O}$ as the set of pairs (a_1, a_2) such that there is as least an action a_1' or a_2' for which one of the followers suffers from a strictly negative regret. Relying on $\mathcal{O}$, Problem (4.4) can be rewritten as:

$$
\begin{aligned}
\min_{y} \quad & \sum_{a_1 \in A_1} \sum_{a_2 \in A_2} y^{a_1 a_2} \sum_{a_3 \in A_3} U_3^{a_1 a_2 a_3} x_3^{a_3} \\
\text{s.t.} \quad & \sum_{a_1 \in A_1} \sum_{a_2 \in A_2} y^{a_1 a_2} = 1 \\
& y^{a_1 a_2} = 0 & \forall (a_1, a_2) \in \mathcal{O} \\
& y^{a_1 a_2} \in \{0, 1\} & \forall a_1 \in A_1, a_2 \in A_2.
\end{aligned}
$$

All variables $y^{a_1 a_2}$ with $(a_1, a_2) \in \mathcal{O}$ can be discarded. We obtain a problem with a single constraint imposing that the sum of all the $y^{a_1 a_2}$ variables with $(a_1, a_2) \notin \mathcal{O}$ be equal to 1. The linear programming relaxation of such problem always admits an optimal solution with $y^{a_1 a_2} = 1$ for the pair

(a_1, a_2) which achieves the largest value of $\sum_{a_3 \in A_3} U_3^{a_1 a_2 a_3} x_3^{a_3}$ (ties can be broken arbitrarily), and with $y^{a_1 a_2} = 0$ otherwise. □

As a consequence of Lemma 4.2, we can prove the following:

Theorem 4.4. *The following single-level QCQP is an exact reformulation of Problem* (4.3)*:*

$$\sup_{\substack{x_3, y \\ \beta_1, \beta_2}} \quad \sum_{a_1 \in A_1} \sum_{a_2 \in A_2} y^{a_1 a_2} \sum_{a_3 \in A_3} U_3^{a_1 a_2 a_3} x_3^{a_3} \qquad \text{s.t.} \tag{4.5a}$$

$$\sum_{a_1 \in A_1} \sum_{a_2 \in A_2} y^{a_1 a_2} = 1 \tag{4.5b}$$

$$y^{a_1 a_2} \sum_{a_3 \in A_3} (U_1^{a_1 a_2 a_3} - U_1^{a_1' a_2 a_3}) x_3^{a_3} \geq 0 \quad \forall a_1, a_1' \in A_1, a_2 \in A_2 \tag{4.5c}$$

$$y^{a_1 a_2} \sum_{a_3 \in A_3} (U_2^{a_1 a_2 a_3} - U_2^{a_1 a_2' a_3}) x_3^{a_3} \geq 0 \quad \forall a_1 \in A_1, a_2, a_2' \in A_2 \tag{4.5d}$$

$$\begin{aligned} \sum_{a_1 \in A_1} \sum_{a_2 \in A_2} y^{a_1 a_2} \sum_{a_3 \in A_3} U_3^{a_1 a_2 a_3} x_3^{a_3} &\leq \sum_{a_3 \in A_3} U_3^{a_1 a_2 a_3} x_3^{a_3} + \\ -\sum_{a_1' \in A_1} \beta_1^{a_1 a_2 a_1'} \sum_{a_3 \in A_3} &(U_1^{a_1 a_2 a_3} - U_1^{a_1' a_2 a_3}) x_3^{a_3} + \\ -\sum_{a_2' \in A_2} \beta_2^{a_1 a_2 a_2'} \sum_{a_3 \in A_3} &(U_2^{a_1 a_2 a_3} - U_2^{a_1 a_2' a_3}) x_3^{a_3} \quad \forall a_1 \in A_1, a_2 \in A_2 \end{aligned} \tag{4.5e}$$

$$\sum_{a_3 \in A_3} x_3 = 1 \tag{4.5f}$$

$$\beta_1^{a_1 a_2 a_1'} \geq 0 \qquad \forall a_1, a_1' \in A_1, a_2 \in A_2 \tag{4.5g}$$

$$\beta_2^{a_1 a_2 a_2'} \geq 0 \qquad \forall a_1 \in A_1, a_2, a_2' \in A_2 \tag{4.5h}$$

$$y^{a_1 a_2} \geq 0 \qquad \forall a_1 \in A_1, a_2 \in A_2 \tag{4.5i}$$

$$x_3^{a_3} \geq 0 \qquad \forall a_3 \in A_3. \tag{4.5j}$$

Proof. By relying on Lemma 4.2, we first introduce the linear programming dual of the linear programming relaxation of Problem (4.4). Thanks to Constraints 4.4b, $y^{a_1,a_2} \in \{0,1\}$ can be relaxed w.l.o.g. into $y^{a_1,a_2} \in \mathbb{Z}^+$ for all $a_1 \in A_1, a_2 \in A_2$. This way, we do not have to introduce a dual variable for each of the constraints $y^{a_1,a_2} \leq 1$ which would be introduced when relaxing $y^{a_1,a_2} \in \{0,1\}$ into $y^{a_1,a_2} \in [0,1]$. Letting α, $\beta_1^{a_1 a_2 a_1'}$, and $\beta_2^{a_1 a_2 a_2'}$ be the dual variables of, respectively, Constraints (4.4b), (4.4c), and (4.4d), the dual reads:

$$\begin{aligned}
\max_{\alpha,\beta_1,\beta_2} \ & \alpha \\
\text{s.t. } & \alpha + \sum_{a_1' \in A_1} \beta_1^{a_1 a_2 a_1'} \sum_{a_3 \in A_3} (U_1^{a_1 a_2 a_3} - U_1^{a_1' a_2 a_3}) x_3^{a_3} + \\
& + \sum_{a_2' \in A_2} \beta_2^{a_1 a_2 a_2'} \sum_{a_3 \in A_3} (U_2^{a_1 a_2 a_3} - U_2^{a_1 a_2' a_3}) x_3^{a_3} \\
& \leq \sum_{a_3 \in A_3} U_3^{a_1 a_2 a_3} x_3^{a_3} & \forall a_1 \in A_1, a_2 \in A_2 \\
& \alpha \text{ free} \\
& \beta_1^{a_1 a_2 a_1'} \geq 0 & \forall a_1, a_1' \in A_1, a_2 \in A_2 \\
& \beta_2^{a_1 a_2 a_2'} \geq 0 & \forall a_1 \in A_1, a_2, a_2' \in A_2.
\end{aligned}$$

A set of optimality conditions for Problem (4.4) can then be derived by simultaneously imposing primal and dual feasibility for the sets of primal and dual variables (by imposing the respective constraints) and equating the objective functions of the two problems. The dual variable α can then be removed by substituting it by the primal objective function, leading to Constraints (4.5e). The result in the claim is obtained after introducing the leader's utility as objective function and then casting the resulting problem as a maximization problem (in which a supremum is sought). □

Since, as shown in Proposition 4.1, the problem of computing a WSPNE in a normal-form SG may only admit a supremum but not a maximum, the same must hold for Problem (4.5) due to its correctness (Theorem 4.4). We formally highlight this property in the following proposition, showing in the proof how this can manifest in terms of the variables of the formulation.

Proposition 4.6. *Problem* (4.5) *may not admit a finite optimal solution.*

Proof. Consider the game introduced in the proof of Proposition 4.1 and let $x_3 = (1-\rho, \rho)$ for $\rho \in [0,1]$. Adopting, for convenience, the notation $(a_1^1, a_2^1) = (1,1)$, $(a_1^1, a_2^2) = (1,2)$, $(a_1^2, a_2^1) = (2,1)$, and $(a_1^2, a_2^2) = (2,2)$, Constraints (4.5e) read:

$$\begin{aligned}
& y^{12}(5+5\rho) + y^{21} \leq -\beta_1^{112}(0.5-\rho) - \beta_2^{112}(-1-\rho) \\
& y^{12}(5+5\rho) + y^{21} \leq 5 + 5\rho - \beta_1^{122}(1+\rho) - \beta_2^{121}(1+\rho) \\
& y^{12}(5+5\rho) + y^{21} \leq 1 - \beta_1^{211}(-0.5+\rho) - \beta_2^{212}(-0.5+\rho) \\
& y^{12}(5+5\rho) + y^{21} - \beta_1^{221}(-1-\rho) - \beta_2^{221}(0.5-\rho).
\end{aligned}$$

Note that the left-hand sides of the four constraints are all equal to the objective function (*i.e.*, to the leader's utility).

Let us consider the case $\rho < 0.5$ for which, as shown in the proof of Proposition 4.1, $(1,2)$ is the unique pure NE in the followers' game. (1,2) is obtained by letting $y^{12} = 1$ and $y^{11} = y^{21} = y^{22} = 0$, for which the left-hand sides of the four constraints become equal to $7.5 - 5\epsilon$. Note that such value converges to the supremum as $\epsilon \to 0$. For this choice of y and letting $\rho = 0.5 - \epsilon$ for $\epsilon \in (0, 0.5]$ (which is equivalent to assuming $\rho < 0.5$), we can rearrange the four constraints as follows:

$$\begin{aligned} \beta_2^{111} &\geq \frac{7.5 - 5\epsilon + \epsilon\beta_1^{112}}{1.5 - \epsilon} \\ (1.5 - \epsilon)\left(\beta_1^{122} + \beta_2^{121}\right) &\leq 0 \\ \beta_1^{211} + \beta_2^{212} &\geq \frac{6.5 - 5\epsilon}{\epsilon} \\ \beta_1^{221} &\geq \frac{7.5 - 5\epsilon + \epsilon\beta_2^{221}}{1.5 - \epsilon}. \end{aligned}$$

The second constraint implies $\beta_1^{122} = \beta_2^{121} = 0$. Letting $\beta_1^{112} = \beta_2^{221} = 0$, which is the least restriction on the first and fourth constraints, we get:

$$\beta_2^{111} \geq \frac{7.5 - 5\epsilon}{1.5 - \epsilon} \text{ and } \beta_1^{211} + \beta_2^{212} \geq \frac{6.5 - 5\epsilon}{\epsilon} \text{ and } \beta_1^{221} \geq \frac{7.5 - 5\epsilon}{1.5 - \epsilon}.$$

As $\epsilon \to 0$, we have a finite lower bound for β_2^{111} and β_1^{221}, but we also have $\beta_1^{211} + \beta_2^{212} \geq \frac{6.5 - 5\epsilon}{\epsilon} \to \infty$, which prevents β_1^{211} and β_2^{212} from taking a finite value. With a similar argument, one can verify that there is no other way of achieving an objective function value approaching 7.5 as, for $\rho \geq 5$, the third constraint in the original system imposes an upper bound on the objective function value of 1. □

4.3.2 A Restricted Single-Level (MILP) Formulation

As state-of-the-art numerical optimization solvers usually rely on the boundedness of their variables when tackling a problem, due to the result in Proposition 4.6 solving the single-level formulation in Problem 4.5 may be numerically impossible.

We consider, here, the option of introducing an upper bound of M on both $\beta_1^{a_1 a_2 a_1'}$ and $\beta_2^{a_1 a_2 a_2'}$, for all $a_1 \in A_1, a_2 \in A_2, a_1' \in A_1, a_2' \in A_2$. Due to the continuity of the objective function, this suffices to obtain a formulation which, although being a restriction of the original one, always admits a maximum (over the reals) as a consequence of Weierstrass' extreme-value theorem. Quite conveniently, this restricted reformulation can be cast as an MILP, as we now show.

Theorem 4.5. *There is an exact MILP reformulation of Problem* (4.5) *for the case where* $\beta_1^{a_1a_2a_1'} \leq M$ *and* $\beta_2^{a_1a_2a_2'} \leq M$ *hold for all* $a_1 \in A_1, a_2 \in A_2, a_1' \in A_1, a_2' \in A_2$*, and a restricted one when the bounds are not valid.*

Proof. After introducing the variable $z^{a_1a_2a_3}$, each bilinear product $y^{a_1a_2}x_3^{a_3}$ in Problem (4.5) can be linearized by substituting $z^{a_1a_2a_3}$ for it and introducing the McCormick envelope constraints (McCormick, 1976), which are sufficient to guarantee $z^{a_1a_2a_3} = y^{a_1a_2}x_3^{a_3}$ if $y^{a_1a_2}$ takes binary values (Al-Khayyal and Falk, 1983). Assuming $\beta_1^{a_1a_2a_1'} \in [0, M]$ for each $a_1 \in A_1, a_2 \in A_2, a_1' \in A_1$, we can restrict ourselves to $\beta_1^{a_1a_2a_1'} \in \{0, M\}$. This is the case also in the dual (reported in the proof of Theorem 4.4). Indeed, the dual problem asks for solving the following problem:

$$\max_{\beta_1,\beta_2 \geq 0} \left\{ \min_{(a_1,a_2) \in A_1 \times A_2} \left\{ \begin{array}{c} \sum_{a_1' \in A_1} \beta_1^{a_1a_2a_1'} \sum_{a_3 \in A_3} (U_1^{a_1a_2a_3} - U_1^{a_1'a_2a_3})x_3^{a_3} + \\ \sum_{a_2' \in A_2} \beta_2^{a_1a_2a_2'} \sum_{a_3 \in A_3} (U_2^{a_1a_2a_3} - U_2^{a_1a_2'a_3})x_3^{a_3} \end{array} \right\} \right\}.$$

The $\min$ operator ranges over functions (one for each pair $(a_1, a_2) \in A_1 \times A_2$) defined on disjoint domains (the β_1, β_2 variables contained in each such function are not contained in any of the other ones). Therefore, we can w.l.o.g. set the value of β_1 and β_2 so that each function be individually maximized. For each $(a_1, a_2) \in A_1 \times A_2$, this is achieved by setting, for each $a_1' \in A_1$ (resp., $a_2' \in A_2$) $\beta_1^{a_1a_2a_1'}$ (resp., $\beta_2^{a_1a_2a_2'}$) to its upper bound M if $\sum_{a_3 \in A_3}(U_1^{a_1a_2a_3} - U_1^{a_1'a_2a_3})x_3^{a_3} \geq 0$ (resp., $\sum_{a_3 \in A_3}(U_2^{a_1a_2a_3} - U_2^{a_1a_2'a_3})x_3^{a_3} \geq 0$), otherwise setting $\beta_1^{a_1a_2a_1'}$ (resp., $\beta_2^{a_1a_2a_2'}$) to its lower bound of 0.

We can, therefore, introduce the variable $p_1^{a_1a_2a_1'} \in \{0, 1\}$, substituting $Mp_1^{a_1a_2a_1'}$ for each occurrence of $\beta_1^{a_1a_2a_1'}$. This way, for each $a_1 \in A_1, a_2 \in A_2, a_1' \in A_1$, the term $\beta_1^{a_1a_2a_1'} \sum_{a_3 \in A_3}(U_1^{a_1a_2a_3} - U_1^{a_1'a_2a_3})x_3^{a_3}$ becomes $M \sum_{a_3 \in A_3}(U_1^{a_1a_2a_3} - U_1^{a_1'a_2a_3})p_1^{a_1a_2a_1'}x_3^{a_3}$. We can, then, introduce the variable $q_1^{a_1a_2a_1'a_3}$ and impose $q_1^{a_1a_2a_1'a_3} = p_1^{a_1a_2a_1'}x_3^{a_3}$ via the McCormick envelope constraints. This way, $M \sum_{a_3 \in A_3}(U_1^{a_1a_2a_3} - U_1^{a_1'a_2a_3})p_1^{a_1a_2a_1'}x_3^{a_3}$ becomes the linear term $M \sum_{a_1' \in A_1} \sum_{a_3 \in A_3}(U_1^{a_1a_2a_3} - U_1^{a_1'a_2a_3})q_1^{a_1a_2a_1'a_3}$. Similar arguments hold for $\beta_2^{a_1a_2a_2'}$, leading to an MILP formulation. □

The impact of bounding $\beta_1^{a_1 a_2 a_1'}$ and $\beta_2^{a_1 a_2 a_2'}$ by M is explained as follows. Assume that those upper bounds are introduced into Problem (4.5). If M is not large enough for the chosen x_3 (remember that, as shown in Proposition 4.6, one may need $M \to \infty$ for x_3 approaching a discontinuity point of the leader's utility function), Constraints (4.5e) may remain active for some $(\hat{a}_1, \hat{a}_2)$ which is not an NE for the chosen x_3. Let (a_1, a_2) be the worst-case NE the followers would play and assume that the right-hand side of Constraint (4.5e) for $(\hat{a}_1, \hat{a}_2)$ is strictly smaller than the utility the leader would obtain if the followers played the NE (a_1, a_2), namely,

$$\sum_{a_3 \in A_3} U_3^{\hat{a}_1 \hat{a}_2 a_3} x_3^{a_3} - \sum_{a_1' \in A_1} \beta_1^{\hat{a}_1 \hat{a}_2 a_1'} \sum_{a_3 \in A_3} (U_1^{\hat{a}_1 \hat{a}_2 a_3} - U_1^{a_1' \hat{a}_2 a_3}) x_3^{a_3} - \\ - \sum_{a_2' \in A_2} \beta_2^{\hat{a}_1 \hat{a}_2 a_2'} \sum_{a_3 \in A_3} (U_2^{\hat{a}_1 \hat{a}_2 a_3} - U_2^{\hat{a}_1 a_2' a_3}) x_3^{a_3} < \sum_{a_3 \in A_3} U_3^{a_1 a_2 a_3} x_3^{a_3}.$$

Letting $y^{a_1 a_2} = 1$, this constraint would be violated (as, with that value of y, the left-hand side of the constraint would be $\sum_{a_3 \in A_3} U_3^{a_1 a_2 a_3} x_3^{a_3}$, which we assumed to be strictly larger than the right-hand side). This forces the choice of a different x_3 for which the upper bound of M on $\beta_1^{a_1 a_2 a_1'}$ and $\beta_2^{a_1 a_2 a_2'}$ is sufficiently large not to cause the same issue with the worst-case NE corresponding to that x_3, thus restricting the set of strategies the leader could play. In spite of this, by solving the MILP reformulation outlined in Theorem 4.5 we are always guaranteed to find optimal (restricted) solutions to it (if M is large enough for the restricted problem to admit feasible solutions). Such solutions correspond to feasible strategies of the leader, guaranteeing her a lower bound on her utility at a WSPNE.

4.4 Exact Algorithms for Computing WSPNEs

In this section, we propose an exact exponential-time algorithm for the computation of a WSPNE, *i.e.*, of $\sup_{x_n \in \Delta_n} f(x_n)$, which does not suffer from the shortcomings of the formulations we introduced in the previous section. In particular, if there is no $x_n \in \Delta_n$ where the leader's utility $f(x_n)$ attains $\sup_{x_n \in \Delta_n} f(x_n)$ (as $f(x_n)$ does not admit a maximum), our algorithm also returns, together with the supremum, a strategy $\hat{x}_n$ which provides the leader with a utility equal to an α-approximation (in the additive sense) of the supremum, namely, a strategy $\hat{x}_n$ satisfying $\sup_{x_n \in \Delta_n} f(x_n) - f(\hat{x}_n) \leq \alpha$ for any additive loss $\alpha > 0$ chosen *a priori*. We first introduce a version of the algorithm based on explicit enumeration, in Subsection 4.4.1, which we then embed into a branch-and-bound scheme in Subsection 4.4.3.

In the remainder of the section, we denote the closure of a set $X \subseteq \Delta_n$ relative to $\text{aff}(\Delta_n)$ by $\overline{X}$, its boundary relative to $\text{aff}(\Delta_n)$ by $\text{bd}(X)$, and its complement relative to Δ_n by X^c. Note that, here, $\text{aff}(\Delta_n)$ denotes the *affine hull* of Δ_n, *i.e.*, the hyperplane in $\mathbb{R}^m$ containing Δ_n.

4.4.1 Enumerative Algorithm

Computing $\sup_{x_n \in \Delta_n} f(x_n)$

The key ingredient of our algorithm is what we call *outcome configurations*. We say that a pair (S^+, S^-) with $S^+ \subseteq A_F$ and $S^- = A_F \setminus S^+$ is an outcome configuration for a given $x_n \in \Delta_n$ if, in the followers' game induced by x_n, all the followers' action profiles $a_{-n} \in S^+$ constitute an NE and all the action profiles $a_{-n} \in S^-$ do not.

For every $a_{-n} \in A_F$, we define $X(a_{-n})$ as the set of all leader's strategies $x_n \in \Delta_n$ for which a_{-n} is an NE in the followers' game induced by x_n. Formally, $X(a_{-n})$ corresponds to the following (closed) polytope:

$$X(a_{-n}) := \left\{ \begin{array}{ll} x_n \in \Delta_n : & \displaystyle\sum_{a_n \in A_n} U_p^{a_{-n},a_n} x_n^{a_n} \geq \sum_{a_n \in A_n} U_p^{a'_{-n},a_n} x_n^{a_n} \\ & \hfill \forall p \in F, a'_p \in A_p \setminus \{a_p\} \\ & \hfill \text{with } a'_{-n} = (a_1, \dots, a_{p-1}, a'_p, a_{p+1}, \dots, a_{n-1}) \end{array} \right\}.$$

For every $a_{-n} \in A_F$, we also introduce the set $X^c(a_{-n})$ of all $x_n \in \Delta_n$ for which a_{-n} is *not* an NE. For that purpose, we first define the following set for each $p \in F$:

$$D_p(a_{-n}, a'_p) := \left\{ \begin{array}{ll} x_n \in \Delta_n : & \displaystyle\sum_{a_n \in A_n} U_p^{a_{-n},a_n} x_n^{a_n} < \sum_{a_n \in A_n} U_p^{a'_{-n},a_n} x_n^{a_n} \\ & \hfill \text{with } a'_{-n} = (a_1, \dots, a_{p-1}, a'_p, a_{p+1}, \dots, a_{n-1}) \end{array} \right\}.$$

$D_p(a_{-n}, a'_p)$, which is a not open nor closed polytope (as it has a missing facet, the one corresponding to its strict inequality), is the set of all values of x_n for which player p would achieve a better utility by deviating from a_{-n} and playing a different action $a'_p \in A_p$. For every $p \in F$, $a_{-n} \in A_F$, and $a'_p \in A_p$, we call the corresponding set $D_p(a_{-n}, a'_p)$ *degenerate* if $U_p^{a_{-n},a_n} = U_p^{a'_{-n},a_n}$ for each $a_n \in A_n$ (recall that $a'_{-n} = (a_1, \dots, a_{p-1}, a'_p, a_{p+1}, \dots, a_{n-1})$). In a degenerate $D_p(a_{-n}, a'_p)$, the constraint $\sum_{a_n \in A_n} U_p^{a_{-n},a_n} x_n^{a_n} < \sum_{a_n \in A_n} U_p^{a'_{-n},a_n} x_n^{a_n}$ reduces to $0 < 0$. Since, in principle, any player could deviate from a_{-n} by playing any action not in a_{-n}, $X^c(a_{-n})$ is the following *disjunctive set*:

$$X^c(a_{-n}) := \bigcup_{p \in F} \left(\bigcup_{a'_p \in A_p \setminus \{a_p\}} D_p(a_{-n}, a'_p) \right).$$

Notice that, since any point in $\mathrm{bd}(X^c(a_{-n}))$ which is not in $\mathrm{bd}(\Delta_n)$ would satisfy, for some a'_p, the (strict, originally) inequality of $D_p(a_{-n}, a'_p)$ as an equation, such point is not in $X^c(a_{-n})$ and, hence, $\mathrm{bd}(X^c(a_{-n})) \cap X^c(a_{-n}) \subseteq \mathrm{bd}(\Delta_n)$. The closure $\overline{X^c(a_{-n})}$ of $X^c(a_{-n})$ is obtained by discarding any degenerate $D_p(a_{-n}, a'_p)$ and by turning the strict constraint in the definition of each nondegenerate $D_p(a_{-n}, a'_p)$ into a nonstrict one. Note that degenerate sets are discarded as turning their strict inequality into a $\leq$ inequality would result in turning the empty set $D_p(a_{-n}, a'_p)$ (whose closure is the empty set) into Δ_n. An illustration of $X(a_{-n})$ and $X^c(a_{-n})$, together with the closure $\overline{X^c(a_{-n})}$ of the latter, is reported in Figure 4.5.

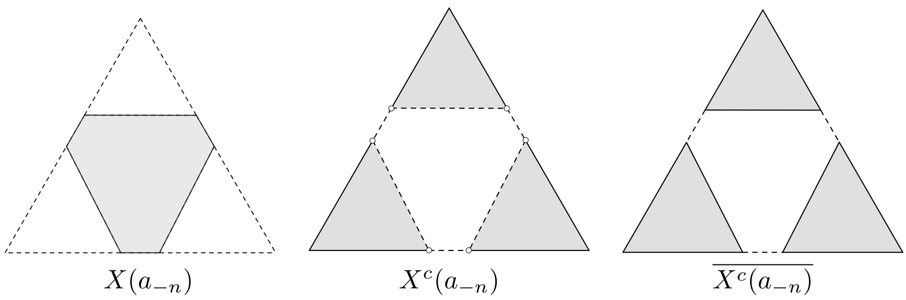


Figure 4.5: *An illustration of $X(a_{-n})$, $X^c(a_{-n})$, and $\overline{X^c(a_{-n})}$ for the case with $m = 3$. The three sets are depicted as subsets (highlighted in gray and continuous lines) of the leader's strategy space Δ_n. Dashed lines and circles indicate parts of Δ_n which are not contained in the sets.*

For every outcome configuration (S^+, S^-), we define the following sets:

$$X(S^+) := \bigcap_{a_{-n} \in S^+} X(a_{-n}) \quad \text{and} \quad X(S^-) := \bigcap_{a_{-n} \in S^-} X^c(a_{-n}).$$

While the former is a closed polytope, the latter is the union of not open nor closed polytopes and, thus, it is not open nor closed itself. Similarly to $X^c(a_{-n})$, $X(S^-)$ satisfies $\mathrm{bd}(X(S^-)) \cap X(S^-) \subseteq \mathrm{bd}(\Delta_n)$. The closure $\overline{X(S^-)}$ of $X(S^-)$ is obtained by taking the closure of each $X^c(a_{-n})$. Hence, $\overline{X(S^-)} = \bigcap_{a_{-n} \in S^-} \overline{X^c(a_{-n})}$.

By leveraging these definitions, we can focus on the set of all leader's strategies which *realize* the outcome configuration (S^+, S^-), namely:

$$X(S^+) \cap X(S^-).$$

As for $X(S^-)$, $X(S^+) \cap X(S^-)$ is not an open nor a closed set. Due to $X(S^+)$ being closed, the only points of $\mathrm{bd}(X(S^+) \cap X(S^-))$ which are not in $X(S^+) \cap X(S^-)$ itself are the very points in $\mathrm{bd}(X(S^-))$ which are not in $X(S^-)$. As a consequence, $\overline{X(S^+) \cap X(S^-)} = X(S^+) \cap \overline{X(S^-)}$.

Let us define the set $P := \{(S^+, S^-) : S^+ \in 2^{A_F} \wedge S^- = 2^{A_F} \setminus S^+\}$, which contains all the outcome configurations of the game. The following theorem highlights the structure of $f(x_n)$, suggesting an iterative way of expressing the problem of computing $\sup_{x_n \in \Delta_n} f(x_n)$. We will rely on it when designing our algorithm.

Theorem 4.6. *Let* $\psi(x_n; S^+) := \min_{a_{-n} \in S^+} \sum_{a_n \in A_n} U_n^{a_{-n}, a_n} x_n^{a_n}$. *It holds:*

$$\sup_{x_n \in \Delta_n} f(x_n) = \max_{\substack{(S^+, S^-) \in P: \\ X(S^+) \cap X(S^-) \neq \varnothing}} \max_{x_n \in X(S^+) \cap \overline{X(S^-)}} \psi(x_n; S^+).$$

Proof. Let Δ'_n be the set of leader's strategies x_n for which there exists a pure NE in the followers' game induced by x_n, namely, $\Delta'_n := \{x_n \in \Delta_n : f(x_n) > -\infty\}$. Since, by definition, $f(x_n) = -\infty$ for any $x_n \notin \Delta'_n$ and the supremum of $f(x_n)$ is finite due to the finiteness of the payoffs (and assuming the followers' game admits at least a pure NE for some $x_n \in \Delta_n$), we can, w.l.o.g., focus on Δ'_n and solve $\sup_{x_n \in \Delta'_n} f(x_n)$. In particular, the collection of the sets $X(S^+) \cap X(S^-) \neq \varnothing$ which are obtained for all $(S^+, S^-) \in P$ forms a partition of Δ'_n. Due to the fact that at any $x_n \in X(S^+) \cap X(S^-)$ the only pure NEs induced by x_n in the followers' game are those in S^+, $f(x_n) = \psi(x_n; S^+)$. Since the supremum of a function defined over a set is equal to the largest of the suprema of that function over the subsets of such set, we have:

$$\sup_{x_n \in \Delta_n} f(x_n) = \max_{\substack{(S^+, S^-) \in P: \\ X(S^+) \cap X(S^-) \neq \varnothing}} \sup_{x_n \in X(S^+) \cap X(S^-)} \psi(x_n; S^+).$$

What remains to show is that the following relationship holds for all $X(S^+) \cap X(S^-) \neq \varnothing$:

$$\sup_{x_n \in X(S^+) \cap X(S^-)} \psi(x_n; S^+) = \max_{x_n \in X(S^+) \cap \overline{X(S^-)}} \psi(x_n; S^+).$$

Since $\psi(x_n; S^+)$ is a continuous function (it is the point-wise minimum of finitely many continuous functions), its supremum over $X(S^+) \cap X(S^-)$ equals its maximum over the closure $\overline{X(S^+) \cap X(S^-)}$ of that set. Hence, the relationship follows due to $\overline{X(S^+) \cap X(S^-)} = X(S^+) \cap \overline{X(S^-)}$. $\square$

In particular, Theorem 4.6 shows that $f(x_n)$ is a piecewise function with a piece for each set $X(S^+) \cap X(S^-)$, each of which corresponding to the (continuous over its domain) piecewise-affine function $\psi(x_n; S^+)$. It follows that the only discontinuities of $f(x_n)$ (due to which $f(x_n)$ may admit a supremum but not a maximum) are those where, in Δ_n, x_n transitions from a set $X(S^+) \cap X(S^-)$ to another one.

We show how to translate the formula in Theorem 4.6 into an algorithm by proving the following theorem:

Theorem 4.7. *There exists a finite, exponential-time algorithm which computes* $\sup_{x_n \in \Delta_n} f(x_n)$ *and, whenever* $\sup_{x_n \in \Delta_n} f(x_n) = \max_{x_n \in \Delta_n} f(x_n)$*, also returns a strategy* x_n^* *with* $f(x_n^*) = \max_{x_n \in \Delta_n} f(x_n)$.

Proof. The algorithm relies on the expression given in Theorem 4.6. All pairs $(S^+, S^-) \in P$ can be constructed by enumeration in time exponential in the size of the instance. [1] In particular, the set P contains $2^{m^{n-1}}$ outcome configurations, each corresponding to a bi-partition of the outcomes of the followers' game into S^+ and S^- (there are m^{n-1} such outcomes, due to having m actions and $n-1$ followers).

For every $p \in F$, let us define the following sets, parametric in $\epsilon \geq 0$:

$$D_p(a_{-n}, a'_p; \epsilon) := \left\{ \begin{array}{c} x_n \in \Delta_n : \sum\limits_{a_n \in A_n} U_p^{a_{-n}, a_n} x_n^{a_n} + \epsilon \leq \sum\limits_{a_n \in A_n} U_p^{a'_{-n}, a_n} x_n^{a_n} \\ \text{with } a'_{-n} = (a_1, \ldots, a_{p-1}, a'_p, a_{p+1}, \ldots, a_{n-1}) \end{array} \right\},$$

$$X^c(a_{-n}; \epsilon) := \bigcup_{p \in F} \left(\bigcup_{a'_p \in A_p \setminus \{a_p\}} D_p(a_{-n}, a'_p; \epsilon) \right),$$

$$X(S^-; \epsilon) := \bigcap_{a_{-n} \in S^-} X^c(a_{-n}; \epsilon).$$

We can verify whether $X(S^+) \cap X(S^-) \neq \varnothing$ by verifying whether there exists some $\epsilon > 0$ such that $X(S^+) \cap X(S^-; \epsilon) \neq \varnothing$. This can be done by

[1] Recall that the size of a game instance is lower bounded by m^n.

solving the following problem and checking if $\epsilon > 0$ in its solution:

$$\begin{array}{ll} \max\limits_{\epsilon, x_n} & \epsilon \\ \text{s.t.} & x_n \in X(S^+) \cap X(S^-; \epsilon) \\ & \epsilon \geq 0 \\ & x_n \in \Delta_n. \end{array} \tag{4.7}$$

Notice that degenerate sets $D_p(a_{-n}, a'_p)$ play no role in Problem (4.7). This is because if $D_p(a_{-n}, a'_p)$ is degenerate, its constraint reduces to $\epsilon \leq 0$ and, thus, any solution to Problem (4.7) with x_n belonging to a degenerate set $D_p(a_{-n}, a'_p)$ would achieve ϵ equal to 0. Thus, $\epsilon > 0$ can be obtained only by choosing x_n not belonging to a degenerate $D_p(a_{-n}, a'_p)$. Problem (4.7) can be cast as an MILP. To see this, observe that each $X^c(a_{-n}; \epsilon)$ can be expressed as an MILP with a binary variable for each term of the disjunction which composes it, namely:

$$\sum_{a_n \in A_n} U_p^{a_{-n}, a_n} x_n^{a_n} + \epsilon \leq \sum_{a_n \in A_n} U_p^{a'_{-n}, a_n} x_n^{a_n} + M_p^{a_{-n}, a'_p} z_p^{a_{-n}, a'_p}$$

$$\forall p \in F, a'_p \in A_p \setminus \{a_p\}, \text{with } a'_{-n} = (a_1, \dots, a'_p, \dots, a_{n-1}) \tag{4.8a}$$

$$\sum_{p \in F} \sum_{a'_p \in A_p \setminus \{a_p\}} 1 - z_p^{a_{-n}, a'_p} = 1 \tag{4.8b}$$

$$z_p^{a_{-n}, a'_p} \in \{0, 1\} \qquad \forall p \in F, a'_p \in A_p \setminus \{a_p\} \tag{4.8c}$$

$$x_n \in \Delta_n \tag{4.8d}$$

$$\epsilon \geq 0. \tag{4.8e}$$

In Constraints (4.8), the constant $M_p^{a_{-n}, a'_p} := \max_{a_n \in A_n} \{U_p^{a_{-n}, a_n} - U_p^{a'_{-n}, a_n}\}$ is key to deactivate any instance of Constraints (4.8a) when the corresponding $z_p^{a_{-n}, a'_p}$ is equal to 1. The set $X(S^-; \epsilon)$ is obtained by simultaneously imposing Constraints (4.8) for all $a_{-n} \in S^-$.

After verifying $X(S^+) \cap \overline{X(S^-)} \neq \varnothing$ by solving Problem (4.7), the value of $\max_{x_n \in X(S^+) \cap \overline{X(S^-)}} \psi(x_n; S^+)$ can be computed in, at most, exponential time by solving the following MILP:

$$\begin{array}{lll} \max\limits_{\eta, x_n} & \eta & \\ \text{s.t.} & \eta \leq \sum\limits_{a_n \in A_n} U_n^{a_{-n}, a_n} x_n^{a_n} & \forall a_{-n} \in S^+ \\ & x_n \in X(S^+) \cap X(S^-; 0) & \\ & \eta \in \mathbb{R} & \\ & x_n \in \Delta_n, & \end{array} \tag{4.9}$$

where the first constraint accounts for the maxmin aspect of the problem. The largest value of η found over all sets $X(S^+) \cap X(S^-)$ for all $(S^+, S^-) \in P$ corresponds to $\sup_{x_n \in \Delta_n} f(x_n)$.

In the algorithm, to verify whether $f(x_n)$ admits $\max_{x_n \in \Delta_n} f(x_n)$ (and to compute it if it does) we solve the following problem (rather than the aforementioned $\max_{x_n \in X(S^+) \cap \overline{X(S^-)}} \psi(x_n; S^+)$):

$$\operatorname*{lex-max}_{\epsilon \geq 0, x_n \in X(S^+) \cap X(S^-; \epsilon)} [\psi(x_n; S^+); \epsilon]. \tag{4.10}$$

This problem calls for a pair (x_n, ϵ) with $x_n \in X(S^+) \cap X(S^-; \epsilon)$ such that, among all pairs which maximize $\psi(x_n; S^+)$, ϵ is as large as possible. This way, in any solution (x_n, ϵ) with $\epsilon > 0$ we have $x_n \in X(S^+) \cap X(S^-)$ (rather than $x_n \in X(S^+) \cap \overline{X(S^-)}$). Since, there, $\psi(x_n; S^+) = f(x_n)$, we conclude that $f(x_n)$ admits a maximum (equal to the value of the supremum) if $\epsilon > 0$, whereas it only admits a supremum if $\epsilon = 0$.

Problem (4.10) can be solved in, at most, exponential time by solving the following lex-MILP:

$$\begin{array}{lll} \max\limits_{\eta, x_n, \epsilon} & [\eta \,; \epsilon] & \\ \text{s.t.} & \eta \leq \sum\limits_{a_n \in A_n} U_n^{a_{-n}, a_n} x_n^{a_n} & \forall a_{-n} \in S^+ \\ & x_n \in X(S^+) \cap X(S^-; \epsilon) & \\ & \eta \in \mathbb{R} & \\ & \epsilon \geq 0 & \\ & x_n \in \Delta_n, & \end{array} \tag{4.11}$$

where η is maximized first, and ϵ second. In practice, it suffices to solve two MILPs in sequence: one in which the first objective function is maximized, and then another one in which the second objective function is maximized after imposing the first objective to be equal to its optimal value. $\square$

Finding an α-Approximate Strategy

For those cases where $f(x_n)$ does not admit a maximum, we look for a strategy $\hat{x}_n$ such that, for any given additive loss $\alpha > 0$, $\sup_{x_n \in \Delta_n} f(x_n) - f(\hat{x}_n) \leq \alpha$, *i.e.*, for an (additively) α-approximate strategy $\hat{x}_n$. Its existence is guaranteed by the following lemma:

Lemma 4.3. *Consider the sets $X \subseteq \mathbb{R}^n$, for some $n \in \mathbb{N}$, and $Y \subseteq \mathbb{R}$, and a function $f : X \to Y$ with $s := \sup_{x \in X} f(x)$ satisfying $s < \infty$. For any $\alpha \in (0, s]$, there exists then an $x \in X : s - f(x) \leq \alpha$.*

Proof. By negating the conclusion, we deduce the existence of some $\alpha \in (0, s]$ such that, for every $x \in X$, $s - f(x) > \alpha$. Then, $f(x) < s - \alpha$ for all $x \in X$. This implies $s = \sup_{x \in X} f(x) \leq s - \alpha < s$: a contradiction. □

After running the algorithm we outlined in the proof of Theorem 4.6 to compute the value of the supremum, an α-approximate strategy $\hat{x}_n$ can be computed *a posteriori* thanks to the following result:

Theorem 4.8. *Assume that $f(x_n)$ does not admit a maximum over Δ_n and that, according to the formula in Theorem 4.6, $s := \sup_{x_n \in \Delta_n} f(x_n)$ is attained at some outcome configuration (S^+, S^-). Then, an α-approximate strategy $\hat{x}_n$ can be computed for any $\alpha > 0$ in at most exponential time by solving the following MILP:*

$$\begin{aligned} \max_{\epsilon, x_n} \quad & \epsilon \\ \text{s.t.} \quad & \sum_{a_n \in A_n} U_n^{a_{-n}, a_n} x_n^{a_n} \geq s - \alpha \qquad \forall a_{-n} \in S^+ \\ & x_n \in X(S^+) \cap X(S^-; \epsilon) \\ & \epsilon \geq 0 \\ & x_n \in \Delta_n . \end{aligned} \tag{4.12}$$

Proof. Let $x_n^* \in X(S^+) \cap \overline{X(S^-)}$ be the strategy where the supremum is attained according to the formula in Theorem 4.6, namely, where $\psi(x_n^*, S^+) = \max_{x_n \in X(S^+) \cap \overline{X(S^-)}} \psi(x_n; S^+) = s$. Problem (4.12) calls for a solution x_n of value at least $s - \alpha$ (thus, for an α-approximate strategy) belonging to $X(S^+) \cap X(S^-; \epsilon)$ with ϵ as large as possible, whose existence is guaranteed by Lemma 4.3. Let $(\hat{x}_n, \hat{\epsilon})$ be an optimal solution to Problem (4.12). If $\hat{\epsilon} > 0$, $\hat{x}_n \in X(S^+) \cap X(S^-)$ (rather than $\hat{x}_n \in X(S^+) \cap \overline{X(S^-)}$). Thus, $f(x_n)$ is continuous at $x_n = \hat{x}_n$, implying $\psi(x_n; S^+) = f(x_n)$. Therefore, by playing $\hat{x}_n$ the leader achieves a utility of at least $s - \alpha$. □

Outline of the Explicit Enumeration Algorithm

The complete enumerative algorithm is detailed in Algorithm 4.1. In the pseudocode, CHECKEMPTYNESS(S^+, S^-) is a subroutine which looks for a value of $\epsilon \geq 0$ which is optimal for Problem (4.7), while SOLVE-LEX-MILP(S^+, S^-) is another subroutine which solves Problem (4.11). Note that Problem (4.7) may be infeasible. If this is the case, we assume that CHECKEMPTYNESS(S^+, S^-) returns $\epsilon = 0$, so that the outcome configuration (S^+, S^-) is discarded. Let us also observe that (in Algorithm 4.1) Problem (4.11) cannot be infeasible, as it is always solved for an outcome configuration (S^+, S^-) whose corresponding Problem (4.7) is feasible. Due to the lexicographic nature of the algorithm, $f(x_n)$ admits a maximum if and only if the algorithm returns a solution with $best.\epsilon^* > 0$. If $best.\epsilon^* = 0$, x_n^* is just a strategy where $\sup_{x_n \in \Delta_n} f(x_n)$ is attained (in the sense of Theorem 4.6). In the latter case, an α-approximate strategy is found by invoking SOLVE-MILP-APPROX$(best.S^+, best.S^-, best_value)$, which solves Problem (4.12) on the configuration $(best.S^+, best.S^-)$ on which the supremum has been found.

Algorithm 4.1 Explicit Enumeration

function EXPLICIT ENUMERATION
    $best \leftarrow nil$
    $best\_val \leftarrow -\infty$
    for all $S^+ \subseteq A_F$ **do**
        $S^- \leftarrow A_F \setminus S^+$
        $(\epsilon, \cdot) \leftarrow$ CHECKEMPTYNESS(S^+, S^-) ▷ Solve MILP Problem (4.7)
        if $\epsilon > 0$ **then**
            $(\eta, \epsilon^*, x_n^*) \leftarrow$ SOLVE-LEX-MILP(S^+, S^-) ▷ Solve lex-MILP Problem (4.11)
            if $\eta > best\_val$ **then**
                $best \leftarrow (S^+, S^-, x_n^*, \epsilon^*)$
                $best\_val \leftarrow \eta$
            end if
        end if
    end for
    if $best.\epsilon^* > 0$ **then**
        $\hat{x}_n \leftarrow best.x_n$
    else
        $\hat{x}_n \leftarrow$ SOLVE-MILP-APPROX$(best.S^+, best.S^-, best_val)$ ▷ Solve MILP Problem (4.12)
    end if
    return $best\_val$, $best.x_n^*$, $\hat{x}_n$
end function

4.4.2 On The Polynomial Representability of WSPNEs

The algorithm that we have presented is based on solving Problem 4.11 a number of times, once per $(S^+, S^-) \in P$. As Problem 4.11 is an MILP, its solutions can be computed by a standard branch-and-bound algorithm based on solving, in an enumeration tree, a set of linear programming relaxations of Problem 4.11 in which the value of (some of) its binary variables is fixed to either 0 or 1. We remark that both Problem 4.11 and its relaxations with fixed binary variables contain a polynomial (in the game size) number of variables and constraints. Moreover, all the coefficients in the problem are polynomially bounded, as they are produced by adding/subtracting the players' payoffs. Since the extreme solution of an LP can be encoded by a number of bits which is also bounded by a polynomial function of the instance size (see Lemma 8.2, page 373, in (Bertsimas and Tsitsiklis, 1997)), we have that any x_n which constitutes a WSPNE can be succinctly encoded by a polynomial number of bits. This observation completes the proof of Theorem 4.2, showing that WSPNE belongs to NP.

4.4.3 Branch-and-Bound Algorithm

As it is clear, computing $\sup_{x_n \in \Delta_n} f(x_n)$ with the enumerative algorithm can be impractical for any game of interesting size, as it requires the explicit enumeration of all the outcome configurations of a game—many of which will, incidentally, yield empty regions $X(S^+) \cap X(S^-)$. A more efficient algorithm, albeit one still running in exponential time in the worst-case, can be designed by relying on a branch-and-bound scheme.

Computing $\sup_{x_n \in \Delta_n} f(x_n)$

Rather than defining $S^- = A_F \setminus S^+$, assume now $S^- \subseteq A_F \setminus S^+$. In this case, we call the corresponding pair (S^+, S^-) a *relaxed outcome configuration*. Starting from any followers' profile $a_{-n} \in A_F$ with $X(a_{-n}) \neq \varnothing$, the algorithm constructs and explores, through a sequence of branching operations, two search trees, whose nodes correspond to relaxed outcome configurations. One tree accounts for the case where a_{-n} is an NE and contains the relaxed outcome configuration $(S^+, S^-) = (\{a_{-n}\}, \varnothing)$ as root node. The other tree accounts for the case where a_{-n} is not an NE, featuring as root node the relaxed outcome configuration $(S^+, S^-) = (\varnothing, \{a_{-n}\})$.

If $S^- \subset A_F \setminus S^+$ (which can often be the case when relaxed outcome configurations are adopted), solving $\max_{x_n \in X(S^+) \cap \overline{X(S^-)}} \psi(x_n; S^+)$ might not give a strategy x_n for which the only pure NEs in the followers' game it induces are those in S^+, even if $x_n \in X(S^+) \cap X(S^-)$ (rather than $x_n \in X(S^+) \cap \overline{X(S^-)}$). This is because, due to $S^+ \cup S^- \subset A_F$, there might be another action profile, say $a'_{-n} \in A_F \setminus (S^+ \cup S^-)$, providing the leader with a utility strictly smaller than that corresponding to all the action profiles in S^+. Since, if this is the case, the followers would respond to x_n by playing a'_{-n} rather than any of the profiles in S^+, $\max_{x_n \in X(S^+) \cap \overline{X(S^-)}} \psi(x_n; S^+)$ could be strictly larger than $\sup_{x_n \in \Delta_n} f(x_n)$, thus not being a valid candidate for the computation of the latter.

In order to detect whether one such a'_{-n} exists, it suffices to carry out a *feasibility check* (on x_n). This corresponds to looking for a pure NE in the followers' game different from those in S^- (which may become NEs on $\mathrm{bd}(X(S^+) \cap X(S^-))$ which minimizes the leader's utility—this can be done by inspection in $O(m^{n-1})$. If the feasibility check returns some $a'_{-n} \notin S^+$, the branch-and-bound tree is expanded by performing a *branching* operation. Two nodes are introduced: a *left node* with (S^+_L, S^-_L) where

$S^+_L = S^+ \cup \{a'_{-n}\}$ and $S^-_L = S^-$ (which accounts for the case where a'_{-n} is a pure NE), and a *right node* with (S^+_R, S^-_R) where $S^+_R = S^+$ and $S^-_R = S^- \cup \{a'_{-n}\}$ (which accounts for the case where a'_{-n} is not a pure NE). If, differently, $a'_{-n} \in S^+$, then $\psi(x_n; S^+)$ represents a valid candidate for the computation of $\sup_{x_n \in \Delta_n} f(x_n)$ and, thus, no further branching is needed (and (S^+, S^-) is a leaf node).

Proposition 4.7. *Solving* $\max_{x_n \in X(S^+) \cap \overline{X(S^-)}} \psi(x_n; S^+)$ *for some relaxed outcome configuration* (S^+, S^-) *gives an* upper bound *on the leader's utility under the assumption that all followers' action profiles in* S^+ *constitute an NE and those in* S^- *do not.*

Proof. Due to (S^+, S^-) being a relaxed outcome configuration, there could be outcomes not in S^+ which are NEs for some $x_n \in X(S^+) \cap \overline{X(S^-)}$. Due to $\psi(x_n; S^+)$ being defined as $\min_{a_{-n} \in S^+} \sum_{a_n \in A_n} U_n^{a_{-n}, a_n} x_n^{a_n}$, ignoring any such NE at any $x_n \in X(S^+) \cap \overline{X(S^-)}$ can only result in the min operator considering fewer outcomes a_{-n}, thus overestimating $\psi(x_n; S^+)$ and, ultimately, $f(x_n)$. Thus, the claim follows. $\square$

As a consequence of Proposition 4.7, optimal values obtained when computing $\max_{x_n \in X(S^+) \cap \overline{X(S^-)}} \psi(x_n; S^+)$ throughout the search tree can be used as bounds as in a standard branch-and-bound method. Given that $\max_{x_n \in X(S^+) \cap \overline{X(S^-)}} \psi(x_n; S^+)$ is not well-defined for nodes with $S^+ = \varnothing$, for them we solve a restriction of Problem (4.1) with constraints imposing that all the followers' action profiles in S^- are not NEs. We employ the following formulation, introduced directly for the lexicographic case:

$$\max_{y, x_n, \epsilon} \left[\sum_{a \in A} U_n^{a_{-n}, a_n} y^{a_{-n}} x_n^{a_n}; \epsilon \right] \tag{4.13a}$$

$$\text{s.t.} \sum_{a_{-n} \in A_F} y^{a_{-n}} = 1 \tag{4.13b}$$

$$y^{a_{-n}} \sum_{a_n \in A_n} (U_p^{a_{-n}, a_n} - U_p^{a'_{-n}, a_n}) x_n^{a_n} \geq 0 \quad \forall p \in F, a_{-n} \in A_F, a'_p \in A_p \setminus \{a_p\} \text{ with } a'_{-n} = (a_1, \ldots, a_{p-1}, a'_p, a_{p+1}, \ldots, a_{n-1}) \tag{4.13c}$$

$$y^{a_{-n}} \in \{0, 1\} \qquad \forall a_{-n} \in A_F \tag{4.13d}$$

$$x_n \in \Delta_n \tag{4.13e}$$

$$x_n \in X(S^-; \epsilon). \tag{4.13f}$$

The problem can be turned into a lex-MILP by linearising each bilinear product $y^{a_{-n}} x_n^{a_n}$ by means of McCormick's envelope constraints and by restating Constraint (4.13f) as done in the MILP Constraints (4.8).

Finding an α-Approximate Strategy

In the context of the branch-and-bound algorithm, an α-approximate strategy $\hat{x}_n$ cannot be found by just relying on the *a posteriori* procedure outlined in Theorem 4.8. This is because when (S^+, S^-) is a relaxed outcome configuration there might be an action profile $a'_{-n} \in A_F \setminus (S^+ \cup S^-)$ (*i.e.*, one not accounted for in the relaxed outcome configuration) which not only is an NE in the followers' game induced by $\hat{x}_n$, but which also provides the leader with a utility strictly smaller than $\psi(\hat{x}_n; S^+)$. Then, the strategy $\hat{x}_n$ found with the procedure of Theorem 4.8 may return a utility arbitrarily smaller than the supremum s and, in particular, smaller than $s - \alpha$.

To cope with this shortcoming and establish whether such an a'_{-n} exists, we first compute $\hat{x}_n$ according to the *a posteriori* procedure of Theorem 4.8 and, then, perform a feasibility check. If we obtain an action profile $a'_{-n} \in S^+$, $\hat{x}_n$ is then an α-approximate strategy and the algorithm halts. If, differently, we obtain some $a'_{-n} \notin S^+$ for which the leader obtains a utility strictly smaller than $\psi(\hat{x}_n; S^+)$, we carry out a new branching operation, creating a left and a right child node in which a'_{-n} is added to, respectively, S^+ and S^-. This procedure is then reapplied on both nodes, recursively, until a strategy $\hat{x}_n$ for which the feasibility check returns an action profile in S^+ is found. Such a strategy is, by construction, α-approximate.

Observe that, due to the correctness of the algorithm for the computation of the supremum, there cannot be at x_n^* an NE a'_{-n} worse than the worst-case one in S^+. If a new outcome a'_{-n} becomes the worst-case NE at $\hat{x}_n$, due to the fact that it is not a worst-case NE at x_n^* there must be a strategy $\tilde{x}_n$ which is a convex combination of x_n^* and $\hat{x}_n$ where either a'_{-n} is not an NE or, if it is, it yields a leader's utility not worse than that obtained with the worst-case NE in S^+. An α-approximate strategy is thus guaranteed to be found on the segment joining $\tilde{x}_n$ and x_n^* by applying Lemma 4.3 with X equal to that segment. Thus, the algorithm is guaranteed to converge.

Outline of the Branch-and-Bound Algorithm

The complete outline of the branch-and-bound algorithm is detailed in Algorithm 4.2. $\mathcal{F}$ is the frontier of the two search trees, containing all nodes which have yet to be explored. INITIALIZE() is a subprocedure which creates the root nodes of the two search trees, while PICK() extracts from $\mathcal{F}$ the

Algorithm 4.2 Branch-and-Bound

function BRANCH-AND-BOUND
$best \leftarrow nil, \quad lb \leftarrow -\infty, \quad ub \leftarrow \infty$
$\mathcal{F} \leftarrow$ INITIALIZE()
while $\mathcal{F} \neq \varnothing$ **do**
$node \leftarrow \mathcal{F}$.PICK()
if $node.ub > lb$ **then**
$a_{-n} \leftarrow$ FEASIBILITYCHECK$(node.x_n^*, node.S^-)$
if $a_{-n} \in node.S^+$ **then**
$best \leftarrow (node.S^+, node.S^-, node.x_n^*, node.\epsilon^*)$
$lb \leftarrow node.ub$
else
$S_L^+ = node.S^+ \cup \{a_{-n}\}$
$\mathcal{F} \leftarrow \mathcal{F} +$ CREATENODE$(S_L^+, node.S^-)$
$S_R^- = node.S^- \cup \{a_{-n}\}$
$\mathcal{F} \leftarrow \mathcal{F} +$ CREATENODE$(node.S^+, S_R^-)$
end if
$ub \leftarrow \max_{node \in \mathcal{F}} \{node.ub\}$
end if
end while
if $best.\epsilon^* > 0$ **then**
$\hat{x}_n \leftarrow best.x_n^*$
else
$\hat{x}_n \leftarrow$ SOLVE-MILP-APPROX$(best.S^+, best.S^-, best_val)$ ▷ Solve MILP Problem (4.12)
$a'_{-n} \leftarrow$ FEASIBILITYCHECK$(\hat{x}_n, best.S^-)$
if $a'_{-n} \notin best.S^+$ **then**
$\hat{x}_n \leftarrow$ BRANCH-AND-BOUND-APPROX$(best.S^+, best.S^-, best.x_n^*)$
end if
end if
return $ub, best.x_n^*, \hat{x}_n$
end function

Algorithm 4.3 CreateNode

function CREATENODE(S^+, S^-)
$(\epsilon, \cdot) \leftarrow$ CHECKEMPTYNESS(S^+, S^-) ▷ Solve MILP Problem (4.7)
if $\epsilon > 0$ **then**
$node \leftarrow$ EMPTYNODE()
$node.S^+ \leftarrow S^+$
$node.S^- \leftarrow S^-$
if $S^+ = \varnothing$ **then**
$(\eta, \epsilon^*, x_n^*) \leftarrow$ SOLVE-LEX-MILP-OPT(S^+, S^-) ▷ Solve lex-MILP Problem (4.13)
else
$(\eta, \epsilon^*, x_n^*) \leftarrow$ SOLVE-LEX-MILP(S^+, S^-) ▷ Solve lex-MILP Problem (4.11)
end if
$node.ub \leftarrow \eta$
$node.x_n^* \leftarrow x_n^*$
$node.\epsilon^* \leftarrow \epsilon^*$
return $node$
end if
return $\varnothing$
end function

next node to be explored. FEASIBILITYCHECK(x_n, S^-) performs the feasibility check operation for the leader's strategy x_n, looking for the worst-case pure NE in the game induced by x_n and ignoring any outcome in S^-. CREATENODE(S^+, S^-) (detailed in Algorithm 4.3) adds a new node to $\mathcal{F}$, also computing its upper bound and the corresponding values of x_n and ϵ. More specifically, CREATENODE(S^+, S^-) performs the same operations of a generic step of the enumerative procedure in Algorithm 4.1 for a given S^+ and S^-, with the only difference that, here, we invoke SOLVE-LEX-MILP-OPT(S^+, S^-) when $S^+ = \varnothing$ to solve Problem (4.13), while we invoke SOLVE-LEX-MILP(S^+, S^-) to solve Problem (4.11) if $S^+ \neq \varnothing$. In the last part of the algorithm, SOLVE-MILP-APPROX$(best.S^+, best.S^-, best_val)$ attempts to compute an α-approximate strategy as done in Algorithm 4.1. If the feasibility check fails for the returned strategy $\hat{x}_n$, then the subprocedure BRANCH-AND-BOUND-APPROX$(best.S^+, best.S^-, best.x_n^*)$ is run, executing a second branch-and-bound method, as described in Subsection 4.4.3, until an α-approximate solution is found.

CHAPTER 5

Computing Stackelberg-Nash Equilibria in Stackelberg Polymatrix Games

In this chapter and the following Chapter 6, we study the problem of computing SPNEs in succinct SGs with a single leader and multiple followers. Specifically, we focus here on polymatrix games. As for normal-form SGs, we restrict ourselves to the case in which the followers are only allowed to play pure strategies, since the unrestricted problem is computationally intractable even with only two followers (Basilico et al., 2016, 2017a).

First, in Section 5.1, we introduce two classes of Stackelberg polymatrix games which allow us to characterize the computational complexity of finding SPNEs in polymatrix games. We also show that these two classes of games are intimately connected with two-player Bayesian Stackelberg games, and, thus, our computational results can be directly extended to the Bayesian setting. Section 5.2 formally defines the computational problems addressed in the rest of the chapter. Finally, Section 5.3 presents our complexity results for Stackelberg polymatrix games, while Section 5.4 provides exact algorithms for finding SPNEs in such games.

5.1 Two Relevant Classes of Stackelberg Polymatrix Games

We introduce two classes of *Stackelberg polymatrix games* (SPGs) which are crucial for providing a complete characterization of the computational complexity of finding SPNEs in polymatrix games. Moreover, these classes of games are of interest in their own, as they are connected with particular Bayesian (normal-form) SGs and security games.

Definition 5.1 (One-Level Tree SPG)**.** *An SPG with a single leader and multiple followers* $\Gamma = (N, \mathcal{A}, U)$ *is a* one-level tree Stackelberg polymatrix game (OLTSPG) *if, for every pair* $p, q \in F$*, it holds* $U_{p,q} = U_{q,p} = 0$.

Intuitively, in an OLTSPG the followers play only against the leader, while they do not play against each other (this is encoded by letting the follower-follower utility matrices be identically equal to zero). Thus, while a general SPG can be graphically depicted as a complete graph whose vertices represent players, the graphical representation of an OLTSPG is a tree with only one level, in which the root node corresponds to the leader and the leaves are associated to the followers. Moreover, we also introduce the following subclass of OLTSPGs:

Definition 5.2 (Star SPG)**.** *An OLTSPG* $\Gamma = (N, \mathcal{A}, U)$ *is a* star Stackelberg polymatrix game (SSPG) *if, for every* $p \in F$*, it holds* $m_p = m$ *and* $U_{n,p} = U_n$*, where* m *is the number of actions available to each follower and* $U_n \in \mathbb{Q}^{m_n \times m}$ *is the leader's utility matrix when playing against a follower.*

An SSPG is a particular OLTSPG in which the leader's payoffs are always the same, regardless of the follower she is playing against.

Notice that OLTSPGs and SSPGs are closely connected with many security scenarios. Indeed, in security games with multiple attackers (*i.e.*, multiple followers), it is often the case that the attackers do not influence each other's payoffs, since they have different preferences over the targets, as it is the case, *e.g.*, when some groups of criminals attack different spots in the same city. This scenario corresponds to the OLTSPG model. Moreover, SSPGs can represent situations in which the payoffs of the defender (*i.e.*, the leader) are not affected by the identity of the attacker who performed the attack, as, from the defender's perspective, it may be more important protecting the targets than knowing who committed the attack.

5.1.1 Connection with Bayesian Normal-Form SGs

Next, we show the key connection between our classes of SPGs and Bayesian SGs with a single leader and a single follower, where the latter can be of different types, while the former has only one type. Specifically, OLTSPGs are equivalent to what we define as Bayesian SGs with *interdependent types*, in which the leader's utility may depend on the follower's type. Furthermore, SPGs are equivalent to what we call Bayesian SGs with *independent types*, where the leader's utility does not depend on the follower's type. Formally:

Definition 5.3 (Bayesian SG with Interdependent Types)**.** *A* Bayesian SG with interdependent types (BSG-INT) *is a Bayesian SG with a single leader and a single follower* $\Gamma = (N, \Theta, \Omega, A, U)$ *in which the leader has a single type, i.e.,* $|\Theta_\ell| = 1$*, and, thus, we can define* $\Theta := \Theta_f$*,* $\Omega \in \Delta(\Theta_f)$*,* $A := \bigtimes_{\theta_f \in \Theta_f} A_{f,\theta_f} \times A_\ell$*, and* $U := \{U_{p,\theta_f}\}_{p \in N, \theta_f \in \Theta_f}$ *with* $U_{p,\theta_f} \in \mathbb{Q}^{|A_{f,\theta_f}| \times |A_\ell|}$*.*

Definition 5.4 (Bayesian SG with Independent Types)**.** *A* Bayesian SG with independent types (BSG-IND) *is an instance* $\Gamma = (N, \Theta, \Omega, A, U)$ *of BSG-INT in which, for every follower's type* $\theta_f \in \Theta_f$*, it holds* $A_{f,\theta_f} = A_f$ *and* $U_{\ell,\theta_f} = U_\ell$*, where* A_f *is the finite set of follower's actions (common to all the follower's types) and* $U_\ell \in \mathbb{Q}^{|A_f| \times |A_\ell|}$ *is the leader's utility matrix (which does not depend on the follower's type).*

The following theorem, whose formal proof follows from (Howson Jr and Rosenthal, 1974), shows the connection between our classes of SPGs and Bayesian SGs with interdependent and independent types.

Theorem 5.1. *There exists a polynomial-time-computable function which maps any instance of BSG-INT (respectively, BSG-IND) to an OLTSPG (respectively, SSPG) and vice versa, where:*

- *each follower's type* $\theta_f \in \Theta_f$ *in the Bayesian SG corresponds to a follower* $p \in F$ *in the SPG, i.e.,* $A_{f,\theta_f} = A_p$ *and* $U_{f,\theta_f} = U_{p,n}$*;*
- *the leader* ℓ *in the Bayesian SG corresponds to the leader* n *in the SPG, i.e.,* $A_\ell = A_n$ *and* $U_{n,p} = U_{\ell,\theta_f}^\top$ *(respectively,* $U_\ell = U_n^\top$*);*

such that, given any mixed strategy profile, the expected utility of each player in the OLTSPG (respectively, SSPG) and the corresponding player or follower's type in the BSG-INT (respectively, BSG-IND) are the same.

We remark that, given the equivalences established in Theorem 5.1, all the computational results (including approximation results) that hold for OLTSPGs can be directly extended to BSGs-INT, while the results regarding SSPGs are also valid for BSGs-IND, and *vice versa*. As a consequence, computing an SSPNE in OLTSPGs is already know to be Poly-APX-complete (Letchford et al., 2009).

5.2 The Problem and Its Formulation

As for normal-form SGs, we let $A_F := \times_{p \in F} A_p$ be the set of followers' action profiles, *i.e.*, all the collections of followers' actions $a_{-n} = (a_1, \dots, a_{n-1})$. Then, we can formally define the problem of computing an SSPNE in an n-player SPGs as the following *bilevel problem*:

$$\begin{aligned} \max_{x_n \in \Delta_n} \max_{a_{-n} \in A_F} \quad & \sum_{p \in F} \sum_{a_n \in A_n} U_{n,p}^{a_n a_p} \, x_n^{a_n} \\ \text{s.t.} \quad a_p \in \quad & \operatorname*{arg\,max}_{a_p \in A_p} \sum_{q \neq p \in N} U_{p,q}^{a_p a_q} + \sum_{a_n \in A_n} U_{p,n}^{a_p a_n} \, x_n^{a_n} \quad \forall p \in F. \end{aligned} \tag{5.1}$$

Notice that the objective in Problem (5.1) is the leader's expected utility when the followers play the actions prescribed by a_{-n} and the leader plays the mixed strategy x_n. Moreover, the $\arg\max$ constraints require that each follower's action a_p be a best response, thus guaranteeing that $a_{-n} = (a_1, \dots, a_{n-1})$ be a pure NE for the given x_n. In particular, the first term in the $\arg\max$ constraint accounts for follower p's utility when playing action a_p against the other followers $q \neq p \in F$, while the second term is the utility obtained by playing against the leader.

Analogously, computing a WSPNE in an n-player SPG amounts to solve the following *bilevel problem*:

$$\begin{aligned} \sup_{x_n \in \Delta_n} \min_{a_{-n} \in A_F} \quad & \sum_{p \in F} \sum_{a_n \in A_n} U_{n,p}^{a_n a_p} \, x_n^{a_n} \\ \text{s.t.} \quad a_p \in \quad & \operatorname*{arg\,max}_{a_p \in A_p} \sum_{q \neq p \in N} U_{p,q}^{a_p a_q} + \sum_{a_n \in A_n} U_{p,n}^{a_p a_n} \, x_n^{a_n} \quad \forall p \in F. \end{aligned} \tag{5.2}$$

We remark that the $\sup$ operator in Problem (5.2) is needed since, as it is the case in general normal-form SGs, the problem may not admit a maximum (see also Proposition 4.1).

Notice that, when focusing on the special case of OLTSPGs, the first term in the $\arg\max$ constraint can be removed, as $U_{p,q} = 0$ for every pair of followers $p \neq q \in F$. As a result, $a_{-n} \in A_F$ is a pure NE in the followers' game resulting from the leader's mixed-strategy commitment $x_n \in \Delta_n$ if and only if each follower is playing a best response to x_n in the two-player normal-form game played against the leader. Thus, since the followers' utilities in such games are liner functions of x_n (specifically, for $p \in F$, it is equal to $\sum_{a_n \in A_n} U_{p,n}^{a_p a_n} \, x_n^{a_n}$), the restriction to followers playing pure strategies is without loss of generality when working with OLTSPGs.

5.3 Computational Complexity

In Subsection 5.3.1, we show that the problem of computing a WSPNE is Poly-APX-complete, even if we restrict the attention to OLTSPGs. [1] Moreover, given the connection between OLTSPGs and BSGs-INT, established in Theorem 5.1, the same result also holds for the latter. Then, in Subsection 5.3.2, we prove that, when the number of players in non-fixed, finding an SSPNE is *not* in Poly-APX unless P = NP, while the same holds for WSPNEs even with a fixed number of players. This is in contrast with the case of n-player normal-form SGs, in which the problem of finding an SSPNE can be solved in polynomial time. We remark that, when the number of players n in an SPG is fixed, an SSPNE can be computed in polynomial time by adapting the algorithm proposed in Theorem 4.1.

5.3.1 Approximating a WSPNE in OLTSPGs

We study the computational complexity of approximating a WSPNE. Our results rely on an approximation-preserving reduction from the Maximum Clique problem, which is Poly-APX-hard (Zuckerman, 2006).

Definition 5.5 (MAX-CLIQUE). *Given an undirected graph $\mathcal{G} := (V, E)$, find a maximum clique of $\mathcal{G}$, i.e., a complete subgraph with maximum size.*

Theorem 5.2. *Computing a WSPNE in SSPGs is Poly-APX-hard.*

Proof. First, we provide a polynomial reduction from MAX-CLIQUE to the problem of finding a WSPNE, reducing an arbitrary instance of MAX-CLIQUE to an SSPG. Then, we prove that the correspondence among instances is correct and the reduction is approximation-preserving. Specifically, we show that the graph $\mathcal{G}$ defined by MAX-CLIQUE admits a clique of size J if and only if the leader gets a utility of J in a WSPNE.

Mapping. Letting $V := \{v_1, \dots, v_{|V|}\}$, for every vertex $v_p \in V$ we introduce a follower p, *i.e.*, $N = F \cup \{n\}$ with $F = \{1, \dots, |V|\}$ and $n = |V| + 1$. Each follower has two actions, *i.e.*, $A_p = \{\chi_0, \chi_1\}$ for all $p \in F$, while the leader has an action per vertex, *i.e.*, $A_n = \{1, \dots, |V|\}$. Players' utilities are defined as follows:

- $U_{p,n}^{\chi_0 a_n} = 1 + |V|^2$ for every $p \in F$ and $a_n \in A_n$ with $(v_p, v_{a_n}) \notin E$;
- $U_{p,n}^{\chi_0 a_n} = 1$ for every $p \in F$ and $a_n \in A_n$ with $(v_p, v_{a_n}) \in E$;
- $U_{p,n}^{\chi_1 a_n} = |V|$ for every $p \in F$ and $a_n \in A_n$ with $a_n = p$;

[1] Computing an SSPNE is already known to be Poly-APX-complete in OLTSPGs (Letchford et al., 2009).

- $U_{p,n}^{\chi_1 a_n} = 0$ for every $p \in F$ and $a_n \in A_n$ with $a_n \neq p$;
- $U_n^{a_n \chi_0} = 0$ and $U_n^{a_n \chi_1} = 1$ for all $a_n \in A_n$.

If. Suppose that the graph $\mathcal{G}$ admits a clique $\mathcal{C} \subseteq V$ of size J. W.l.o.g., we can assume $J < |V|$, since instances with a maximum clique of size $|V|$ can be safely ruled out as we can check if the graph is complete in polynomial time. Consider a leader's mixed strategy such that each action $a_n \in A_n$ with $v_{a_n} \in \mathcal{C}$ is played with probability equal to $\frac{1}{J}$. Then, each follower $p \in F$ with $v_p \in \mathcal{C}$ plays χ_1: in fact, by playing χ_1, they get a utility of $\frac{|V|}{J} > 1$, while by playing χ_0 they can only get 1, since no $a_n \in A_n$ with $(v_p, v_{a_n}) \notin E$ is ever played by the leader, given that $\mathcal{C}$ is a clique. Therefore, the leader's utility is $|\mathcal{C}| = J$ by playing such strategy.

Only if. Suppose that, in a WSPNE of the SSPG, the leader gets a utility of J and, thus, given the definition of the game, there are exactly J followers who play action χ_1. Let $\mathcal{C}$ be the subset of vertices v_p such that follower p plays action χ_1: we prove that $\mathcal{C}$ is a clique. In order for follower p to play χ_1 instead of χ_0, the leader must play action $a_n = p$ with probability greater than or equal to $\frac{1}{|V|}$, otherwise the follower would get a higher utility by playing χ_0. Moreover, the leader cannot play any action $a_n \in A_n$ such that $(v_p, v_{a_n}) \notin E$ with probability at least $\frac{1}{|V|}$, because otherwise the follower would play χ_0, getting a utility greater than or equal to $1 + \frac{1}{|V|} \cdot |V|^2 = 1 + |V|$, which is strictly greater than $|V|$, *i.e.*, the maximum utility the leader can get by playing action χ_1. Thus, the leader must play all the J actions $a_n = p$ such that $v_p \in \mathcal{C}$ with probability at least $\frac{1}{|V|}$, and there is no pair of vertices $v_p, v_{a_n} \in \mathcal{C}$ such that $(v_p, v_{a_n}) \notin E$. So, the vertices in $\mathcal{C}$ are completely connected, and $\mathcal{C}$ is a clique of size J.

The reduction is approximation-preserving since the leader's utility coincides with the cardinality of the clique. Thus, given that MAX-CLIQUE is Poly-APX-hard, the result follows. Notice that the reduction works in both the strong SPNE and the weak SPNE cases, as there is no follower who is indifferent among multiple best responses. □

Next, we provide a polynomial-time approximation algorithm for the WSPNE finding problem guaranteeing an approximation factor polynomial in the game size, thus showing that the problem is in the Poly-APX class.

Theorem 5.3. *Computing a WSPNE in OLTSPGs is in Poly-*APX.

Proof. To prove the result, we provide an algorithm $\mathcal{A}$ working as follows. First, $\mathcal{A}$ makes the leader play a two-player normal-form SG against each follower independently. Let $U_{n,p}^*$ be the utility the leader gets in the game played against follower $p \in F$. Then, the algorithm selects the leader's strategy which is played against a follower p such that $U^*_{n,p}$ is maximum. The utility the leader gets adopting the strategy computed by means of algorithm $\mathcal{A}$ is equal to $U_n^{\text{APX}} \geq \max_{p \in F} U^*_{n,p}$, while the utility she would get in a WSPNE is equal to $U_n^{\text{OPT}} \leq (n-1) \cdot \max_{p \in F} U^*_{n,p}$. Thus, algorithm $\mathcal{A}$ guarantees an approximation factor equal to

$$\frac{U_n^{\text{APX}}}{U_n^{\text{OPT}}} \geq \frac{\max_{p \in F} U^*_{n,p}}{(n-1) \cdot \max_{p \in F} U^*_{n,p}} = \frac{1}{n-1} = \frac{1}{O(n)}.$$

This concludes the proof. □

The next result directly follows from Theorems 5.2 and 5.3.

Theorem 5.4. *Computing a WSPNE in OLTSPGs is Poly-*APX*-complete.*

5.3.2 Inapproximability of SPNEs in General SPGs

In the previous subsection, we analyzed the approximation complexity of finding equilibria in the specific setting of OLTSPGs. Now, we investigate the approximability of computing an S/WSPNE in general SPGs.

We provide two inapproximability results. The first one is for the problem of computing an SSPNE in general SPGs with a non-fixed number of players, and it relies on a reduction from 3-SAT (see Definition 4.3).

Theorem 5.5. *The problem of computing an SSPNE in SPGs is not in Poly-*APX *unless* P = NP.

Proof. We provide a reduction from 3-SAT.

Mapping. Given a 3-SAT instance, *i.e.*, a set $V := \{v_1, \ldots, v_{|V|}\}$ of variables and a set of three-literal clauses $C := \{\phi_1, \ldots, \phi_{|C|}\}$, we build an SPG with $n = |C| + 1$ players, as follows. The set of players is $N = F \cup \{n\}$, where the followers in $F = \{1, \ldots, |C|\}$ are associated with the clauses in C, *i.e.*, follower $p \in F$ corresponds to $\phi_p \in C$. The leader (player n) has an action for each variable in V, plus an additional one, *i.e.*, $A_n = \{1, \ldots, |V|, w\}$ (where $w = |V| + 1$). On the other hand, each follower has only four actions, namely $A_p = \{\chi_0, \chi_1, \chi_2, \chi_3\}$ for every $p \in F$. For any clause $\phi_p \in C$, with $\phi_p = l_1 \vee l_2 \vee l_3$, the payoffs of the corresponding follower p are so defined:

- $U_{p,n}^{\chi_i a_n} = |V| + 1$ for every $i \in \{1, 2, 3\}$ and $a_n \in A_n$ with $v_{a_n} = v(l_i)$ and l_i positive (recall that $v(l_i)$ denotes the variable of l_i);
- $U_{p,n}^{\chi_i a_n} = 0$ for every $i \in \{1, 2, 3\}$ and $a_n \in A_n$ such that $v_{a_n} \neq v(l_i)$ and l_i is positive;

- $U_{p,n}^{\chi_i a_n} = 0$ for every $i \in \{1,2,3\}$ and $a_n \in A_n$ such that $v_{a_n} = v(l_i)$ and l_i is negative;
- $U_{p,n}^{\chi_i a_n} = \frac{|V|+1}{|V|}$ for every $i \in \{1,2,3\}$ and $a_n \in A_n$ such that $v_{a_n} \neq v(l_i)$ and l_i is negative;
- $U_{p,n}^{\chi_0 a_n} = 0$ for every $a_n \in A_n$;
- $U_{p,q}^{a_p a_q} = 0$ for $a_p \in A_p \setminus \{\chi_0\}$ and $a_q \in A_q$, for every $q \in F \setminus \{p\}$;
- $U_{p,q}^{\chi_0 a_q} = \frac{1}{|C|-1}$ for $a_q \in A_q \setminus \{\chi_0\}$, for every $q \in F \setminus \{p\}$;
- $U_{p,q}^{\chi_0 \chi_0} = |V| + 1$ for every $q \in F \setminus \{p\}$;

The leader's payoffs are defined as follows:

- $U_{n,p}^{a_n a_p} = \frac{1}{|C|}$ for every $a_n \in A_n$, $a_p \in A_p \setminus \{\chi_0\}$, and $p \in F$;
- $U_{n,p}^{a_n \chi_0} = \frac{\epsilon}{|C|}$ for every $a_n \in A_n$;

where $\epsilon > 0$ is an arbitrarily small positive constant. In the following, for the ease of presentation and with abuse of notation, we define $U_{p,n}^{a_p x_n}$ as the utility follower $p \in F$ expects to obtain by playing against the leader, when the latter plays strategy $x_n \in \Delta_n$, *i.e.*, $U_{p,n}^{a_p x_n} = \sum_{a_n \in A_n} U_{p,n}^{a_p a_n} x_n^{a_n}$. Furthermore, given a truth assignment to the variables $T : V \to \{0,1\}$, let us define $X(T)$ as the set of leader's strategies $x_n \in \Delta_n$ such that $x_n^{a_n} > \frac{1}{|V|+1}$ if $T(v_{a_n}) = 1$, while $x_n^{a_n} < \frac{1}{|V|+1}$ whenever $T(v_{a_n}) = 0$. Clearly, no matter the truth assignment T, the set $X(T)$ is always non-empty, as one can make the probabilities in the strategy x_n sum up to one by properly choosing x_n^w. On the other hand, given a leader's strategy $x_n \in \Delta_n$, we define T^{x_n} as the truth assignment in which $T^{x_n}(v_{a_n}) = 1$ if $x_n^{a_n} > \frac{1}{|V|+1}$, while $T^{x_n}(v_{a_n}) = 0$ whenever $x_n^{a_n} < \frac{1}{|V|+1}$ (the case $x_n^{a_n} = \frac{1}{|V|+1}$ deserves a different treatment, although the proof can be easily extended to take it into consideration, we omit it for simplicity). W.l.o.g., let us assume $|C| \geq 3$.

Before going into the core of the proof, we prove the following:

Lemma 5.1. *For any leader's strategy $x_n \in \Delta_n$ and follower $p \in F$, there exists an action $a_p \in A_p \setminus \{\chi_0\}$ such that $U_{p,n}^{a_p x_n} > 1$ if and only if ϕ_p evaluates to true under T^{x_n}.*

Proof. Suppose that T^{x_n} makes $\phi_p = l_1 \vee l_2 \vee l_3$ true, and let l_i be one of the literals that evaluate to true in ϕ_p (at least one must exist). Letting $a_n \in A_n$ be such that $v_{a_n} = v(l_i)$, given the definition of T^{x_n}, $x_n^{a_n} > \frac{1}{|V|+1}$ if l_i is positive, whereas $x_n^{a_n} < \frac{1}{|V|+1}$ when l_i is negative. Two cases are possible.

If l_i is positive, then $U_{p,n}^{\chi_i x_n} = x_n^{a_n} \cdot (|V|+1) > 1$, while, if l_i is negative we have $U_{p,n}^{\chi_i x_n} = (1 - x_n^{a_n}) \cdot \frac{|V|+1}{|V|} > 1$. Thus, χ_i is the desired action.

Now, let us prove the other way around. Suppose $a_p \in A_p \setminus \{\chi_0\}$ is such that $U_{p,n}^{a_p x_n} > 1$ and consider the case in which $a_p = \chi_i$ and literal l_i is positive in ϕ_p (similar arguments also hold for the case where l_i is negative). Letting $a_n \in A_n$ be such that $v_{a_n} = v(l_i)$, it easily follows that $x_n^{a_n} \cdot (|V|+1) > 1$, implying that $x_n^{a_n} > \frac{1}{|V|+1}$. Thus, given the definition of T^{x_n}, ϕ_p must evaluate to true. $\square$

***Yes* instance.** Suppose that the given 3-SAT instance has a *yes* answer, *i.e.*, there exists a truth assignment T that satisfies all the clauses. We prove that, if this is the case, then in an SSPNE the leader gets a utility of 1. Consider a leader's strategy $x_n \in X(T)$ and a followers' action profile $a_{-n} \in A_F$ where follower p's action a_p is such that $a_p = \chi_i$ and literal l_i of ϕ_p evaluates to true under truth assignment T. Clearly, the action profile is always well-defined since T satisfies all the clauses. Moreover, when there are many possible choices for action a_p, we assume that the follower plays the one providing her with the maximum utility given x_n. Now, we prove that a_{-n} is a pure NE in the followers' game resulting from the leader's commitment to x_n. Let $p \in F$ be a follower. Clearly, the follower's expected utility in action profile a_{-n} is $U_{p,n}^{a_p x_n}$ since she gets 0 by playing against the other followers. The follower could deviate from a_p in two different ways, either by playing an action corresponding to a different literal in the clause or by playing χ_0. In the first case, the follower cannot get more than what she gets by playing a_p, given the definition of a_p. In the second case, the follower gets $\frac{1}{|C|-1} \cdot (|C|-1) = 1$, which is the utility obtained by playing against the other followers. Observing that T is actually the same as T^{x_n} and using Lemma 5.1, we conclude that $U_{p,n}^{a_p x_n} > 1$ and no follower has an incentive to deviate from a_{-n}, which makes it a pure NE given x_n. Finally, since we are in the strong case, the followers always play a_{-n} since it is the NE maximizing the leader's utility, as, in it, the leader gains $|C| \cdot \frac{1}{|C|} = 1$, which is her maximum payoff. Moreover, for the same reason, the leader's utility in an SSPNE is 1.

***No* instance.** Suppose the 3-SAT instance has a *no* answer, *i.e.*, there is no truth assignment which satisfies all the clauses. First, we prove that the followers' action profile $a_{-n} \in A_F$ in which all the followers play χ_0 is a pure NE, no matter the leader's strategy x_n. In a_{-n}, every follower gets a utility of $(|C|-1) \cdot (|V|+1)$ which does not depend on the leader's strategy. Now, suppose that follower $p \in F$ deviates from a_{-n} by playing some action $a_p \neq \chi_0$, then she would get $U_{p,n}^{a_p x_n} \leq |V|+1$, which is clearly strictly less than $|V| \cdot (|C|-1)$ given the assumption $s \geq 3$. Hence, a_{-n} is always a pure NE in the followers' game and it provides the leader with a utility of $|C| \cdot \frac{\epsilon}{|C|} = \epsilon$. Finally, we show that, for all leader's strategies $x_n \in \Delta_n$, there cannot be other NEs in the followers' game, and, thus, a_{-n} is the unique NE the followers can play. Let us start proving that all the action profiles in which some followers play $a_p \neq \chi_0$ and some others play χ_0 cannot be NEs. Let $p \in F$ be a follower such that $a_p \neq \chi_0$. Clearly, p has an incentive to deviate by playing χ_0 since $U_{p,n}^{a_p x_n} \leq |V| + 1 < \sharp_i \cdot \frac{1}{|C|-1} + \sharp_0 \cdot (|V|+1)$ given that $\sharp_0 \geq 1$, where $\sharp_i$ is the number of followers other than p who are playing $a_p \neq \chi_0$ and $\sharp_0$ is the number of followers playing χ_0. In conclusion, it remains to prove that the followers' action profile in which they all play actions $a_p \neq \chi_0$ cannot be an NE. Let $p \in F$ be a follower such that ϕ_p is false under truth assignment T^{x_n} (she must exist, as, otherwise, the 3-SAT instance would have answer *yes*). Clearly, p has an incentive to deviate playing χ_0 since, using Lemma 5.1, $U_{p,n}^{a_p x_n} < 1 = (|C|-1) \cdot \frac{1}{|C|-1}$. Therefore, in an SSPNE, the leader must get a utility of ϵ.

Contradiction. Suppose there exists a polynomial-time approximation algorithm $\mathcal{A}$ with approximation factor $\frac{1}{\text{poly}(n)}$, where $\text{poly}(n)$ is any polynomial function of n. Moreover, let us fix $\epsilon = \frac{1}{2^n}$ (notice that the polynomiality of the reduction is preserved, as ϵ can still be represented with a number of bits polynomial in n). If the 3-SAT instance has answer *yes*, then $\mathcal{A}$, when applied to the corresponding polymatrix game, must return a solution with value greater than or equal to $\frac{1}{\text{poly}(n)} > \epsilon$. Instead, if the answer is *no*, $\mathcal{A}$ must return a solution of value $\frac{\epsilon}{\text{poly}(n)} < \epsilon$. Thus, the existence of $\mathcal{A}$ would imply that 3-SAT is solvable in polynomial time (the answer is *yes* if and only if the returned solution has value greater than ϵ), which is an absurd, unless P = NP. □

The second inapproximability result we provide (still based on a reduction from 3-SAT) is for the problem of computing a WSPNE in general SPGs, which we prove to be harder than the corresponding problem for SSPNEs, as we show that it is *not* in Poly-APX even when the number of players is fixed. Formally:

Theorem 5.6. *The problem of computing a WSPNE in SPGs is not in Poly-*APX *even when $n = 4$, unless* P = NP.

Proof. We provide a reduction from 3-SAT.

Mapping. Given a 3-SAT instance, *i.e.*, $V := \{v_1, \ldots, v_{|V|}\}$ and $C := \{\phi_1, \ldots, \phi_{|C|}\}$, we build an SPG with $n = 4$ players, as follows. The leader (player 4) has an action for each variable in V, plus an additional one, *i.e.*,

$A_4 = \{1, \ldots, |V|, w\}$ (where $w = |V| + 1$). On the other hand, each follower has 8 actions per clause (each corresponding to a truth assignment to the variables in the clause), plus an additional one, namely $A = A_1 = A_2 = A_3 = \{\phi_{ca} = l_1 l_2 l_3 \mid c \in \{1, \ldots, |C|\}, a \in \{1, \ldots, 8\}\} \cup \{\chi\}$, where $\phi_{ca} = l_1 l_2 l_3$ identifies a truth assignment to the variables in ϕ_c such that $v(l_i)$ is set to true if and only if l_i is a positive literal. For each follower $p \in F$, her payoffs are defined as follows:

- $U_{p,n}^{a_p a_n} = 1$ for all $a_p = \phi_{ca} = l_1 l_2 l_3 \in A_p \setminus \{\chi\}$ and $a_n \in A_n \setminus \{w\}$ with either $v(l_p) = v_{a_n}$ and l_p positive or $v(l_p) \neq v_{a_n}$ and l_p negative;
- $U_{p,n}^{a_p a_n} = 0$ for all $a_p = \phi_{ca} = l_1 l_2 l_3 \in A_p \setminus \{\chi\}$ and $a_n \in A_n \setminus \{w\}$ with either $v(l_p) = v_{a_n}$ and l_p negative or $v(l_p) \neq v_{a_n}$ and l_p positive;
- $U_{p,n}^{a_p w} = 0$ for all $a_p = \phi_{ca} = l_1 l_2 l_3 \in A \setminus \{\chi\}$ if l_p is positive, while $U_{p,n}^{a_p w} = 1$ if it is negative;
- $U_{p,n}^{\chi a_n} = 0$ for all $a_n \in A_n$;
- $U_{p,q}^{a_p a_q} = 0$ for all $a_p \in A \setminus \{\chi\}$, $a_q = a_p \in A \setminus \{\chi\}$, and $q \in F \setminus \{p\}$;
- $U_{p,q}^{a_p a_q} = -1$ for all $a_p \in A \setminus \{\chi\}$, $a_q \neq a_p \in A \setminus \{\chi\}$, and $q \in F \setminus \{p\}$;
- $U_{p,q}^{\chi\chi} = 0$ and $U_{q,p}^{\chi\chi} = 1$ for all $q > p \in F$;
- $U_{p,q}^{\chi a_q} = \frac{1}{2(|V|+1)}$ for all $a_q = \phi_{ca} = l_1 l_2 l_3 \in A_q \setminus \{\chi\}$ with l_p positive, while $U_{p,q}^{\chi a_q} = \frac{|V|}{2(|V|+1)}$ if l_p is negative, for every $q > p \in F$;
- $U_{q,p}^{a_q \chi} = \frac{1}{2(|V|+1)}$ for all $a_q = \phi_{ca} = l_1 l_2 l_3 \in A \setminus \{\chi\}$ with l_q positive, while $U_{q,p}^{a_q \chi} = \frac{|V|}{2(|V|+1)}$ if l_q is negative, for every $q > p \in F$;
- $U_{q,p}^{\chi a_p} = 0$ for all $a_p \in A \setminus \{\chi\}$ and $q > p \in F$;
- $U_{p,q}^{a_p \chi} = 1$ for all $a_p \in A \setminus \{\chi\}$ and $q > p \in F$.

The payoffs for the leader are so defined:

- $U_{n,p}^{a_n a_p} = \frac{1}{3}$ for all $a_n \in A_n$ and $a_p = \phi_{ca} = l_1 l_2 l_3 \in A \setminus \{f\}$ if the truth assignment identified by ϕ_{ca} makes ϕ_c true, while $U_{n,p}^{a_n a_p} = \frac{\epsilon}{3}$ otherwise, where $\epsilon > 0$;
- $U_{n,p}^{a_n \chi} = 1$ for all $a_n \in A_n$.

Before going into the core of the proof, we prove the following:

Lemma 5.2. *For every $\phi_{ca} = l_1 l_2 l_3 \in A \setminus \{\chi\}$, the outcome $(\phi_{ca}, \phi_{ca}, \phi_{ca})$ is a pure NE in the followers' game whenever the leader commits to a strategy $x_n \in \Delta_n$ satisfying the following constraints:*

- $x_n^{a_n} \geq \frac{1}{|V|+1}$ *if $v(l_p) = v_{a_n}$ and l_p is a positive literal, for some $p \in F$;*
- $x_n^{a_n} \leq \frac{1}{|V|+1}$ *if $v(l_p) = v_{a_n}$ and l_p is a negative literal, for some $p \in F$.*

All the outcomes of the followers' game that are not in $\{(\phi_{ca}, \phi_{ca}, \phi_{ca}) \mid a_p \in A \setminus \{\chi\}\}$ cannot be part of a WSPNE.

Proof. Initially, we prove the first part of the statement. Let $x_n \in \Delta_n$ be an arbitrary leader's strategy. Then, for every $\phi_{ca} = l_1 l_2 l_3 \in A \setminus \{\chi\}$, the outcome $(\phi_{ca}, \phi_{ca}, \phi_{ca})$ provides follower p with the following utilities U_p:

- $U_p \coloneqq x_n^{a_n}$ if $v(l_p) = v_{a_n}$ and l_p is positive;
- $U_p \coloneqq 1 - x_n^{a_n}$ if $v(l_p) = v_{a_n}$ and l_p is negative.

Thus, by definition, $(\phi_{ca}, \phi_{ca}, \phi_{ca})$ is an NE if the following conditions hold:

- $U_p \geq \frac{1}{|V|+1}$ for each $p \in F$ such that l_p is positive, as otherwise p would deviate and play χ;
- $U_p \geq \frac{|V|}{|V|+1}$ for each $p \in F$ such that l_p is negative, as otherwise p would deviate and play χ.

Notice that, for every $x_n \in \Delta_n$, there always exists at least one outcome $(\phi_{ca}, \phi_{ca}, \phi_{ca})$ which is an NE in the followers' game. Moreover, all the outcomes (a_1, a_2, a_3) such that $a_1, a_2, a_3 \in A \setminus \{\chi\}$ and $a_p \neq a_q$ for some $p, q \in F$ cannot be NEs since the followers get a negative payoff, while they can obtain a positive utility by deviating to χ. Furthermore, the following outcomes cannot be NEs:

- (χ, χ, χ), as the first follower would deviate by playing any other action, increasing her utility from zero to at least 1;
- (χ, χ, a_3), for any $a_3 \in A \setminus \{\chi\}$, as the third follower would deviate playing action χ, which guarantees her a utility of 2 instead of ≤ 1;
- (χ, a_2, χ), for any $a_2 \in A \setminus \{\chi\}$, as the first follower would deviate to a_2, thus increasing her utility above 1;
- (a_1, χ, χ), for any $a_1 \in A \setminus \{\chi\}$, as the second follower would deviate to a_1, thus increasing her utility above 1;

- (χ, a_2, a_3), for any $a_2 = a_3 \in A \setminus \{\chi\}$, as the second follower would deviate to χ, thus increasing her utility above 1;
- (a_1, χ, a_3), for any $a_1 = a_3 \in A \setminus \{\chi\}$, as the third follower would deviate to χ, thus increasing her utility above 1.

Finally, all the outcomes (a_1, a_2, χ), for all $a_1 = a_2 \in A \setminus \{\chi\}$, are never played by the followers in a WSPNE since, even if they could become NEs for some leader's commitment, they always provide the leader with a utility greater than 1, while, as previously shown, there is always at least another NE which gives her a utility at most equal to 1. □

***Yes* instance.** Suppose that the given 3-SAT instance has a *yes* answer, *i.e.*, there exists a truth assignment which satisfies all the clauses. Then, by Lemma 5.2, there exists a strategy $x_n \in \Delta_n$ such that the worst (for the leader) NE in the followers' game provides her with a utility of 1. Thus, the leader's utility in a WSPNE is 1.

***No* instance.** Let us consider the case in which the instance has a *no* answer. By Lemma 5.2, for every leader commitment $x_n \in \Delta_n$, there exists an NE in the followers' game that gives the leader a utility of ϵ. Thus, the leader's utility in a WSPNE is ϵ.

Contradiction. Now, suppose there exists a polynomial-time approximation algorithm $\mathcal{A}$ with approximation factor $\frac{1}{\text{poly}(n)}$, where $\text{poly}(n)$ is any polynomial function of n. Moreover, let us fix $\epsilon = \frac{1}{2^n}$. If the 3-SAT instance has answer *yes*, then $\mathcal{A}$, when applied to the corresponding SPG, must return a solution with value greater than or equal to $\frac{1}{\text{poly}(n)} > \epsilon$. Instead, if the answer is *no*, $\mathcal{A}$ must return a solution of value $\frac{\epsilon}{\text{poly}(n)} < \epsilon$. Thus, the existence of $\mathcal{A}$ would imply that 3-SAT is solvable in polynomial time, which is an absurd, unless P = NP. □

5.4 Exact Algorithm for Finding a WSPNE in OLTSPGs

We provide an exact algorithm for computing a WSPNE in OLTSPGs whose compute time is exponential in the number of players and polynomial in the number of actions available to the players. The algorithm extends the procedure given in (Von Stengel and Zamir, 2010) to find a supremum of the leader's utility function with two-player normal-form SGs, and it also includes a procedure to compute a strategy that allows the leader to achieve an α-approximation (in additive sense) of the supremum when there is no maximum, for any $\alpha > 0$.

The algorithm is based on the enumeration of all the followers' action profiles, *i.e.*, all the tuples $a_{-n} = (a_1, \dots, a_{n-1}) \in A_F$, and, for each of them, it computes the best strategy the leader can commit to (in the weak case) provided that a_p is a best response for follower p, for every $p \in F$. For the ease of notation, given $a_p \in A_p$ with $p \in F$, let $U_p^{a_p} \in \mathbb{Q}^{|A_n|}$ be a vector whose components are defined as $U_{p,n}^{a_p a_n}$, for every $a_n \in A_n$. The complete algorithm procedure is detailed in Algorithm 5.1, where the parameter α defines the quality of the approximation of the supremum, whenever a maximum does not exist. At each iteration, the algorithm calls two sub-procedures that solve two LP programs. Specifically, SOLVE-EMPTYNESS-CHECK$(\{T_p\}_{p \in F}, a_{-n})$ computes the optimum of the following program:

$$\max_{\epsilon \geq 0, x_n \in \Delta_n} \quad \epsilon$$
$$\text{s.t.} \sum_{a_n \in A_n} U_{p,n}^{a_p a_n} x_n^{a_n} - \sum_{a_n \in A_n} U_{p,n}^{a'_p a_n} x_n^{a_n} \geq \epsilon \quad \forall p \in F, a'_p \in A_p \setminus T_p;$$

while SOLVE-MAX-MIN$(\{T_p\}_{p \in F}, a_{-n})$ solves the following:

$$\begin{aligned}
\max_{x_n \in \Delta_n} \quad & \sum_{p \in F} v_p \\
\text{s.t.} \quad & v_p \leq \sum_{a_n \in A_n} U_{n,p}^{a_n a'_p} x_n^{a_n} && \forall p \in F, a'_p \in T_p \\
& \sum_{a_n \in A_n} U_{p,n}^{a_p a_n} x_n^{a_n} - \sum_{a_n \in A_n} U_{p,n}^{a'_p a_n} x_n^{a_n} - \zeta_{a'_p}^p = 0 && \forall p \in F, a'_p \in A_p \setminus T_p \\
& \zeta_{a'_p}^p \geq 0 && \forall p \in F, a'_p \in A_p \setminus T_p.
\end{aligned}$$

Finally, FIND-APX$(\{T_p\}_{p \in F}, a^*_{-n}, v(a^*_{-n}), \alpha)$ employs the following LP program to find a leader's strategy providing an α-approximation of the supremum:

$$\begin{aligned}
\max_{\epsilon \geq 0, x_n \in \Delta_n} \quad & \epsilon \\
\text{s.t.} \quad & \sum_{p \in F} v_p \geq v(a^*_{-n}) - \alpha \\
& v_p \leq \sum_{a_n \in A_n} U_{n,p}^{a_n a'_p} x_n^{a_n} && \forall p \in F, a'_p \in T_p \\
& \sum_{a_n \in A_n} U_{p,n}^{a_p, a_n} x_n^{a_n} - \sum_{a_n \in A_n} U_{p,n}^{a'_p a_n} x_n^{a_n} \geq \epsilon && \forall p \in F, a'_p \in A_p \setminus T_p.
\end{aligned}$$

Algorithm 5.1 Exact-WSPNE

function EXACT-WSPNE(α)
  for all $a_{-n} = (a_1, \ldots, a_{n-1}) \in A_F$ **do**
    for all $p \in F$ **do**
      $T_p \leftarrow \left\{ a'_p \in A_p \mid U_p^{a_p} = U_p^{a'_p} \right\}$
    end for
    $\epsilon \leftarrow$ SOLVE-EMPTYNESS-CHECK($\{T_p\}_{p \in F}, a_{-n}$)
    if $\epsilon > 0$ **then**
      $\left(v(a_{-n}), x_n(a_{-n}), \zeta^p_{a'_p} \right) \leftarrow$ SOLVE-MAX-MIN($\{T_p\}_{p \in F}, a_{-n}$)
      $\beta(a_{-n}) \leftarrow |\{\zeta^p_{a'_p} \mid \zeta^p_{a'_p} = 0\}| > 0$
    end if
  end for
  $a^*_{-n} \leftarrow \arg\max_{a_{-n} \in A_F} v(a_{-n})$
  if $\beta(a^*_{-n})$ **then**
    return FIND-APX($\{T_p\}_{p \in F}, a^*_{-n}, v(a^*_{-n}), \alpha$)
  end if
  return $x_n(a^*_{-n})$
end function

The following theorem shows that Algorithm 5.1 is correct.

Theorem 5.7. *Given an OLTSPG, Algorithm 5.1 finds a WSPNE, and, whenever the leader's utility function does not admit a maximum, it returns an α-approximation of the supremum.*

Proof. Before proving the statement, we introduce some useful notation. Given $a_p \in A_p$, with $p \in F$, let $X(a_p) \subseteq \Delta_n$ be the set of those strategies $x_n \in \Delta_n$ such that follower p's best response to x_n is the action a_p, *i.e.*, $X(a_p) := \{x_n \in \Delta_n \mid a_p \in \arg\max_{a'_p \in A_p} \sum_{a_n \in A_n} U_{p,n}^{a'_p a_n} x_n^{a_n}\}$. Given a followers' action profile $a_{-n} = (a_1, \ldots, a_{n-1}) \in A_F$, let $X(a_{-n}) := \bigcap_{p \in F} X(a_p)$. We denote with $X^o(\cdot)$ the interior of $X(\cdot)$ relative to Δ_n, and we call $X(\cdot)$ full-dimensional if $X^o(\cdot)$ is non-empty.

In order to prove the result, we define the search problem of computing a WSPNE in OLTSPGs, as follows:

$$\max_{a_{-n} \in A_D} \max_{x_n \in X(a_{-n})} \min_{a'_{-n} \in A_F \,:\, U_p^{a_p} = U_p^{a'_p}} \sum_{p \in F} \sum_{a_n \in A_n} U_{n,p}^{a_n a'_p} x_n^{a_n}, \tag{5.3}$$

where $A_D = \{a_{-n} \in A_F \mid X(a_{-n}) \text{ is full-dimensional } \}$.

First, using a simple inductive argument, we derive a new definition for Δ_n, which is as follows:

$$\Delta_n := \bigcup_{a_{-n} \in A_D} X(a_{-n}) \tag{5.4}$$

Let us start noticing that $\Delta_n = \bigcup_{a_{-n} \in A_F} X(a_{-n})$. Then, take $a'_{-n} \in A_F \setminus A_D$ and define $\mathcal{S} \coloneqq \Delta_n \setminus \bigcup_{a_{-n} \in A_F \setminus \{a'_{-n}\}} X(a_{-n})$. We observe that $\mathcal{S}$ is a subset of $X(a'_{-n})$, and, thus, it is also a subset of $X^o(a'_{-n})$, which is empty since $a'_{-n} \notin A_D$, so $\mathcal{S}$ is empty. Therefore, we can write $\Delta_n = \bigcup_{a_{-n} \in A_F \setminus \{a'_{-n}\}} X(a_{-n})$, which we use as new definition for Δ_n. Iterating in this manner until all the elements in $A_F \setminus A_D$ have been considered, we eventually obtain the result.

Second, we recall a result from Von Stengel and Zamir (2010), *i.e.*, for every $a_p, a'_p \in A_p$, it holds:

$$x_n \in X^o(a_p) \wedge x_n \in X(a'_p) \implies U_p^{a_p} = U_p^{a'_p}. \tag{5.5}$$

We are now ready to prove Equation (5.3), as follows:

$$\begin{aligned} V &\coloneqq \sup_{x_n \in \Delta_n} \min_{a'_{-n} \in A_F : x_n \in X(a'_{-n})} \sum_{p \in F} \sum_{a_n \in A_n} U_{n,p}^{a_n a'_p} x_n^{a_n} \\ &= \max_{a_{-n} \in A_D} \sup_{x_n \in X(a_{-n})} \min_{a'_{-n} \in A_F : x_n \in X(a'_{-n})} \sum_{p \in F} \sum_{a_n \in A_n} U_{n,p}^{a_n a'_p} x_n^{a_n}, \end{aligned}$$

where the first equality directly follows from the definition of the problem, while the second one is obtained rewriting Δ_n as given by (5.4). Restricting $X(a_{-n})$ to $X^o(a_{-n})$ and using (5.5), we obtain:

$$\begin{aligned} V &\geq \max_{a_{-n} \in A_D} \sup_{x_n \in X^o(a_{-n})} \min_{a'_{-n} \in A_F : x_n \in X(a'_{-n})} \sum_{p \in F} \sum_{a_n \in A_n} U_{n,p}^{a_n a'_p} x_n^{a_n} \\ &= \max_{a_{-n} \in A_D} \sup_{x_n \in X^o(a_{-n})} \min_{a'_{-n} \in A_F : U_p^{a_p} = U_p^{a'_p}} \sum_{p \in F} \sum_{a_n \in A_n} U_{n,p}^{a_n a'_p} x_n^{a_n} \\ &= \max_{a_{-n} \in A_D} \sup_{x_n \in X(a_{-n})} \min_{a'_{-n} \in A_F : U_p^{a_p} = U_p^{a'_p}} \sum_{p \in F} \sum_{a_n \in A_n} U_{n,p}^{a_n a'_p} x_n^{a_n} \\ &\geq \max_{a_{-n} \in A_D} \sup_{x_n \in X(a_{-n})} \min_{a'_{-n} \in A_F : x_n \in X(a'_{-n})} \sum_{p \in F} \sum_{a_n \in A_n} U_{n,p}^{a_n a'_p} x_n^{a_n} = V, \end{aligned}$$

where the last equality holds since the minimum is taken over a finite set of linear functions and it is continuous, while the last inequality comes from the fact that the minimum is taken over a larger set of elements. Hence, all the inequalities must hold as equalities, which proves Equation (5.3).

The algorithm exploits Equation (5.3) to compute a WSPNE. Notice that, if $X(a_{-n})$ is not full-dimensional, then by calling the sub-procedure SOLVE-EMPTYNESS-CHECK$(\{T_p\}_{p \in F}, a_{-n})$ we get zero, as, if there is no strategy $x_n \in X^o(a_{-n})$, then there is always at least one inequality in the LP program which can be satisfied only by setting $\epsilon = 0$. The algorithm iterates over all the followers' action profiles in A_D, as every $a_{-n} \in A_F \setminus A_D$ is discarded since $\epsilon = 0$ for such a_{-n}. Then, for each remaining action profile, the algorithm solves the max-min expression on the right of Equation (5.3), which can be done with the LP program solved by SOLVE-MAX-MIN$(\{T_p\}_{p \in F}, a_{-n})$. Finally, the algorithm selects the followers' action profile with the highest max-min expression value.

In conclusion, note that, given some $a_{-n} \in A_D$, $\beta(a_{-n})$ is true if and only if $x_n(a_{-n})$ is such that there is at least one follower p who has a best response a'_p that is not in T_p, *i.e.*, at least one variable $\zeta^p_{a'_p}$ is zero. Thus, if $\beta(a^*_{-n})$ is true, the leader's utility function does not admit a maximum, since for $x_n^{a^*_{-n}}$ there is some follower who can play a best response which is worse than the one played in a^*_{-n} in terms of leader's utility. If that is the case, FIND-APX$(\{T_p\}_{p \in F}, a^*_{-n}, v^{a^*_{-n}}, \alpha)$ finds an α-approximation of the supremum $v^{a^*_{-n}}$ by looking for a strategy $x_n \in X^o(a^*_{-n})$, with the additional constraints imposing that the leader's utility (in the weak case) does not fall below $v^{a^*_{-n}} - \alpha$. Such approximation always exists since $X^o(a^*_{-n})$ is non-empty and the leader's utility is the minimum of a finite set of affine functions. □

Even though, as described next, one can adopt the algorithm proposed in (Von Stengel and Zamir, 2010) to find a WSPNE in an OLTSPG, this would result in a procedure that is more inefficient than Algorithm 5.1. Indeed, one should first transform an OLTSPG into a BSG-INT, by means of the mapping provided in Theorem 5.1, and, then, cast the resulting game in normal form. However, this would require the solution of an exponential number of LP programs, each with an exponential number of constraints, since the number of actions of the resulting normal-form SG is exponential in the size of the original game. Conversely, Algorithm 5.1 exploits the separability of players' utilities, avoiding the explicit construction of the normal form before the execution of the algorithm. As a result, our algorithm still requires the solution of an exponential number of LP programs, but each with a polynomial number of constraints. Notice that avoiding the explicit construction of the normal form also allows the execution of Algorithm 5.1 in an anytime fashion, stopping the algorithm whenever the available time is expired.

CHAPTER 6

Computing Stackelberg-Nash Equilibria in Stackelberg Congestion Games

In this chapter, we continue the study of the problem of computing SPNEs in succinct SGs with a single leader and multiple followers, focusing on congestion games. We remark that, in these games, restricting the followers to pure strategies is without loss of generality, since the followers' game resulting from a leader's mixed-strategy commitment is still a congestion game, and, thus, it admits at least one pure NE (Rosenthal, 1973) which can be reached by the followers in an iterative fashion by playing a best-response dynamics (Monderer and Shapley, 1996).

Initially, in Section 6.1, we formally define the models and the computational problems we study, also pointing out some application examples. Then, we extensively address Stackelberg singleton congestion games, identifying two features which allow for a characterization of hard and easy game instances. The first feature concerns the *symmetry* of the players (whether they have the same action spaces or not), while the second feature is about the shape of the cost functions (*monotonically increasing* or not). In particular, Section 6.2 presents computational complexity results that completely characterize hard game instances, while Section 6.3 provides polynomial-time algorithms for easy instances. Then, we switch the attention to more general settings, specifically, Section 6.4 studies games with non-singleton actions, while Section 6.5 extends some of the positive results for symmetric Stackelberg singleton congestion games to the case in which the players are split into a finite number of *classes*. In conclusion, Section 6.6 shows how to formulate the problem of finding SSPNEs in congestion games (in different game classes) as an MILP.

6.1 The Model and Its Applications

First, we introduce some additional notation useful in the following and provide the formal definitions of Stackelberg congestion games and the related equilibrium-computation problems studied in this chapter. We conclude the section with some application examples showing how the games we consider map to real-world problems.

6.1.1 Stackelberg Congestion Games and Their Variants

As for other game models, in the Stackelberg counterpart of a congestion game, we define $N = F \cup \{n\}$, where F is the set of followers and player n is the leader. Moreover, in order to capture as much real-world settings as possible, we assume that the leader's costs may differ from the followers'. Formally, we study the following class of games:

Definition 6.1 (Stackelberg Congestion Game)**.** *A* Stackelberg congestion game (SCG) *is a tuple* $(N, R, \mathcal{A}, c_n, c_F)$*, where:*

- *N, R, and $\mathcal{A}$ are defined as in a congestion game, with $N = F \cup \{n\}$;*
- *$c_n = \{c_{i,n}\}_{i \in R}$ and $c_F = \{c_{i,F}\}_{i \in R}$ are finite sets of, respectively, leader's and followers' cost functions, with $c_{i,n}, c_{i,F} : \mathbb{N} \to \mathbb{Q}$ being the costs of resource i as a function of its congestion for, respectively, the leader and the followers.*

As usual, we assume $c_{i,n}(0) = c_{i,F}(0) = 0$ for every $i \in R$. We call the players' cost functions *(weakly) monotonic* if, for every resource $i \in R$, $c_{i,n}(y) \leq c_{i,n}(y+1)$ and $c_{i,F}(y) \leq c_{i,F}(y+1)$ for all $y \in \mathbb{N}$, and *strictly monotonic* if all the inequalities are strict. Whenever the inequalities are not satisfied, we say that the players' cost functions are *non-monotonic*.

In this chapter, we collectively denote by $x = (x_n, a_{-n})$ a strategy profile in which the leader plays a (possibly) mixed strategy $x_n \in \Delta_n$ and the followers play as prescribed by the action profile $a_{-n} := (a_1, \dots, a_{n-1}) \in A_F$, where $A_F := \bigtimes_{p \in F} A_p$ is the set of followers' action profiles.

Let $a_{-n} = (a_1, \dots, a_{n-1}) \in A_F$ be a followers' action profile. Overloading notation, we denote by $\nu^i_{a_{-n}} := |\{p \in F \mid a_p = i\}|$ the number of followers selecting resource $i \in R$ in a_{-n}. This quantity is equal to the resource congestion caused by the followers' presence only. Moreover, we call *followers' configuration* (induced by the action profile a_{-n}) the vector $\nu_{a_{-n}} \in \mathbb{N}^r$ whose i-th component is $\nu^i_{a_{-n}}$ for all $i \in R$.

For any leader's strategy $x_n \in \Delta_n$, we define the followers' expected cost for resource $i \in R$ given x_n as the function $c^{x_n}_{i,F} : \mathbb{N} \to \mathbb{Q}$. Specifically, $c^{x_n}_{i,F}$ is a function of the number $y \in \mathbb{N}$ of followers selecting resource i, *i.e.*:

$$c^{x_n}_{i,F}(y) := x^i_n c_{i,F}(y+1) + (1 - x^i_n) c_{i,F}(y).$$

Note that, given a leader's strategy x_n and a followers' congestion y, all the followers who select resource $i \in R$ experience a congestion that may (with probability x^i_n) or may not (with probability $1 - x^i_n$) be incremented by one w.r.t. y, depending on whether the leader chooses resource i or not. Given the strategy profile $x = (x_n, a_{-n})$, the costs experienced by each follower $p \in F$ and the leader are, respectively,

$$c^x_p := \sum_{i \in a_p} c^{x_n}_{i,F}(\nu^i_{a_{-n}}) \quad \text{and} \quad c^x_n := \sum_{a_n \in A_n} x^{a_n}_n \sum_{i \in a_n} c_{i,n}(\nu^i_{a_{-n}} + 1).$$

Note that, whenever the leader selects resource $i \in R$ (which happens with probability x^i_n), her costs depends on the followers' congestion $\nu^i_{a_{-n}}$ plus an additional unit of congestion which due to her choosing that resource.

Different subclasses of SCGs can be defined by making additional assumptions on their elements. One possibility is to restrict the structure of the players' action sets A_p. Along this direction, we address, in the first part of this chapter, games in which players' actions are required to be *singletons*, *i.e.*, $|a_p| = 1$ for every $p \in N$ and $a_p \in A_p$. Thus, when studying such games, we identify actions with resources. Formally:

Definition 6.2 (Stackelberg Singleton Congestion Game)**.** *A* Stackelberg singleton congestion game (SSCG) *is an SCG* $\Gamma = (N, R, \mathcal{A}, c_n, c_F)$ *in which we define* $A_p \subseteq R$ *for all* $p \in N$.

We recall that, when working with singleton games, we use $x_n \in \Delta_n$ as if it were directly defined over resources in A_n, with x^i_n being the probability of playing resource $i \in A_n$. Moreover, given a strategy profile $x = (x_n, a_{-n})$, the following holds:

$$c^x_p = c^{x_n}_{a_p,F}(\nu^{a_p}_{a_{-n}}) \quad \text{and} \quad c^x_n = \sum_{i \in A_n} x^i_n c_{i,n}(\nu^i_{a_{-n}} + 1).$$

Another possibility is to consider different kinds of players' structures. We focus on two notable cases. In the first one, all players share the same set of actions, *i.e.*, $A_p := A \subseteq 2^R$ for all $p \in N$. We refer to these games as *symmetric*. Instead, in the second case, there exists a finite set $\mathcal{T} := \{1, \dots, T\}$ of *followers' classes*, with followers of the same class sharing the same set of actions. We say that these games are $\mathcal{T}$*-class*. Specifically, in a $\mathcal{T}$-class SCG, we can partition the followers into T disjoint sets, *i.e.*, $F := \bigcup_{t \in \mathcal{T}} F_t$, so that, for each $t \in \mathcal{T}$, $A_p := A_t \subseteq 2^R$ for all $p \in F_t$. We also let $n_t := |F_t|$ be the number of followers of class $t \in \mathcal{T}$. When studying these games, given a followers' action profile $a_{-n} \in A_F$, we let $\nu^{t,i}_{a_{-n}} := |\{p \in F_t \mid i \in a_p\}|$ be the number of followers of class $t \in \mathcal{T}$ selecting resource $i \in R$ in a_{-n}. Moreover, we define the *followers' configuration of class* t induced by a_{-n} as the vector $\nu^t_{a_{-n}} \in \mathbb{N}^r$ whose i-th component is $\nu^{t,i}_{a_{-n}}$. Let us remark that symmetric SCGs are a special case of $\mathcal{T}$-class SCGs with only one class, *i.e.*, $\mathcal{T} = \{1\}$, and leader's action set equal to the followers', *i.e.*, $A_n = A_1$.

Observe that $\mathcal{T}$-class SSCGs, and, in particular, *symmetric Stackelberg singleton congestion games* (SSSCGs), can be fully analyzed using followers' configurations, rather than action profiles. This is because only the number of followers of each class selecting each resource is significant, and, thus, a followers' action profile $a_{-n} \in A_F$ can be equivalently represented with the followers' configurations $\{\nu^t_{a_{-n}}\}_{t \in \mathcal{T}}$. Thus, we can directly use the vector $\nu^t \in \mathbb{N}^r$ with $\sum_{i \in R} \nu^{t,i} = n_t$ to denote a followers' configuration of class $t \in \mathcal{T}$. Moreover, for notational convenience, given $\{\nu^t\}_{t \in \mathcal{T}}$, we let $\nu \in \mathbb{N}^r$ be such that $\nu^i := \sum_{t \in \mathcal{T}} \nu^{t,i}$ for $i \in R$.

6.1.2 Computing SPNEs in SCGs

We study the computational problem of finding SPNEs in SCGs. As usual, we address two versions of the problem: the *strong* one (*i.e.*, finding an SSPNE), where we assume that followers play a pure NE minimizing the leader's cost, and the *weak* one (*i.e.*, finding a WSPNE), which assumes that they play a pure NE maximizing the leader's cost.

In an SCG, after observing a leader's commitment $x_n \in \Delta_n$, the followers play a congestion game where the resource costs are specified by the functions $c^{x_n}_{i,F}$, for $i \in R$. Given a strategy profile $x = (x_n, a_{-n})$, the followers' action profile a_{-n} is a pure NE in the followers' game for x_n if, for every follower $p \in F$ and action $a'_p \in A_p$, it holds:

$$c^x_p \leq c^{x'}_p, \text{ where } x' = (x_n, a'_{-n}) \text{ and } a'_{-n} = (a_1, \dots, a'_p, \dots, a_{n-1}).$$

Moreover, given $x_n \in \Delta_n$, overloading notation, we let $\mathcal{E}(x_n)$ be the set of pure NEs in the followers' game for x_n.

In the specific setting of SSCGs, given $x = (x_n, a_{-n})$, the followers' action profile a_{-n} is a pure NE for x_n if the following holds:

$$c^{x_n}_{a_p,F}(\nu^{a_p}_{a_{-n}}) \leq c^{x_n}_{a'_p,F}(\nu^{a'_p}_{a_{-n}} + 1) \quad \forall p \in F, a'_p \in A_p.$$

Furthermore, in $\mathcal{T}$-class SSCGs, given $\sigma = (x_n, \{\nu^t\}_{t\in\mathcal{T}})$, the followers' configurations $\{\nu^t\}_{t\in\mathcal{T}}$ define a pure NE for x_n if the following holds:

$$c^{x_n}_{i,F}(\nu^i) \leq c^{x_n}_{j,F}(\nu^j + 1) \quad \forall t \in \mathcal{T}, i \in A_t : \nu^{t,i} > 0, j \in A_t.$$

Given the previous definitions, finding an SSPNE and a WSPNE amounts to solving two *bilevel problems*, respectively,

$$\min_{x_n \in \Delta_n} \min_{a_{-n} \in \mathcal{E}(x_n)} c_n^{(x_n, a_{-n})} \quad \text{and} \quad \inf_{x_n \in \Delta_n} \max_{a_{-n} \in \mathcal{E}(x_n)} c_n^{(x_n, a_{-n})}.$$

Clearly, an SSPNE always exists in SCGs (Von Stengel and Zamir, 2010), and, since the same objective function is minimized in both levels, the problem can be equivalently rewritten as the following single-level problem:

$$\min_{x_n \in \Delta_n,\, a_{-n} \in \mathcal{E}(x_n)} c_n^{(x_n, a_{-n})}.$$

Moreover, let us observe that the problem of computing a WSPNE calls for an $\inf$ rather than a $\min$ since, in general, the problem may not admit a minimum (but only an infimum). When the problem does admit a minimum, a WSPNE does not exist (Von Stengel and Zamir, 2010). The following proposition shows that this happens even in the basic case of SSSCGs.

Proposition 6.1. *There are SSSCGs in which a WSPNE does not exist.*

Proof. Consider the following instance of an SSSCGs (whose cost functions are reported in the table below), where $|F| = 1$ and $R = \{r_1, r_2\}$. [1]

y	$c_{r_1,n}$	$c_{r_1,F}$	$c_{r_2,n}$	$c_{r_2,F}$
1	2	1	2	1
2	0	2	2	2

Clearly, the single follower selects r_1 if $x_n^{r_1} < \frac{1}{2}$, she chooses r_2 if $x_n^{r_1} > \frac{1}{2}$, and she is indifferent between r_1 and r_2 if $x_n^{r_1} = \frac{1}{2}$. Thus, the leader's cost

[1] In the rest of the chapter, for the ease of presentation and to avoid cumbersome notation, instead of denoting players and resources with integers, *i.e*, $N \coloneqq \{1, \ldots, n\}$ and $R \coloneqq \{1, \ldots, r\}$, we assign labels to them (such as r_1 and r_2 in the proof of Proposition 6.1). Clearly, we can always map such labels to integers.

is $2-2x_n^{r_1}$ if $x_n^{r_1} < \frac{1}{2}$, while it is 2 if $x_n^{r_1} \geq \frac{1}{2}$, since, in the pessimistic case, the follower selects r_2 rather than r_1 when $x_n^{r_1} = \frac{1}{2}$. As a result, the problem of computing a WSPNE achieves an infimum with value 1 at $x_n^{r_1} = \frac{1}{2}$, but it does not admit a minimum. Thus, the game does not admit a WSPNE. □

6.1.3 Some Applications of SCGs

Introduced in (Suri et al., 2007), one of the simplest problems that can be modeled as an SCG is a job scheduling problem where the users (players) select which machines (resources) have to execute their jobs (such as in virtualized environments or data centers). The time needed to complete a job on a machine (the resource cost) depends on its workload (the resource congestion). Assuming a single job per player to be executed on a single machine without preemption, the players' actions are singletons and the problem fits in the specific setting of SSCGs. The case of a single-leader Stackelberg game arises when one of the players is the owner of the machines and is willing to share her resources with the others, but, being the owner, she decides which resource/machine to pick before the others do. Under the assumption that the players schedule their jobs in an open-loop fashion, *i.e.*, without any knowledge of the current congestion of the machines, it is plausible that the followers could not observe the machine on which the leader's job is running. It is therefore natural, for the leader, to try and achieve a smaller cost by committing to a mixed strategy.

Another application can be found in facility location problems (Konur and Geunes, 2012) where the players are firms and they have to decide on which site to locate their infrastructures (which, depending on the specific application, may be, *e.g.*, factories, shops, or mineral extraction plants). Each firm selects a location from a list of candidate sites (the resources) and the cost it incurs depends on the number of firms that made the same choice. In these scenarios, the single-leader Stackelberg game case arises whenever one of the firms either has a competitive advantage over the others (due to, *e.g.*, being a governmental agency) or, as in the job scheduling problem, it owns the candidate sites and, thus, can decide before the other ones where to locate its infrastructures. Mixed-strategy commitments are plausible when the time between the choice of the location and the beginning of the construction works of the facility is extremely long, due to, *e.g.*, administrative issues and/or the time needed for obtaining the authorizations. In this case, the follower could prefer to choose her location before observing the beginning of the construction works of the leader's facility to avoid incurring an excessive delay with respect to the leader.

6.2 Computational Complexity Results on SSCGs

In this section, we address the problem of computing SPNEs in SSCGs, *i.e.*, games with singleton actions. We start our analysis with some negative results that identify which are the hard-to-solve game instances. In particular, Subsection 6.2.1 focuses on the case of general SSCGs with players having different action spaces, while Subsection 6.2.2 analyses the special case of SSSCGs, in which the players are symmetric.

6.2.1 NP-hardness and Inapproximability of SSCGs

We start showing that the problem of computing an S/WSPNE in SSCGs with different action spaces is computationally intractable even if the leader can only select a single resource and all the costs are monotonic functions of the resource congestion. This also shows that, in general non-Stackelberg singleton congestion games with different action spaces, computing an NE which maximizes/minimizes the usage of a resource (or the cost incurred by a player) is hard, which may be of independent interest. Moreover, given that our intractability results hold even when the leader has only one resource available, computing an S/WSPNE in SSCGs with different action spaces is intractable even if we restrict the leader to pure strategies.

First, we prove that finding an SSPNE is *not* in Poly-APX unless $\mathsf{P} = \mathsf{NP}$ using a reduction from 3-SAT. As a result, the leader's cost in an SSPNE cannot be approximated, in polynomial time, up to any approximation factor which depends polynomially on the size of the game given as input, unless $\mathsf{P} = \mathsf{NP}$. Then, we show that the same intractability result holds for WSPNEs by means of a different reduction still based on 3-SAT.

Computational Complexity of Finding an SSPNE in SSCGs

We analyze the problem of computing an SSPNE in SSCGs with different action spaces. The hardness and inapproximability results that we present are based on a reduction from 3-SAT (see Definition 4.3). In the following, we simply denote a 3-SAT instance as a pair (C, V). We introduce our reduction in the proof of the following theorem.

Theorem 6.1. *Computing an SSPNE in SSCGs with different action spaces is* NP*-hard, even if the leader has only one action (i.e., she can only select a single resource) and the cost functions are monotonic.*

Proof. We provide a reduction from 3-SAT showing that the existence of a polynomial-time algorithm for finding an SSPNE in SSCGs would allow us to solve any 3-SAT instance in polynomial time. Specifically, given a 3-SAT instance (C, V) and a real number $0 < \epsilon < 4$, we build an instance $\Gamma_\epsilon(C, V)$ of SSCG admitting an SSPNE in which the leader's cost is ϵ if and only if (C, V) is satisfiable; if not, the leader's cost is 4 in any SSPNE.

Mapping. $\Gamma_\epsilon(C, V)$ is defined as follows:

- $N = F \cup \{n\}$, with $F = \{p_\phi, p_{\phi,t} \mid \phi \in C\} \cup \{p_v \mid v \in V\} \cup \{p_{v,k}, p_{\bar{v},k} \mid v \in V, k \in \{1, \dots, |C|\}\} \cup \{p_{\phi,v}, p_{\phi,\bar{v}} \mid \phi \in C, v \in V\}$;
- $R = \{r_t\} \cup \{r_\phi \mid \phi \in C\} \cup \{r_v, r_{v,t}, r_{\bar{v}}, r_{\bar{v},t} \mid v \in V\} \cup \{r_{\phi,v}, r_{\phi,\bar{v}} \mid \phi \in C, v \in V\}$;
- $A_{p_\phi} = \{r_\phi\} \cup \{r_{\phi,l} \mid l \in \phi\}$, $A_{p_{\phi,t}} = \{r_\phi, r_t\}$ for all $\phi \in C$;
- $A_{p_{v,k}} = \{r_{v,t}, r_v\}$, $A_{p_{\bar{v},k}} = \{r_{\bar{v},t}, r_{\bar{v}}\}$ for all $v \in V, k \in \{1, \dots, |C|\}$;
- $A_{p_v} = \{r_t, r_{v,t}, r_{\bar{v},t}\}$ for all $v \in V$;
- $A_{p_{\phi,v}} = \{r_v, r_{\phi,v}\}$, $A_{p_{\phi,\bar{v}}} = \{r_{\bar{v}}, r_{\phi,\bar{v}}\}$ for all $\phi \in C, v \in V$;
- $A_n = \{r_t\}$.

The cost functions take values according to the following table, and satisfy $c_{r_{\bar{v}},F} = c_{r_v,F}$, $c_{r_{\phi,\bar{v}},F} = c_{r_{\phi,v},F}$, $c_{r_{\bar{v},t},F} = c_{r_{v,t},F}$, and $c_{r_t,F} = c_{r_t,n}$ (let us remark that, given $\epsilon < 4$, they are all monotonic functions of the congestion):

y	$c_{r_\phi,F}$	$c_{r_v,F}$	$c_{r_{v,t},F}$	$c_{r_{\phi,v},F}$	$c_{r_t,F}$
1	2	0	0	1	ϵ
$[2, \lvert C\rvert]$	5	0	6	6	4
$[\lvert C\rvert + 1, \infty]$	5	7	6	6	4

Figure 6.1 shows an example of a game instance $\Gamma_\epsilon(C, V)$.

Given a 3-SAT instance (C, V), $\Gamma_\epsilon(C, V)$ can be constructed in polynomial time, as it features $n = 2|C| + |V| + 4|C||V| + 1$ players and $r = |C| + 4|V| + 2|C||V| + 1$ resources. Since, in $\Gamma_\epsilon(C, V)$, the leader can only select a single resource, r_t, the only leader's commitment is $x_n \in \Delta_n : x_n^{r_t} = 1$. As a result, the leader's cost is ϵ if and only if no follower selects resource r_t; otherwise, it is 4.

If. Assume that (C, V) is satisfiable, and let $T : V \to \{0, 1\}$ be a truth assignment satisfying all the clauses in C. Using T, we show how to recover a followers' action profile $a_{-n} = (a_1, \dots, a_{n-1}) \in A_F$ such that $a_{-n} \in \mathcal{E}(x_n)$, with $x = (x_n, a_{-n})$ providing $c_n^x = \epsilon$. Note that, since ϵ is the minimum cost the leader can achieve and the followers behave in favor of the leader, x is an SSPNE. In particular, let $a_{p_{\phi,t}} = r_\phi$, for all $\phi \in C$.

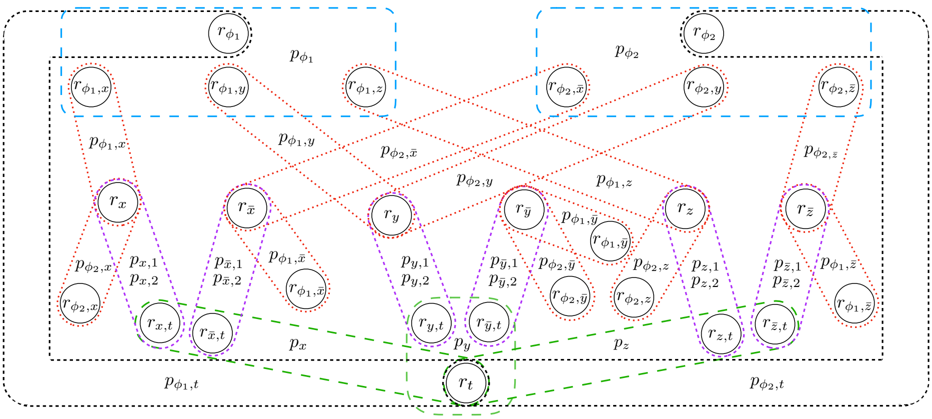


Figure 6.1: *Example of a game instance $\Gamma_\epsilon(C, V)$ used in the reduction in the proof of Theorem 6.1, with $V = \{x, y, z\}$, $C = \{\phi_1, \phi_2\}$, $\phi_1 = x \vee y \vee z$, and $\phi_2 = \bar{x} \vee y \vee \bar{z}$.*

Moreover, if $T(v) = 1$, let $a_{p_v} = r_{\bar{v},t}$ and $a_{p_{\phi,v}} = r_v$, $a_{p_{\phi,\bar{v}}} = r_{\phi,\bar{v}}$ for all $\phi \in C$, while, for all $k \in \{1, \dots, |C|\}$, let $a_{p_{v,k}} = r_{v,t}$ and $a_{p_{\bar{v},k}} = r_{\bar{v}}$. Instead, if $T(v) = 0$, let $a_{p_{\bar{v}}} = r_{v,t}$ and $a_{p_{\phi,\bar{v}}} = r_{\bar{v}}$, $a_{p_{\phi,v}} = r_{\phi,v}$ for all $\phi \in C$, while, for all $k \in \{1, \dots, |C|\}$, let $a_{p_{\bar{v},k}} = r_{\bar{v},t}$ and $a_{p_{v,k}} = r_v$. Notice that, since either $T(v) = 1$ or $T(v) = 0$, two cases are possible. If $T(v) = 1$, we have $\nu^{r_v}_{a_{-n}} = |C|$ (followers $p_{\phi,v}$), $\nu^{r_{\bar{v}}}_{a_{-n}} = |C|$ (followers $p_{\bar{v},k}$), $\nu^{r_{v,t}}_{a_{-n}} = |C|$ (followers $p_{v,k}$), and $\nu^{r_{\bar{v},t}}_{a_{-n}} = 1$ (follower p_v). If $T(v) = 0$, we have $\nu^{r_{\bar{v}}}_{a_{-n}} = |C|$ (followers $p_{\phi,\bar{v}}$), $\nu^{r_v}_{a_{-n}} = |C|$ (followers $p_{v,k}$), $\nu^{r_{\bar{v},t}}_{a_{-n}} = |C|$ (followers $p_{\bar{v},k}$), and $\nu^{r_{v,t}}_{a_{-n}} = 1$ (follower p_v). Assume, w.l.o.g., $T(v) = 1$, as the other case is analogous. First, no follower $p_{\phi,v}$ would deviate from r_v to $r_{\phi,v}$, as, otherwise, she would incur a cost of at least 1, rather than 0. The same holds for followers $p_{\phi,\bar{v}}$, as their cost is at most 6 while, if any of them switched to $r_{\bar{v}}$, she would incur a cost of 7. Similarly, followers $p_{v,k}$ would not deviate from $r_{v,t}$ (as $6 < 7$) and followers $p_{\bar{v},k}$ would not deviate from $r_{\bar{v}}$ (as $0 < 6$). Since $\nu^{r_{\bar{v},t}}_{a_{-n}} = 1$, follower p_v would not deviate from $r_{\bar{v},t}$ (as $0 < 6$ and $0 < 4$). Furthermore, since T is a truth assignment satisfying (C, V), at least one literal $l \in \phi$ evaluates to true under T for every $\phi \in C$. Let $a_{p_\phi} = r_{\phi,l}$ for every $\phi \in C$. Since l evaluates to true, it must be $a_{p_{\phi,l}} = r_l$, thus p_ϕ is the only follower who selects $r_{\phi,l}$. As a result, p_ϕ incurs a cost of 1, and she has no incentive to deviate. Finally, $p_{\phi,t}$ does not deviate from r_ϕ to r_t as $2 < 4$. Thus, we can conclude that a_{-n} is an NE and that, since no follower chose r_t, the leader's cost is ϵ.

Only if. Suppose there exists an SSPNE $x = (x_n, a_{-n})$ in which the leader's cost is ϵ. We show that, in polynomial time, one can recover a truth assignment T that satisfies all the clauses in C from $a_{-n} = (a_1, \ldots, a_{n-1})$. First, notice that no follower selects r_t in a_{-n} as, otherwise, the leader's cost would be $4 > \epsilon$. As a consequence, all followers $p_{\phi,t}$ and p_v must select one of the other resources available to them, i.e, $a_{p_{\phi,t}} = r_\phi$ and $a_{p_v} \in \{r_{v,t}, r_{\bar{v},t}\}$. Moreover, there cannot be two followers using resource r_ϕ for every $\phi \in C$ as, otherwise, $p_{\phi,t}$ would have an incentive to deviate from r_ϕ to r_t (as $5 > 4$). Thus, $a_{p_\phi} \neq r_\phi$, and, for all $\phi \in C$, there must be a literal $l \in \phi$ such that $a_{p_\phi} = r_{\phi,l}$. In addition, there cannot be two followers selecting $r_{\phi,l}$ as, otherwise, p_ϕ would have an incentive to deviate to r_ϕ (as $5 < 6$). Thus, it must be the case that $a_{p_{\phi,l}} = r_l$. This implies that $\nu^{r_l}_{a_{-n}} \leq |C|$ as, otherwise, the cost of $p_{\phi,l}$ would be $7 > 6$, and that follower would change resource, switching to $r_{\phi,l}$. Thus, at least one of the followers $p_{l,k}$ must select $r_{l,t}$ as, otherwise, $\nu^{r_l}_{a_{-n}} > |C|$. As a consequence, if l is positive and $v(l) = v$, p_v selects $r_{\bar{v},t}$ as, if she selected $r_{v,t}$, she would have an incentive to deviate (as $6 > 4$). Moreover, no other follower would select $r_{\bar{v},t}$ as, otherwise, p_v would deviate to r_t (as $6 > 4$). This implies that $\nu^{r_{\bar{v},t}}_{a_{-n}} = 1$ (follower p_v) and all the followers $p_{\bar{v},k}$ select resource $r_{\bar{v}}$, while the followers $p_{\phi,\bar{v}}$ choose resources $r_{\phi,\bar{v}}$. On the other hand, if l is negative and $v(l) = v$, similar arguments allow us to conclude that $\nu^{r_{v,t}}_{a_{-n}} = 1$ (follower p_v) and all the followers $p_{v,k}$ select resource r_v, while the followers $p_{\phi,v}$ choose resources $r_{\phi,v}$. As a result, either $\nu^{r_{v,t}}_{a_{-n}} = 1$ or $\nu^{r_{\bar{v},t}}_{a_{-n}} = 1$. In conclusion, we can define a well-defined truth assignment T such that $T(v) = 1$ if $a_{p_v} = r_{\bar{v},t}$ and $T(v) = 0$ if $a_{p_v} = r_{v,t}$. As previously shown, for every $\phi \in C$ there exists a literal $l \in \phi$ such that $a_{p_{\phi,l}} = r_l$, which, letting $v = v(l)$, implies $\nu^{r_{\bar{v},t}}_{a_{-n}} = 1$. Thus, $T(v(l)) = 1$ if l is positive, while $\nu^{r_{v,t}}_{a_{-n}} = 1$ and $T(v(l)) = 0$ if negative. Hence, T satisfies all the clauses. □

The proof of Theorem 6.1 also shows the following:

Corollary 6.1.1. *In general non-Stackelberg singleton congestion games with different action spaces, computing an NE minimizing the cost of a given player (or the usage of a given resource) is* NP*-hard even if the cost functions are monotonic.*

Proof. The result is easily proved by noticing that, in the $\Gamma_\epsilon(C, V)$ games defined in the proof of Theorem 6.1, since the leader can only use a single resource any SSPNE $x = (x_n, a_{-n})$ is also an NE. Thus, given that the followers behave in favor of the leader, such games admit an NE with $c^x_n = \epsilon$ if and only if the corresponding 3-SAT instance is satisfiable; otherwise $c^x_n = 4$ in any NE. As a result, due to 3-SAT being NP-complete, computing an NE minimizing the cost of a given player (the leader) is NP-hard. Since

$c_n^x = \epsilon$ if and only if $\nu_{a_{-n}}^{r_t} = 0$, the same holds for the problem of finding an NE which minimizes the usage of a given resource. $\square$

Theorem 6.1 also implies that the leader's cost in an SSPNE cannot be efficiently approximated up to any factor which depends polynomially on the size of the input:

Corollary 6.1.2. *The problem of computing an SSPNE in SSCGs with different action spaces is not in Poly-*APX *unless* $\mathsf{P} = \mathsf{NP}$ *even if the leader has only one action and the cost functions are monotonic.*

Proof. Given a 3-SAT instance (C, V), let us build an SSCG $\Gamma_\epsilon(C, V)$ as in the proof of Theorem 6.1. We have already proved that $\Gamma_\epsilon(C, V)$ admits an SSPNE $x = (x_n, a_{-n})$ in which $c_n^x = \epsilon$ if and only if (C, V) is satisfiable and that, otherwise, $c_n^x = 4$. Let $\epsilon = \frac{4}{2^{n+r}}$. Assume that there exists a polynomial-time approximation algorithm $\mathcal{A}$ with approximation factor poly(n, r), *i.e.*, a polynomial function of n and r. Assume (C, V) is satisfiable. $\mathcal{A}$ applied to $\Gamma_\epsilon(C, V)$ would return a solution with $c_n^x \leq \frac{4}{2^{n+r}}$ poly(n, r). Since, for n and r large enough, $\frac{4}{2^{n+r}}$ poly$(n, r) < 4$, $\mathcal{A}$ would allows us to decide in polynomial time whether (C, V) is satisfiable, a contradiction unless $\mathsf{P} = \mathsf{NP}$. $\square$

Since the intractability results in Theorem 6.1 and Corollary 6.1.2 hold even when the leader can select only a single resource, we also obtain the following:

Corollary 6.1.3. *The problem of computing an SSPNE in SSCGs with different action spaces is* NP*-hard and not in Poly-*APX *unless* $\mathsf{P} = \mathsf{NP}$ *even if we restrict the leader to pure-strategy commitments.*

Since the followers break ties in favor of the leader in the reduction, the results in Theorem 6.1 and Corollaries 6.1.2 and 6.1.3 do not apply to the problem of finding a WSPNE. We consider this case in the next subsection.

Computational Complexity of Finding a WSPNE in SSCGs

The hardness and inapproximability results that we are about to present for the problem of computing a WSPNE in SSCGs with different action spaces are still based on 3-SAT but rely on a different reduction.

Theorem 6.2. *Computing a WSPNE in SSCGs with different action spaces is* NP*-hard even if the leader has only one action and the cost functions are monotonic.*

Proof. We provide a reduction from 3-SAT showing that the existence of a polynomial-time algorithm for computing a WSPNE in SSCGs would allow us to solve any 3-SAT instance in polynomial time. Specifically, given a 3-SAT instance (C, V) and a real number $0 < \epsilon < 4$, we build an SSCG instance $\Gamma_\epsilon(C, V)$ such that it admits a WSPNE where the leader's cost is ϵ if and only if the 3-SAT instance admits a *no* answer, *i.e.*, if and only if (C, V) is not satisfiable. Instead, if the 3-SAT instance has answer *yes*, *i.e.*, if (C, V) is satisfiable, then the leader's cost is 4 in any WSPNE.

Mapping. $\Gamma_\epsilon(C, V)$ is defined as follows:

- $N = F \cup \{n\}$, where $F = \{p_{\phi,t} \mid \phi \in C\} \cup \{p_{v,t}, p_v, p_{\bar{v}} \mid v \in V\} \cup \{p_{l,\phi} \mid \phi \in C, l \in \phi\} \cup \{p_{\phi,v}, p_{\phi,\bar{v}} \mid \phi \in C, v \in V\}$;
- $R = \{r_t\} \cup \{r_\phi \mid \phi \in C\} \cup \{r_{v,t}, r_v, r_{\bar{v}} \mid v \in V\} \cup \{r_{\phi,v}, r_{\phi,\bar{v}} \mid \phi \in C, v \in V\}$;
- $A_{p_{\phi,t}} = \{r_\phi, r_t\}$ for all $\phi \in C$;
- $A_{p_v} = \{r_{v,t}, r_v\}$, $A_{p_{\bar{v}}} = \{r_{v,t}, r_{\bar{v}}\}$, $A_{p_{v,t}} = \{r_{v,t}, r_t\}$ for all $v \in V$;
- $A_{p_{l,\phi}} = \{r_\phi\} \cup \{r_{\phi,l}\}$ for al $\phi \in C, l \in \phi$;
- $A_{p_{\phi,v}} = \{r_v, r_{\phi,v}\}$, $A_{p_{\phi,\bar{v}}} = \{r_{\bar{v}}, r_{\phi,\bar{v}}\}$ for all $\phi \in C, v \in V$;
- $A_n = \{r_t\}$.

The cost functions take values according to the following table, and satisfy $c_{r_{\bar{v}},F} = c_{r_v,F}$, $c_{r_{\phi,\bar{v}},F} = c_{r_{\phi,v},F}$, and $c_{r_t,F} = c_{r_t,n}$ (let us remark that, given $\epsilon < 4$, they are all monotonic functions of the resource congestion):

y	$c_{r_\phi,F}$	$c_{r_v,F}$	$c_{r_{v,t},F}$	$c_{r_{\phi,v},F}$	$c_{r_t,F}$
1	2	1	2	0	ϵ
$[2, \lvert C\rvert]$	5	1	5	7	ϵ
$\lvert C\rvert + 1$	5	6	5	7	ϵ
$[\lvert C\rvert + \lvert V\rvert + 1, \infty]$	5	6	5	7	4

Figure 6.2 shows an example of the game instance $\Gamma_\epsilon(C, V)$.

Given (C, V), $\Gamma_\epsilon(C, V)$ can be constructed in polynomial time, as it features $n = 3|C| + |C| + 3|V| + 2|C||V| + 1$ players and $r = |C| + 3|V| + 2|C||V| + 1$ resources. Observe that, in $\Gamma_\epsilon(C, V)$, the leader can only select a single resource r_t and, hence, the only leader's commitment is $x_n \in \Delta_n : x_n^{r_t} = 1$. As a result, the leader's cost is 4 if and only if all followers $p_{\phi,t}$ and $p_{v,t}$ select resource r_t; otherwise, it is ϵ.

If. Suppose that the 3-SAT instance has answer *no*, *i.e.*, there is no truth assignment to the variables in V that satisfies all the clauses in C. We prove

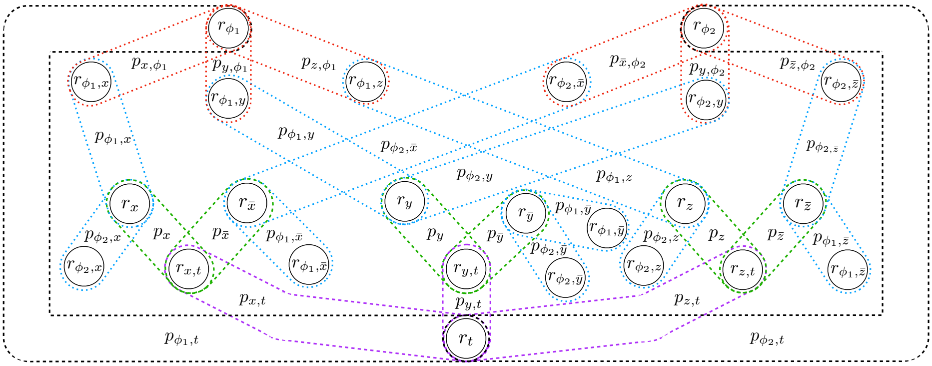


Figure 6.2: *Example of a game instance $\Gamma_\epsilon(C,V)$ used in the reduction in the proof of Theorem 6.2 with $V = \{x, y, z\}$, $C = \{\phi_1, \phi_2\}$, $\phi_1 = x \vee y \vee z$, and $\phi_2 = \bar{x} \vee y \vee \bar{z}$.*

that, in that case, $\Gamma_\epsilon(C,V)$ admits a WSPNE with leader's cost equal to ϵ. By contradiction, let us assume there exists a WSPNE $x = (x_n, a_{-n})$ in which the leader's cost c_n^x is $4 > \epsilon$. We show that $a_{-n} = (a_1, \dots, a_{n-1})$ can be employed to recover, in polynomial time, a truth assignment T that satisfies all the clauses in C, which is a contradiction. First, let us notice that all the followers $p_{\phi,t}$ and $p_{v,t}$ select r_t in a_{-n} as, otherwise, the leader's cost would be $\epsilon < 4$. As a result, $a_{p_{\phi,t}} = r_t$ for all $\phi \in C$ and $a_{p_{v,t}} = r_t$ for all $v \in V$. Thus, for every $v \in V$, at least one between p_v and $p_{\bar{v}}$ must select $r_{v,t}$ as, otherwise, player $p_{v,t}$ would deviate from r_t (as $2 < 4$). If $a_{p_v} = r_{v,t}$, then all the followers $p_{\phi,v}$ select r_v as, otherwise, p_v would have an incentive to deviate from $r_{v,t}$ (since $\nu_{a_{-n}}^{r_v} < |C|$ and p_v would incur a cost of $1 < 2$ by switching to r_v). Similarly, if $a_{p_{\bar{v}}} = r_{v,t}$, then all the followers $p_{\phi,\bar{v}}$ select $r_{\bar{v}}$. Let us define a truth assignment T such that $T(v) = 1$ if $a_{p_v} = r_v$, $T(v) = 0$ if $a_{p_{\bar{v}}} = r_{\bar{v}}$ and $T(v)$ is either 1 or 0 whenever $a_{p_v} = a_{p_{\bar{v}}} = r_{v,t}$. Clearly, T is well-defined. Since $a_{p_{\phi,t}} = r_t$ for all $\phi \in C$, there must be at least one follower using resource r_ϕ for every $\phi \in C$ as, otherwise, $p_{\phi,t}$ would have an incentive to deviate from r_t to r_ϕ (as $4 > 2$). Thus, for each $\phi \in C$ there must be a literal $l \in \phi$ such that $a_{p_{l,\phi}} = r_\phi$. This implies that $a_{p_{\phi,l}} = r_{\phi,l}$ as, otherwise, follower $p_{l,\phi}$ would deviate from r_ϕ to $r_{\phi,l}$ (as $2 > 0$). As a result, it must be the case that $a_{p_l} = r_l$, since, if $a_{p_l} = r_{v(l),t}$, then $p_{\phi,l}$ would select r_l instead of $r_{\phi,l}$. Thus, $T(v(l)) = 1$ if l is positive and $T(v(l)) = 0$ if negative. Therefore, T satisfies all the clauses, a contradiction.

Only if. Suppose that the 3-SAT instance admits answer *yes*, i.e., there exists a truth assignment to the variables which satisfies all the clauses in

C. We prove that in any WSPNE of $\Gamma_\epsilon(C,V)$ the leader's cost is $4 > \epsilon$. Let $T : V \to \{1,0\}$ be one such truth assignment. We show how to recover from T a followers' action profile $a_{-n} = (a_1,\dots,a_{n-1}) \in A_F$ such that $a_{-n} \in \mathcal{E}(x_n)$, with $x = (x_n, a_{-n})$ providing $c_n^x = 4$. Since 4 is the maximum cost the leader can achieve and the followers behave against the leader, x is clearly a WSPNE. In particular, let $a_{p_{\phi,t}} = r_t$ for all $\phi \in C$ and $a_{p_{v,t}} = r_t$ for all $v \in V$. Moreover, if $T(v) = 1$, let $a_{p_v} = r_v$, $a_{p_{\bar{v}}} = r_{v,t}$, and, for all $\phi \in C$, $a_{p_{\phi,v}} = r_{\phi,v}$ and $a_{p_{\phi,\bar{v}}} = r_{\bar{v}}$. Additionally, for every clause $\phi \in C$ and $l \in \phi$ such that $v(l) = v$, let $a_{p_{l,\phi}} = r_\phi$ if l is positive, while $a_{p_{l,\phi}} = r_{\phi,l}$ if it is negative. Conversely, if $T(v) = 0$, let $a_{p_{\bar{v}}} = r_{\bar{v}}$, $a_{p_v} = r_{v,t}$, and, for all $\phi \in C$, $a_{p_{\phi,\bar{v}}} = r_{\phi,\bar{v}}$ and $a_{p_{\phi,v}} = r_v$. Furthermore, for every clause $\phi \in C$ and $l \in \phi$ such that $v(l) = v$, let $a_{p_{l,\phi}} = r_\phi$ if l is negative, and $a_{p_{l,\phi}} = r_{\phi,l}$ if it is positive. Notice that, since either $T(v) = 1$ or $T(v) = 0$, one between p_v and $p_{\bar{v}}$ selects $r_{v,t}$. Assume, w.l.o.g., $T(v) = 1$ (as the other case is analogous). First, no follower $p_{\phi,v}$ would deviate from $r_{\phi,v}$ to r_v, as, otherwise, she would incur a cost of 1, rather than 0. The same holds for followers $p_{\phi,\bar{v}}$, as their cost is 1 while, if any of them switched to $r_{\phi,\bar{v}}$, she would incur a cost of 7, because $a_{p_{\bar{v},\phi}} = r_{\phi,\bar{v}}$. Similarly, since there is one follower selecting $r_{v,t}$, follower $p_{v,t}$ would not deviate from r_t (as $4 < 5$), while follower p_v would not deviate from r_v because her cost is $1 < 5$ and $p_{\bar{v}}$ would not switch from $r_{v,t}$ (as she would get 6 rather than 1). Furthermore, since T is a truth assignment satisfying (C,V), for each $\phi \in C$ there exists at least one literal $l \in \phi$ that evaluates to true under T. Thus, $p_{l,\phi}$ would not deviate from r_ϕ (as she pays either 2 or 5 instead of 7), and all the followers $p_{\phi,t}$ would not deviate from r_t (as $4 < 5$). We can conclude that a_{-n} is an NE. Since $|C| + |V|$ follower use r_t, $c_n^x = 4$. □

Theorem 6.2 also implies the following:

Corollary 6.2.1. *In general non-Stackelberg singleton congestion games with different action spaces, computing an NE maximizing the cost of a given player (or the usage of a given resource) is* NP*-hard even if the cost functions are monotonic.*

Proof. In games $\Gamma_\epsilon(C,V)$ such as those used in the proof of Theorem 6.2, any WSPNE is also an NE (since the leader can choose a single action). Moreover, $\Gamma_\epsilon(C,V)$ admits a WSPNE $x = (x_n, a_{-n})$ in which $c_n^x = 4$ if and only if the given 3-SAT instance has answer *no*, otherwise $c_n^x = \epsilon$. This proves the result for the problem of finding an NE maximizing the cost of a given player. Since $c_n^x = 4$ if and only if $\nu_{a_{-n}}^{r_t} = |C| + |V|$, the same holds for computing an NE maximizing the usage of a given resource. □

Furthermore, from Theorem 6.2 it directly follows that the leader's cost in a WSPNE cannot be efficiently approximated up to any approximation factor which depends polynomially on the size of the input:

Corollary 6.2.2. *The problem of computing a WSPNE in SSCGs with different action spaces is not in Poly-*APX *unless* $\mathsf{P} = \mathsf{NP}$*, even if the leader has only one action and the cost functions are monotonic.*

Proof. Given a 3-SAT instance (C, V), let us build an instance $\Gamma_\epsilon(C, V)$ of SSCG as in the proof of Theorem 6.2. We have already proven that $\Gamma_\epsilon(C, V)$ admits a WSPNE $x = (x_n, a_{-n})$ in which $c_n^x = \epsilon$ if and only if the 3-SAT instance has answer *no*; otherwise, $c_n^x = 4$ in any WSPNE. Let $\epsilon = \frac{4}{2^{n+r}}$. Assume that there exists a polynomial-time approximation algorithm $\mathcal{A}$ with approximation factor poly(n, r), *i.e.*, a polynomial function of n and r. Assume the answer to the 3-SAT instance is *no*. $\mathcal{A}$ applied to $\Gamma_\epsilon(C, V)$ would return a solution with $c_n^x \leq \frac{4}{2^{n+r}}$ poly(n, r). Since, for n and r large enough, $\frac{4}{2^{n+r}}$ poly$(n, r) < 4$, $\mathcal{A}$ would allow us to decide in polynomial time whether the answer to the 3-SAT instance is *yes* or *no*, a contradiction unless $\mathsf{P} = \mathsf{NP}$. □

Since in the reduction the leader only has one resource available we can conclude the following:

Corollary 6.2.3. *The problem of computing a WSPNE in SSCGs with different action spaces is* NP*-hard and not in Poly-*APX *unless* $\mathsf{P} = \mathsf{NP}$ *even if we restrict the leader to pure-strategy commitments.*

6.2.2 NP-hardness and Inapproximability of SSSCGs

Now, we focus on SSSCGs (the subset of SSCGs in which the players have identical action spaces), showing that the problem of finding an S/WSPNE in such games is NP-hard and *not* in Poly-APX unless $\mathsf{P} = \mathsf{NP}$. This result matches the other result that we have established for the problem of computing an S/WSPNE in general SSCGs with different action spaces. For SSSCGs, the inapproximability result relies on the non-monotonicity of the players' cost functions and on the leader's ability to commit to mixed strategies. This must necessarily be the case since, as we will show in Section 6.3, the problem is easy when the cost functions are monotonic and the players are symmetric (Theorem 6.9), and the same holds even with generic cost functions if we restrict the leader to pure-strategy commitments (Theorem 6.11 and its Corollary 6.11.1).

For the problem of computing an SSPNE, we rely on a reduction from K-PARTITION, a variant of PARTITION with an additional size constraint, whereas we adopt a different reduction based on the classical version of PARTITION for the problem of computing a WSPNE. The two problems are defined as follows:

Definition 6.3 (PARTITION). *Given a finite set $\mathcal{S} = \{y_1, \dots, y_{|\mathcal{S}|}\}$ of positive integers $y_i \in \mathbb{Z}^+$ with $\sum_{i \in \mathcal{S}} y_i$ even, is there a partition $(\mathcal{S}', \mathcal{S} \setminus \mathcal{S}')$ of $\mathcal{S}$, with $\mathcal{S}' \subseteq \mathcal{S}$, such that $\sum_{y_i \in \mathcal{S}'} y_i = \sum_{y_i \in \mathcal{S} \setminus \mathcal{S}'} y_i$?*

Definition 6.4 (K-PARTITION). *Given a finite set $\mathcal{S} = \{y_1, \dots, y_{|\mathcal{S}|}\}$ of positive integers $y_i \in \mathbb{Z}^+$ with both $|\mathcal{S}|$ and $\sum_{i \in \mathcal{S}} y_i$ even and a positive integer $K \leq \frac{|\mathcal{S}|}{2}$, is there a partition $(\mathcal{S}', \mathcal{S} \setminus \mathcal{S}')$ of $\mathcal{S}$, with $\mathcal{S}' \subseteq \mathcal{S}$ and $|\mathcal{S}'| = K$ such that $\sum_{y_i \in \mathcal{S}'} y_i = \sum_{y_i \in \mathcal{S} \setminus \mathcal{S}'} y_i$?*

Letting $Y := \frac{1}{2} \sum_{y_i \in \mathcal{S}} y_i$, we assume for both problems that $y_i \leq Y$ for all $y_i \in \mathcal{S}$. Indeed, if some $y_i > Y$ then $\sum_{y_i \in \mathcal{S}'} y_i > Y$ holds for every $\mathcal{S}' \subseteq \mathcal{S}$ and, thus, the answer to both PARTITION and K-PARTITION is trivially *no*. PARTITION is well-known to be NP-complete (Johnson and Garey, 1979). To see that K-PARTITION is also NP-complete (its membership to NP is clear), it suffices to observe that PARTITION has answer *yes* if and only if K-PARTITION has answer *yes* for some $K \in \left\{1, \dots, \frac{|\mathcal{S}|}{2}\right\}$. This gives us a simple reduction from PARTITION to K-PARTITION: after solving K-PARTITION $\frac{|\mathcal{S}|}{2}$ times, once per value of $K \in \left\{1, \dots, \frac{|\mathcal{S}|}{2}\right\}$, if answer *yes* is found for some K, PARTITION has answer *yes*; if, instead, answer *yes* is never found, PARTITION has answer *no*.

Computational Complexity of Finding an SSPNE in SSSCGs

We start our analysis with the problem of computing an SSPNE in SSSCGs. We introduce our main reduction in the proof of the following theorem.

Theorem 6.3. *Computing an SSPNE in SSSCGs is NP-hard.*

Proof. We prove the theorem using a reduction from K-PARTITION, showing that the existence of a polynomial-time algorithm for computing an SSPNE in SSSCGs would allow us to solve K-PARTITION in polynomial time. Let $(\mathcal{S}, K)$ be a given K-PARTITION instance, and let us recall that we assumed $y_i \leq Y$ for all $y_i \in \mathcal{S}$, where $Y = \frac{1}{2} \sum_{y_i \in \mathcal{S}} y_i$. Clearly, any valid partition $(\mathcal{S}', \mathcal{S} \setminus \mathcal{S}')$ is uniquely defined by a subset $\mathcal{S}' \subseteq \mathcal{S}$ such that $\sum_{y_i \in \mathcal{S}'} y_i = Y$ and $|\mathcal{S}'| = K$. Let $w_i = \frac{y_i}{Y}$ for all $y_i \in \mathcal{S}$. Due to having $y_i \leq Y$ for all $y_i \in \mathcal{S}$, we also have $w_i \leq 1$. Given $(\mathcal{S}, K)$, we build an instance $\Gamma_\epsilon(\mathcal{S}, K)$ of SSSCG with $0 < \epsilon < 1$ such that there exists an SSPNE $x = (x_n, \nu)$ in which $c_n^x = \epsilon$ if and only if the K-PARTITION instance $(\mathcal{S}, K)$ admits answer *yes*.

Mapping. $\Gamma_\epsilon(\mathcal{S}, K)$ is defined as follows:

- $N = F \cup \{n\}$, with $|F| = 4|\mathcal{S}| + 2$;
- $R = \{r_{t_1}\} \cup \{r_{t_2}\} \cup \{r_i \mid y_i \in \mathcal{S}\}$;

The players' cost functions are specified in the following table:

y	$c_{r_i,F}$	$c_{r_i,n}$	$c_{r_{t_1},F}$	$c_{r_t,n}$	$c_{r_{t_2},F}$	$c_{r_{t_2},n}$
1	$2Y$	Y	$3Y^2$	Y^4	1	Y^4
2	0	Y	$3Y^2$	Y^4	$4Y^2$	Y^4
3	$\frac{1}{w_i}$	ϵ	$3Y^2$	Y^4	$4Y^2$	Y^4
4	$\frac{2s-\frac{1}{w_i}+1}{w_i}$	Y	$3Y^2$	Y^4	$4Y^2$	Y^4
$[5, 4\|\mathcal{S}\| - 2K]$	$4Y^2$	Y	$3Y^2$	Y^4	$4Y^2$	Y^4
$4\|\mathcal{S}\| - 2K + 1$	$4Y^2$	Y	$2Y$	Y^4	$4Y^2$	Y^4
$4\|\mathcal{S}\| - 2K + 2$	$4Y^2$	Y	1	Y^4	$4Y^2$	Y^4
$[4\|\mathcal{S}\| - 2K + 3, \infty]$	$4Y^2$	Y	0	Y^4	$4Y^2$	Y^4

Clearly, $\Gamma_\epsilon(\mathcal{S}, K)$ can be built in polynomial time, as it features $n = 4|\mathcal{S}| + 3$ players and $r = |\mathcal{S}| + 2$ resources.

If. Suppose that the K-PARTITION instance $(\mathcal{S}, K)$ admits a *yes* answer. Let $\mathcal{S}' \subseteq \mathcal{S}$ be a set of integers with $|\mathcal{S}'| = K$ and $\sum_{y_i \in \mathcal{S}'} y_i = Y$. We prove that $\Gamma_\epsilon(\mathcal{S}, K)$ admits an SSPNE $x = (x_n, \nu)$ in which $c_n^x = \epsilon$. Given $\mathcal{S}'$, we build the followers' configuration $\nu \in \mathbb{R}^r$ and the leader's strategy $x_n \in \Delta_n$. Let $\nu^{r_i} = 2$ and $x_n^{r_i} = w_i$ for all $y_i \in \mathcal{S}'$, while, for every $y_i \notin \mathcal{S}'$, let $\nu^{r_i} = 0$ and $x_n^{r_i} = 0$. Moreover, let $\nu^{r_{t_1}} = 4|\mathcal{S}| - 2K + 1$, $x_n^{r_{t_1}} = 0$, $\nu^{r_{t_2}} = 1$, and $x_n^{r_{t_2}} = 0$. First, let us observe that the leader's strategy x_n is well-defined, as

$$\sum_{y_i \in \mathcal{S}} x_n^{r_i} + x_n^{r_{t_1}} + x_n^{r_{t_2}} = \sum_{y_i \in \mathcal{S}'} x_n^{r_i} = \sum_{y_i \in \mathcal{S}'} w_i = \sum_{y_i \in \mathcal{S}'} \frac{y_i}{Y} = 1,$$

where the last equality follows from the fact that $\mathcal{S}'$ defines a partition of $\mathcal{S}$. Next, we show that ν is an NE for x_n with the following argument.

- All the followers who selected resource r_i, with $y_i \in \mathcal{S}'$, do not have any incentive to change resource, as their cost is $w_i \cdot \frac{1}{w_i} = 1$ and they cannot improve it by switching to another resource. Indeed, if they selected a resource r_j with $y_j \in \mathcal{S}'$, they would incur a cost of $\frac{1}{w_j} \cdot (1 - w_j) + \frac{2Y - \frac{1}{w_j} + 1}{w_j} \cdot w_j = 2Y > 1$. Similarly, their cost would be $2Y$ if they choose r_j with $y_j \notin \mathcal{S}'$. They would not benefit from choosing resource r_{t_1}, as they would incur a cost of 1, which is the same as their current cost, and they would not switch to resource r_{t_2}, as their cost would become $4Y^2 > 1$.

- All the followers who selected resource r_{t_1} incur a cost of $2Y$. Thus, they do not have an incentive to deviate to a resource r_i with $y_i \in \mathcal{S}'$, as they would still incur a cost of $2Y$. The same holds for resources r_i with $y_i \notin \mathcal{S}'$. Similarly, if they chose to play r_{t_2}, they would incur a cost of $4Y^2 > 2Y$.
- The follower who chose resource r_{t_2} does not deviate, as her cost is 1 and she would incur a cost of $2Y$ and 1 if she switched to resource r_i (for some $y_i \in \mathcal{S}$) and r_{t_1}, respectively.

Overall, the leader's cost is:

$$c_n^x = \sum_{y_i \in \mathcal{S}} x_n^{r_i} c_{r_i,n}(\nu^{r_i} + 1) + x_n^{r_{t_1}} c_{r_{t_1},n}(\nu^{r_{t_1}} + 1) + x_n^{r_{t_2}} c_{r_{t_2},n}(\nu^{r_{t_2}} + 1) =$$
$$= \sum_{y_i \in \mathcal{S}'} x_n^{r_i} c_{r_i,n}(\nu^{r_i} + 1) = \sum_{y_i \in \mathcal{S}'} \epsilon w_i = \epsilon.$$

Only if. Suppose that $\Gamma_\epsilon(\mathcal{S}, K)$ has an SSPNE $x = (x_n, \nu)$ in which $c_n^x = \epsilon$. Then, $x_n^{r_{t_1}} = x_n^{r_{t_2}} = 0$ must hold. Moreover, the leader must place positive probability only on resources r_i with $\nu^{r_i} = 2$. Clearly, there is always a resource r_i with $\nu^{r_i} = 2$ and $x_n^{r_i} > 0$. Next, we prove that $\nu^{r_{t_1}} = 4|\mathcal{S}| - 2K + 1$. By contradiction, assume that $\nu^{r_{t_1}} \neq 4|\mathcal{S}| - 2K + 1$. Three cases are possible.

- $\nu^{r_{t_1}} = 0$ implies that either there exists at least one resource r_i with $\nu^{r_i} \geq 5$ or $\nu^{r_{t_2}} = 2$, but, then, the followers who chose r_i or, respectively, r_{t_2}, would deviate by choosing r_{t_1}, decreasing their cost from $4Y^2$ to $3Y^2$.
- $1 \leq \nu^{r_{t_1}} \leq 4|\mathcal{S}| - 2K$ implies that the followers who selected r_{t_1} incur a cost of $3Y^2$. Thus, they would deviate to some resource r_i with $\nu^{r_i} = 2$, since their cost would be at most $\frac{2Y - \frac{1}{w_i} + 1}{w_i} < 3Y^2$.
- $\nu^{r_{t_1}} \geq 4|\mathcal{S}| - 2K + 2$ implies that the followers' cost when they deviate by playing resource r_{t_1} is 0. Thus, the followers who selected

a resource r_i with $\nu^{r_i} = 2$ and $x_n^{r_i} > 0$ would change resource, since their current cost is strictly greater than 0.

The only remaining option for $\nu \in \mathcal{E}(x_n)$ is $\nu^{r_{t_1}} = 4|\mathcal{S}| - 2K + 1$. Then, $\nu^{r_{t_2}} = 1$ must hold as, if $\nu^{r_{t_2}} = 0$, a follower would switch form resource r_{t_1} to resource r_{t_2} (incurring a cost of 1 instead of $2Y > 1$), while, if $\nu^{r_{t_2}} \geq 2$, the followers who selected resource r_{t_2} would deviate to resource r_{t_1} (incurring a cost of 1 instead of $4Y^2 > 1$). Let us now consider a resource r_i with $\nu^{r_i} = 2$. We prove that $x_n^{r_i} = w_i$ by contradiction. Two cases are possible.

- If $x_n^{r_i} < w_i$, the followers' cost by switching to resource r_i satisfies

$$\frac{1}{w_i}(1 - x_n^{r_i}) + \frac{2Y - \frac{1}{w_i} + 1}{w_i} x_n^{r_i} < \frac{1}{w_i}(1 - w_i) + \frac{2Y - \frac{1}{w_i} + 1}{w_i} w_i = 2Y,$$

  where the inequality holds since the left-most quantity is a convex combination of $\frac{1}{w_i}$ and $\frac{2Y - \frac{1}{w_i} + 1}{w_i}$ with weights $(1 - x_n^{r_i})$ and $x_n^{r_i}$, and, since $\frac{1}{w_i} < \frac{2Y - \frac{1}{w_i} + 1}{w_i}$, its maximum for $x_n^{r_i} \leq w_i$ is attained at $x_n^{r_i} = w_i$. Thus, we deduce that a follower would deviate from resource r_{t_1} to r_i (as her current cost is $2Y$), contradicting the fact that $\nu \in \mathcal{E}(x_n)$.

- If $x_n^{r_i} > w_i$, we reach a contradiction since the cost incurred by the followers who are using resource r_i would be $\frac{1}{w_i} x_n^{r_i} > 1$ and they would deviate playing resource r_{t_1}, decreasing their cost to 1.

We have shown that $x_n^{r_i} = w_i$ for every resource r_i with $\nu^{r_i} = 2$. Finally, let r_i be a resource with $\nu^{r_i} \neq 2$. Clearly, it must be the case that $x_n^{r_i} = 0$ since the leader's cost is ϵ. Moreover, it cannot be the case that $\nu^{r_i} = 1$, as, if it were the case, the follower would deviate to resource r_{t_1} with a cost of 1, instead of $2Y$. Similarly, $\nu^{r_i} \geq 3$ cannot hold, as one of the followers who are selecting resource r_i would deviate playing r_{t_1}, since her current cost is greater than 1. Thus, either $\nu^{r_i} = 2$ or $\nu^{r_i} = 0$. As a consequence, there are K resources r_i with $\nu^{r_i} = 2$ and $x_n^{r_i} = w_i$, and $|\mathcal{S}| - K$ resources r_i with $\nu^{r_i} = 0$ and $x_n^{r_i} = 0$. Let us define $\mathcal{S}'$ as the set of $y_i \in \mathcal{S}$ such that the corresponding r_i satisfy $\nu^{r_i} = 2$. Since $\sum_{y_i \in \mathcal{S}'} x_n^{r_i} = 1$ and $x_n^{r_i} = w_i$ for all such resources r_i, we have that $\sum_{y_i \in \mathcal{S}'} w_i = \sum_{y_i \in \mathcal{S}'} \frac{y_i}{Y} = 1$, and, thus, $\sum_{y_i \in \mathcal{S}'} y_i = Y$. As a result, $(\mathcal{S}', \mathcal{S} \setminus \mathcal{S}')$ is solution to K-PARTITION. □

Next, we show that even approximating the leader's cost in an SSPNE up to any polynomial factor of the input size is hard.

Theorem 6.4. *The problem of computing an SSPNE in SSCGs is not in Poly-*APX *unless* P = NP.

Proof. In order to prove the result, we rely on the reduction introduced in the proof of Theorem 6.3. We have already shown that in an SSPNE $x = (x_n, \nu)$ of $\Gamma_\epsilon(\mathcal{S}, K)$ it holds $c_n^x = \epsilon$ if and only if the corresponding instance of K-PARTITION $(\mathcal{S}, K)$ admits a *yes* answer. Now, we prove that, when the K-PARTITION instance admits a *no* answer, $c_n^x \geq 1$ in any SSPNE. By contradiction, assume that there exists an SSPNE $x = (x_n, \nu)$ in which $c_n^x < 1$. Let $\mathcal{S}' \subseteq \mathcal{S}$ be the set of integers corresponding to a group of resources r_i with $\nu^{r_i} = 2$ (at least one must exist since the leader's cost is smaller than 1). Then, $\sum_{y_i \in \mathcal{S}'} x_n^{r_i} > \frac{Y-1}{Y}$ since $\sum_{y_i \in \mathcal{S} \setminus \mathcal{S}'} x_n^{r_i} + x_n^{r_{t_1}} + x_n^{r_{t_2}}$ must be smaller than $\frac{1}{Y}$ in order to have $c_n^x < 1$. Moreover, $x_n^{r_{t_1}} \leq \frac{1}{Y^4}$ and $x_n^{r_{t_2}} \leq \frac{1}{Y^4}$ must both hold as, if not, we would get $c_n^x \geq 1$. We prove, now, that $\nu^{r_{t_1}} = 4|\mathcal{S}| - 2K + 2$ by contradiction. We identify three cases:

- $\nu^{r_{t_1}} = 0$ implies that either there exists at least one resource r_i with $\nu^{r_i} \geq 5$ or $\nu^{r_{t_2}} = 2$, and, thus, either a follower who selected resource r_i or one who selected resource r_{t_2} would have an incentive to deviate to resource r_{t_1} (as $4Y^2 > 3Y^2$).
- $1 \leq \nu^{r_{t_1}} \leq 4|\mathcal{S}| - 2K - 1$ implies that one of the followers who selected r_{t_1} would have an incentive to deviate to r_i with $\nu^{r_i} = 2$, as she would incur a cost smaller than or equal to $\frac{2Y - \frac{1}{w_i} + 1}{w_i} < 3Y^2$.
- $\nu^{r_{t_1}} = 4|\mathcal{S}| - 2K$ implies that the cost incurred by the followers who selected resource r_{t_1} is greater or equal than $3Y^2(1 - \frac{1}{Y^4}) + \frac{2}{Y^3}$, as $x_n^{r_{t_1}} \leq \frac{1}{Y^4}$. Thus, since $\frac{2Y - \frac{1}{w_i} + 1}{w_i} < 2Y^2 < 3Y^2 - \frac{3}{Y^2} + \frac{2}{Y^2}$, these followers would deviate from r_{t_1} to a resource r_i with $\nu^{r_i} = 2$.
- $\nu^{r_{t_1}} \geq 4|\mathcal{S}| - 2K + 2$ implies that the followers' cost after deviating to resource r_{t_1} would be 0 and, since there exists at least one resource r_i with $\nu^{r_i} = 2$ and $x_n^{r_i} > 0$, one of the followers who selected such resource would switch from it in favor of r_{t_1}.

Thus, $\nu^{r_{t_1}} = 4|\mathcal{S}| - 2K + 1$. Let us consider resource r_{t_2}. If $\nu^{r_{t_2}} = 0$, the followers' cost incurred when deviating to resource r_{t_2} would be smaller than or equal to $(1 - \frac{1}{Y^4}) + \frac{4}{Y^2}$ (as $x_n^{r_{t_2}} \leq \frac{1}{Y^4}$), while the cost incurred by choosing resource r_{t_1} is at least $2Y(1 - \frac{1}{Y^4}) + \frac{1}{Y^4} > (1 - \frac{1}{Y^4}) + \frac{4}{Y^4}$. Instead, if $\nu^{r_{t_2}} \geq 2$, the followers' cost for resource r_{t_2} is $4Y^2$ and they would have an incentive to deviate to r_{t_1} to decrease their cost to 1 or less.

Thus, $\nu^{r_{t_2}} = 1$. We deduce $x_n^{r_{t_1}} = 0$ as, otherwise (*i.e.*, with $x_n^{r_{t_1}} > 0$), a follower would deviate from resource r_{t_2} to r_{t_1}, decreasing her cost to 1 or less. Let us focus on resources r_i with $\nu^{r_i} = 2$. If $x_n^{r_i} < w_i$, the followers' cost of deviating to r_i is

$$\frac{1}{w_i}(1 - x_n^{r_i}) + \frac{2Y - \frac{1}{w_i} + 1}{w_i} x_n^{r_i} < \frac{1}{w_i}(1 - w_i) + \frac{2Y - \frac{1}{w_i} + 1}{w_i} w_i = 2Y,$$

and they would deviate from r_{t_1} to r_i, as their current cost is $2Y$. Instead, if $x_n^{r_i} > w_i$ the cost of any follower who selected r_i is greater than 1 and she would deviate to resource r_{t_1} to decrease her cost to 1. Thus, $x_n^{r_i} = w_i$ for all resources r_i with $\nu^{r_i} = 2$. Now, let us consider a resource r_i with $\nu^{r_i} \neq 2$. Clearly, $x_n^{r_i} \leq \frac{1}{Y}$ must hold since $c_n^x \leq 1$. If $\nu^{r_i} = 1$, the followers' cost for resource r_i is at least $2Y\frac{1}{Y} > 1$ while, if $\nu^{r_i} \geq 3$, the followers' cost for resource r_i is at least $\frac{1}{w_i} > 1$. In both cases, the followers who selected resource r_i would have an incentive to deviate to r_{t_1} (as they would pay 1). Thus, either $\nu^{r_i} = 2$ or $\nu^{r_i} = 0$. As a consequence, there are K resources r_i with $\nu^{r_i} = 2$ and $x_n^{r_i} = w_i$ and $|\mathcal{S}| - K$ resources r_i with $\nu^{r_i} = 0$. If $c_n^x \leq 1$, there must be a subset $\mathcal{S}' \subseteq \mathcal{S}$ with $\sum_{y_i \in \mathcal{S}'} x_n^{r_i} = \sum_{y_i \in \mathcal{S}'} w_i > \frac{Y-1}{Y}$, which implies that $\sum_{y_i \in \mathcal{S}'} \frac{y_i}{Y} > \frac{Y-1}{Y}$ and $\sum_{y_i \in \mathcal{S}'} y_i > Y - 1$. Note that $y_i \in \mathbb{N}$ and $\sum_{y_i \in \mathcal{S}'} y_i = Y \sum_{y_i \in \mathcal{S}'} w_i = Y \sum_{y_i \in \mathcal{S}'} x_n^{r_i} \leq Y$. Thus, $\sum_{y_i \in \mathcal{S}'} y_i = Y$ and $(\mathcal{S}', \mathcal{S} \setminus \mathcal{S}')$ is solution to K-PARTITION. So far, we have proven that $\Gamma_\epsilon(\mathcal{S}, K)$ admits an SSPNE $x = (x_n, \nu)$ in which $c_n^x = \epsilon$ if and only if $(\mathcal{S}, K)$ has answer *yes* and that, otherwise, $c_n^x \geq 1$ in any SSPNE. Let $\epsilon = \frac{1}{2^{n+r}}$. Assume that there exists a polynomial-time approximation algorithm $\mathcal{A}$ with approximation factor poly(n, r), *i.e.*, a polynomial function of n and r. Assume $(\mathcal{S}, K)$ has answer *yes*. $\mathcal{A}$ applied to $\Gamma_\epsilon(\mathcal{S}, K)$ would return a solution with $c_n^x \leq \frac{1}{2^{n+r}}$ poly(n, r). Since, for n and r large enough, $\frac{1}{2^{n+r}}$ poly$(n, r) < 1$, $\mathcal{A}$ would allows us to establish, in polynomial time, the answer to $(\mathcal{S}, K)$, a contradiction unless P = NP. □

Computational Complexity of Finding a WSPNE in SSSCGs

We focus now on the problem of computing a WSPNE in SSSCGs. The proof of the following theorem introduces our main reduction.

Theorem 6.5. *Computing a WSPNE in SSSCGs is* NP*-hard.*

Proof. We provide a reduction from PARTITION showing that the existence of a polynomial-time algorithm for computing a WSPNE in SSSCGs would allow us to solve PARTITION in polynomial time. Given a PARTITION instance with a set $\mathcal{S}$ of positive integers, let, as in the previous proof, $Y = \frac{1}{2}\sum_{y_i \in \mathcal{S}} y_i$ and $w_i = \frac{y_i}{Y}$ for all $y_i \in \mathcal{S}$. Let us also recall that we assumed, w.l.o.g., $y_i \leq Y$ for all $y_i \in \mathcal{S}$, and, thus, $w_i \leq 1$. Given $\mathcal{S}$, we build an instance $\Gamma_\epsilon(\mathcal{S})$ of SSSCG with $0 < \epsilon < 1$ such that $c_n^x = \epsilon$ in a WSPNE $x = (x_n, \nu)$ if and only if PARTITION admits answer *yes*.

Mapping. $\Gamma_\epsilon(\mathcal{S})$ is defined as follows:

- $N = F \cup \{n\}$, with $|F| = 3|\mathcal{S}|$;
- $R = \{r_t\} \cup \{r_i \mid y_i \in \mathcal{S}\}$;

with the following cost functions:

y	$c_{r_i,F}$	$c_{r_i,n}$	$c_{r_t,F}$	$c_{r_t,n}$
1	0	ϵ	1	Y^4
2	$\frac{1}{w_i - \frac{1}{Y^4}}$	Y^4	1	Y^4
3	$\frac{1}{1 - w_i - \frac{1}{Y^4}}$	ϵ	1	Y^4
4	0	Y^4	1	Y^4
$[5, \infty]$	Y	ϵ	1	Y^4

Clearly, $\Gamma_\epsilon(\mathcal{S})$ can be built in polynomial time, as it features $n = 3|\mathcal{S}|$ players, and $r = |\mathcal{S}| + 1$ resources.

If. Suppose that the PARTITION instance admits a *yes* answer, and let $\mathcal{S}' \subseteq \mathcal{S}$ be such that $\sum_{y_i \in \mathcal{S}'} y_i = Y$. We show that there exists a WSPNE $x = (x_n, \nu)$ in which $c_n^x = \epsilon$. Let $x_n^{r_i} = w_i$ for all $y_i \in \mathcal{S}'$, $x_n^{r_i} = 0$ for all $y_i \notin \mathcal{S}'$, and $x_n^{r_t} = 0$. We prove that $c_n^x = \epsilon$ for any $x = (x_n, \nu)$ with $\nu \in \mathcal{E}(x_n)$. Assume, by contradiction, that there exists an NE $\nu \in \mathcal{E}(x_n)$ such that $c_n^x = \epsilon$. This implies that there is a resource r_i with $y_i \in \mathcal{S}'$ and either $\nu^{r_i} = 1$ or $\nu^{r_i} = 3$. If $\nu^{r_i} = 1$, the cost incurred by the followers who select r_i is $\frac{1}{w_i - \frac{1}{Y^4}} w_i > 1$ and any of them would deviate to resource r_t to decrease her cost to 1. If $\nu^{r_i} = 3$, the followers' cost is $\frac{1}{1 - w_i - \frac{1}{Y^4}}(1 - w_i) > 1$ and any of them would deviate to resource r_t. In both cases, this contradicts the fact that ν is an NE, and, thus, it must be that $\nu^{r_i} \neq 1$ and $\nu^{r_i} \neq 3$ for all $y_i \in \mathcal{S}'$. As a result, $c_n^x = \epsilon$ for any $x = (x_n, \nu)$ with $\nu \in \mathcal{E}(x_n)$.

Only if. Suppose that $\Gamma_\epsilon(\mathcal{S})$ admits a WSPNE $x = (x_n, \nu)$ in which $c_n^x = \epsilon$. Then, $x_n^{r_t} = 0$ and $x_n^{r_i} > 0$ only if resource r_i is such that $\nu^{r_i} \neq 1$ and $\nu^{r_i} \neq 3$. Let us define $R' \subseteq R$ as the set of resources r_i with $x_n^{r_i} \leq w_i - \frac{1}{Y^4}$, R'' as the set of resources r_i with $w_i - \frac{1}{Y^4} < x_n^{r_i} < w_i + \frac{1}{Y^4}$, and R''' as the set of resources r_i with $x_n^{r_i} \geq w_i + \frac{1}{Y^4}$. Let $\nu \in \mathbb{R}^r$ be a followers' configuration such that $\nu^{r_i} = 1$ for all $r_i \in R'$, $\nu^{r_i} = 0$ for all $r_i \in R''$, $\nu^{r_i} = 3$ for all $r_i \in R'''$, and $\nu^{r_t} = 3|\mathcal{S}| - \sum_{r_i \in R \setminus \{r_t\}} \nu_{r_i}$. First, we show that $\nu \in \mathcal{E}(x_n)$. Indeed, all the followers who selected resource

r_t incurs a cost of 1, all those who selected a resource $r_i \in R'$ incur a cost of $\frac{1}{w_i - \frac{1}{Y^4}} x_n^{r_i} < 1$, and all those who selected resource $r_i \in R'''$ incur a cost of $\frac{1}{1-w_i-\frac{1}{Y^4}}(1 - x_n^{r_i}) < 1$. If any follower deviated, she would incur a cost greater than or equal than 1. In particular, no follower would deviate to a resource $r_i \in R'$, as she would incur a cost that is a convex combination of values greater than 1. Similarly, no follower would deviate to a resource $r_i \in R''$ or $r_i \in R'''$, as she would incur a cost of, respectively, $\frac{1}{w_i - \frac{1}{Y^4}} x_n^{r_i} > 1$ or $Y x_n^{r_i} > 1$. Finally, no follower has an incentive to switch to resource r_t, as her cost would not decrease. This shows that, in the followers' game resulting from x_n, there exists an NE such that, whenever the leader selects a resource r_i in $R' \cup R''$, she incurs a cost of Y^4. Thus, given that $c_n^x = \epsilon$, $R' = R''' = \varnothing$ must hold. Let us define $\mathcal{S}' \subseteq \mathcal{S}$ as the set of integers $y_i \in \mathcal{S}$ whose corresponding resource r_i is such that $w_i - \frac{1}{Y^4} < x_n^{r_i} < w_i + \frac{1}{Y^4}$. For all the other resources r_i, it must be $x_n^{r_i} = 0$. Since $\sum_{y_i \in \mathcal{S}'} x_n^{r_i} = 1$, we have $\sum_{y_i \in \mathcal{S}'} \left(w_i - \frac{1}{Y^4}\right) < 1 < \sum_{y_i \in \mathcal{S}'} \left(w_i + \frac{1}{Y^4}\right)$, and, therefore,

$$\frac{Y-1}{Y} < 1 - \sum_{y_i \in \mathcal{S}'} \frac{1}{Y^4} < \sum_{y_i \in \mathcal{S}'} w_i < 1 + \sum_{y_i \in \mathcal{S}'} \frac{1}{Y^4} < \frac{Y+1}{Y},$$

which implies $Y - 1 < \sum_{y_i \in \mathcal{S}'} y_i < Y + 1$. Since $\sum_{y_i \in \mathcal{S}'} y_i$ is an integer quantity, $\sum_{y_i \in \mathcal{S}'} y_i = Y$, implying that $\mathcal{S}'$ solves PARTITION. □

Finally, we show that the same inapproximability result that we have established for SSPNEs also holds for WSPNEs.

Theorem 6.6. *The problem of computing a WSPNE in SSCGs is not in Poly-*APX *unless* P = NP.

Proof. In order to prove the result, we rely on the reduction introduced in the proof of Theorem 6.5. We have already shown that in a WSPNE $x = (x_n, \nu)$ of $\Gamma_\epsilon(\mathcal{S})$ it holds $c_n^x = \epsilon$ if and only if the corresponding instance of PARTITION admits a *yes* answer. Now, we show that, if the partition problem has *no* answer, then $c_n^x \geq 1$ in any WSPNE. Suppose, by contradiction, that there is a leader's strategy x_n such that all NEs of the resulting followers' game provide the leader with a cost smaller than 1. Then, $x_n^{r_i} < \frac{1}{Y^4}$ for all resources r_i such that $\nu^{r_i} = 3$, $x_n^{r_i} < \frac{1}{Y^4}$ for all resources r_i such that $\nu^{r_i} = 1$, and $x_n^{r_t} < \frac{1}{Y^4}$. If there is a resource r_i with $x_n^{r_i} > w_i + \frac{1}{Y^4}$, we have already proven that there is an NE with $\nu^{r_i} = 3$ providing the leader with a cost greater than $Y^4 x_n^{r_i} > 1$. Consider the set $\mathcal{S}'' \subseteq \mathcal{S}$ of integers y_i corresponding to resources r_i with

$x_n^{r_i} \leq w_i - \frac{1}{Y^4}$. We have already shown that there is an NE with $\nu^{r_i} = 1$ for all $y_i \in \mathcal{S}''$. Since the leader can select these resources with, at most, probability $\frac{1}{Y^4}$ (as $\sum_{y_i \in \mathcal{S}''} x_n^{r_i} \leq \frac{1}{Y^4}$), there is a set $\mathcal{S}'$ of resources r_i with $w_i - \frac{1}{Y^4} < x_n^{r_i} < w_i + \frac{1}{Y^4}$ and $\sum_{y_i \in \mathcal{S}'} x_n^{r_i} \geq 1 - \frac{1}{Y^4}$. From $\sum_{y_i \in \mathcal{S}'}(w_i + \frac{1}{Y^4}) > \sum_{y_i \in \mathcal{S}'} x_n^{r_i} \geq 1 - \frac{1}{Y^4}$, we obtain $\sum_{y_i \in \mathcal{S}'} w_i > 1 - \frac{1}{Y^4} - \frac{|\mathcal{S}'|}{Y^4} > \frac{Y-1}{Y}$. From $\sum_{y_i \in \mathcal{S}'}(w_i - \frac{1}{Y^4}) < \sum_{y_i \in \mathcal{S}'} x_n^{r_i} \leq 1$, we deduce $\sum_{y_i \in \mathcal{S}'} w_i < 1 + \frac{|\mathcal{S}'|}{Y^4} < \frac{Y+1}{Y}$. Thus, $Y - 1 < \sum_{y_i \in \mathcal{S}'} y_i < Y + 1$ and, since $\sum_{y_i \in \mathcal{S}'} y_i$ is an integer quantity, we have that $\sum_{y_i \in \mathcal{S}'} y_i = Y$, showing that $\mathcal{S}'$ is a solution to PARTITION. We have proven that $\Gamma_\epsilon(\mathcal{S})$ admits a WSPNE $x = (x_n, \nu)$ in which $c_n^x = \epsilon$ if and only if the PARTITION instance has a *yes* answer, while, otherwise, $c_n^x \geq 1$ in any WSPNE. Let $\epsilon = \frac{1}{2^{n+r}}$. Assume that there exists a polynomial-time approximation algorithm $\mathcal{A}$ with approximation factor poly(n, r), *i.e.*, a polynomial function of n and r. Assume the PARTITION instance has answer *yes*. $\mathcal{A}$ applied to $\Gamma_\epsilon(\mathcal{S})$ would return a solution with $c_n^x \leq \frac{1}{2^{n+r}}$ poly(n, r). Since, for n and r large enough, $\frac{1}{2^{n+r}}$ poly$(n, r) < 1$, $\mathcal{A}$ would allow us to decide in polynomial time whether the PARTITION instance has a *yes* or *no* answer, a contradiction unless P $=$ NP. □

6.3 Polynomial-Time Algorithms for SSCGs

We have shown that the problem of computing an S/WSPNE in SSCGs is, both in the general case and when restricting ourselves to SSSCGs, computationally intractable. We provide, here, two positive results for SSSCGs, showing that, under certain conditions, the computation of an S/WSPNE in these games can be carried out in polynomial time.

First, we design a polynomial-time algorithm for finding an S/WSPNE in SSSCGs where the players' costs are monotonic functions of the resource congestion. The algorithm relies on the fact that, as we will show, in such games the leader cannot decrease her cost by playing mixed strategies and, thus, pure-strategy commitments are sufficient. We also exhibit a few examples showing that our algorithm cannot be easily extended to more general settings as, if the players have either different action spaces or non-monotonic cost functions, the leader could be better off playing mixed strategies, thus violating the fundamental assumption of our algorithm.

Finally, we show that, if we restrict our attention to pure-strategy commitments in SSSCGs, an S/WSPNE can be found in polynomial time by means of a *dynamic programming* (DP) algorithm, even when the players' cost functions are generic.

6.3.1 Polynomial-Time Algorithms for Computing an S/WSPNE in SSSCGs with Monotonic Cost Functions

Let us recall that, in SSSCGs, an NE minimizing the social cost can be computed in polynomial time (Ieong et al., 2005). It is also easy to show that an NE minimizing/maximizing the cost incurred by one player can be found efficiently using an algorithm similar to that of Ieong et al. (2005) (see Section 6.3.3 for additional details). As a consequence, computing an S/WSPNE would also be easy if an equilibrium could only be induced by a leader's pure-strategy commitment. This is, unfortunately, not the case, as there are SSSCGs admitting S/WSPNEs in which the leader's commitment is a mixed strategy and the followers' configuration could only be induced by the leader committing to a mixed strategy.

Proposition 6.2. *There are SSSCGs with strictly monotonic cost functions which admit an S/WSPNE $x = (x_n, \nu)$ where the leader's strategy x_n is mixed and, additionally, the followers' configuration ν is an NE only for mixed-strategy commitments of the leader.*

Proof. Consider the following SSSCG with strictly monotonic cost functions where $|F| = 3$ and $R = \{r_1, r_2, r_3\}$.

y	$c_{r_1,n}$	$c_{r_1,F}$	$c_{r_2,n}$	$c_{r_2,F}$	$c_{r_3,n}$	$c_{r_3,F}$
1	1	1	3	4	1	1
2	2	3	4	5	2	3
3	3	6	5	6	3	6

The followers configuration $\nu = (1, 1, 1)^\top$ in which each follower selects a different resource is *not* an NE if the leader commits to a pure strategy while, *e.g.*, it is an NE for $x_n^{r_1} = x_n^{r_3} = \frac{1}{2}$ and $x_n^{r_2} = 0$. Notice that the game admits S/WSPNEs in which the leader's commitment is a mixed strategy. For instance, for $x_n^{r_1} = x_n^{r_3} = \frac{1}{2}$ and $x_n^{r_2} = 0$ the leader incurs a cost of 2 and there is no other strategy which allows her to pay less than 2. □

Next, we focus on finding SSPNEs in SSSCGs with weakly monotonic cost functions. We show that, for every strategy profile $x = (x_n, \nu)$ where the leader's commitment x_n is a mixed strategy and ν is an NE in the followers' game, there is another strategy profile $\hat{x} = (\hat{x}_n, \hat{\nu})$ where the leader's commitment $\hat{x}_n$ is a pure strategy with cost no larger than the one for x. In particular, this implies that, in order to achieve an SSPNE, the leader can always commit to a pure strategy. This is formalized in the following theorem, whose proof shows constructively how to build $\hat{x}$ from x.

The idea of the proof is the following. Let $x = (x_n, \nu)$ be a strategy profile where x_n is a mixed strategy and ν is an NE in the followers' game. Assume that the leader switched to selecting with probability one any of the resources for which she would incur the minimum cost when committing to x_n. As this would increase the congestion of that resource, due to the cost functions being weakly monotonic the followers could only react by switching to another resource—this translates in the leader incurring a cost on that resource which is never larger than the one she would incur when committing to the mixed strategy x_n.

Theorem 6.7. *Every SSSCG with weakly monotonic cost functions admits an SSPNE $x = (x_n, \nu)$ in which x_n is pure.*

Proof. Given a strategy profile $x = (x_n, \nu)$ with x_n mixed and $\nu \in \mathcal{E}(x_n)$, we show how to construct another strategy profile $\hat{x} = (\hat{x}_n, \hat{\nu})$ with $\nu \in \mathcal{E}(\hat{x}_n)$ in which $\hat{x}_n$ is pure and $c_n^{\hat{x}} \leq c_n^x$. Let $\mathcal{S} = \{i \in R \mid x_n^i > 0\}$ be the set of resources played by the leader with positive probability in x_n and let $i^\star \in \arg\min_{i \in \mathcal{S}} c_{i,n}(\nu^i + 1)$. Clearly, since the leader's utility is a convex combination weighted by x_n of the costs she incurs in the resources chosen with positive probability, $c_n^x = \sum_{i \in A_n} x_n^i c_{i,n}(\nu^i + 1) \geq c_{i^\star,n}(\nu^{i^\star} + 1)$. Moreover, since ν is an NE for x_n, the following holds by definition:

$$c_{i,F}^{x_n}(\nu^i) \leq c_{j,F}^{x_n}(\nu^j + 1) \;\; \forall\, i \in R : \nu^i > 0,\; j \in R. \tag{6.1}$$

Let us define $\hat{x}_n \in \Delta_n$ such that $\hat{x}_n^{i^\star} = 1$. We now show that such $\hat{x}_n$ is part of an SSPNE. Notice that $c_{i,F}^{\hat{x}_n}(y) = c_{i,F}(y)$ for all $y \in \mathbb{N}$ and $i \in R \setminus \{i^\star\}$ (as the leader does not select these resources), while $c_{i^\star,F}^{\hat{x}_n}(y) = c_{i^\star,F}(y+1)$ for all $y \in \mathbb{N}$ (as the leader selects that resource). Since, in the strong case, the followers behave in favor of the leader, it is sufficient to exhibit a $\hat{\nu} \in \mathcal{E}(\hat{x}_n)$ such that $\hat{x} = (\hat{x}_n, \hat{\nu})$ satisfies $c_n^{\hat{x}} \leq c_n^x$. We construct a sequence of followers configurations which starts from ν and reaches such $\hat{\nu}$. Given $\hat{x}_n$, let us consider the sequence $(\nu(0) = \nu, \nu(1), \ldots, \nu(T) = \hat{\nu})$ such that each configuration differs from the previous one in that a single follower has changed resource, strictly decreasing her cost. Formally, this corresponds to showing that, for all $0 \leq t < T$, there is a pair $i, j \in R$ such that $\nu(t)^i > 0$, $\nu(t+1)^i = \nu(t)^i - 1$, $\nu(t+1)^j = \nu(t)^j + 1$, and $c_{i,F}^{\hat{x}_n}(\nu(t)^i) > c_{j,F}^{\hat{x}_n}(\nu(t+1)^j)$. Moreover, let us assume that a follower deviates to resource $i^\star$, *i.e.*, $\nu(t+1)^{i^\star} > \nu(t)^{i^\star}$, only if this is the only way of strictly decreasing some follower's cost. This is w.l.o.g., as it is consistent with the assumption that the followers break ties in favor of the leader. Let us now prove that the sequence of followers' configurations satisfies the following:

$$\nu(t+1)^{i^\star} \leq \nu(t)^{i^\star} \;\; \forall\, 0 \leq t < T. \tag{6.2}$$

By contradiction, assume there exists $0 \leq t < T$ such that $\nu(t+1)^{i^\star} > \nu(t)^{i^\star}$. Then, there is a follower who can strictly decrease her cost in $\nu(t)$ by choosing $i^\star$ instead of some $j \neq i^\star \in R : \nu(t)^j > 0$, *i.e.*, $c^{\hat{x}_n}_{i^\star,F}(\nu(t)^{i^\star}+1) < c^{\hat{x}_n}_{j,F}(\nu(t)^j)$ holds. Thus, given that $c^{\hat{x}_n}_{i^\star,F}(\nu(t)^{i^\star}+1) = c_{i^\star,F}(\nu(t)^{i^\star}+2)$ and $c^{\hat{x}_n}_{j,F}(\nu(t)^j) = c_{j,F}(\nu(t)^j)$, we conclude that:

$$c^{x_n}_{i^\star,F}(\nu^{i^\star}+1) \leq c_{i^\star,F}(\nu(t)^{i^\star}+2) < c_{j,F}(\nu(t)^j), \tag{6.3}$$

where the first inequality holds since $\nu(t)^{i^\star} = \nu^{i^\star}$ (as Equation (6.2) holds for the elements of the sequence preceding $\nu(t)$ and the number of followers selecting $i^\star$ cannot decrease with respect to $\nu^{i^\star}$). Two cases are possible. In the first one, $\nu(t)^j \leq \nu^j$, implying, by monotonicity, $c_{j,F}(\nu(t)^j) \leq c_{j,F}(\nu^j) \leq c^{x_n}_{j,F}(\nu^j)$, which, together with Equations (6.1) and (6.3), leads to a contradiction. In the second case, $\nu(t)^j > \nu^j$ implies that there exists $k \neq i^\star \in R$ such that $\nu(t)^k < \nu^k$ (and $\nu^k > 0$), otherwise $\sum_{i\in R} \nu(t)^i > n-1$. It follows that $c_{j,F}(\nu(t)^j) \leq c_{k,F}(\nu(t)^k+1) \leq c^{x_n}_{k,F}(\nu^k)$, where the first inequality holds since, due to our assumptions on the sequence, it cannot be $c_{j,F}(\nu(t)^j) > c_{k,F}(\nu(t)^k+1)$ as $\nu(t+1)^{i^\star} > \nu(t)^{i^\star}$, and the second inequality follows from $\nu(t)^k < \nu^k$. Thus, Equations (6.1) and (6.3) give a contradiction. As a result, Equation (6.2) holds, and, thus, $\hat{\nu}^{i^\star} \leq \nu^{i^\star}$. Given the monotonicity of the costs, we conclude that $c^{\hat{x}}_n \leq c^x_n$. The prove the claim, it now suffices to take as strategy profile $x = (x_n, \nu)$ an SSPNE in which x_n is mixed—since the leader's cost at x is the smallest possible, her cost at $\hat{x}$ will be identical to it. □

We prove, now, that a similar result holds for the weak case, *i.e.*, for computing a WSPNE. The result is weaker though, as it requires the stronger assumption that the followers' cost functions be strictly monotonic.

The idea of the proof is similar to the previous one. Given a WSPNE $x = (x_n, \nu)$ in which x_n is a mixed strategy, we show that there exists another WSPNE $\hat{x} = (\hat{x}_n, \hat{\nu})$ where the leader's commitment $\hat{x}_n$ is a pure strategy which selects with probability one any of the resources for which the leader incurs the minimum cost when committing to x_n. In order to show this, we prove, by contradiction, that any NE $\hat{\nu}$ for $\hat{x}_n$ provides the leader with a cost smaller than or equal to the one for x.

Theorem 6.8. *Every SSSCG in which the leader's and followers' cost functions are, respectively, weakly and strictly monotonic, admits a WSPNE $x = (x_n, \nu)$ in which x_n is pure.*

Proof. Assume there exists a WSPNE $x = (x_n, \nu)$ in which x_n is mixed. We show that there must be another WSPNE $\hat{x} = (\hat{x}_n, \hat{\nu})$ such that $\hat{x}_n$ is pure. Let us define $i^\star \in R$ and $\hat{x}_n \in \Delta_n$ as in the proof of Theorem 6.7, so that $c_n^x \geq c_{i^\star,n}(\nu^{i^\star} + 1)$ and Equation (6.1) holds. Given that the followers behave pessimistically, we need to show that, for every $\hat{\nu} \in \mathcal{E}(\hat{x}_n)$, $\hat{x} = (\hat{x}_n, \hat{\nu})$ satisfies $c_n^{\hat{x}} \leq c_n^x$. By contradiction, assume $c_n^{\hat{x}} > c_n^x$, which implies $c_{i^\star,n}(\hat{\nu}^{i^\star} + 1) > c_{i^\star,n}(\nu^{i^\star} + 1)$. It easily follows from the monotonicity of the costs that $\hat{\nu}^{i^\star} > \nu^{i^\star}$. Thus, there must be a resource $j \in R$ such that $\hat{\nu}^j < \nu^j$ as, otherwise, $\sum_{i \in R} \hat{\nu}^i > n - 1$. Let us also remark that $\nu^j > 0$. Thus:

$$c_{i^\star,F}^{x_n}(\nu^{i^\star} + 1) \leq c_{i^\star,F}(\hat{\nu}^{i^\star} + 1) \leq c_{j,F}(\hat{\nu}^j + 1) \leq c_{j,F}^{x_n}(\nu^j), \tag{6.4}$$

where the first inequality follows from $\nu^{i^\star} < \hat{\nu}^{i^\star}$, the second one from the fact that $\hat{\nu}$ is an NE for $\hat{x}_n$, and the third one from $\hat{\nu}^j < \nu^j$. Equation (6.1) implies $c_{j,F}^{x_n}(\nu^j) \leq c_{i^\star,F}^{x_n}(\nu^{i^\star} + 1)$. If $c_{j,F}^{x_n}(\nu^j) < c_{i^\star,F}^{x_n}(\nu^{i^\star} + 1)$, then Equation (6.4) leads to a contradiction. Otherwise, if $c_{j,F}^{x_n}(\nu^j) = c_{i^\star,F}^{x_n}(\nu^{i^\star} + 1)$ all the inequalities in Equation (6.4) hold as equations. This, however, implies $c_{i^\star,F}^{x_n}(\nu^{i^\star} + 1) = c_{i^\star,F}(\hat{\nu}^{i^\star} + 1)$ and $c_{j,F}(\hat{\nu}^j + 1) = c_{j,F}^{x_n}(\nu^j)$, which is a contradiction since x_n is mixed and the followers' cost functions are strictly monotonic. □

Theorem 6.8 fails to hold if the followers' cost functions are weakly, rather than strictly, monotonic, as the following result shows:

Proposition 6.3. *There are SSSCGs with weakly monotonic cost functions where any WSPNE prescribes the leader to play a mixed strategy.*

Proof. Consider the following instance of SSSCG with weakly monotonic cost functions, where $|F| = 1$ and $R = \{r_1, r_2\}$.

y	$c_{r_1,n}$	$c_{r_1,F}$	$c_{r_2,n}$	$c_{r_2,F}$
1	1	1	1	1
2	2	1	2	1

Clearly, any followers' configuration is an NE in this game, independently of the leader's commitment. Whenever the leader commits to a pure strategy, be it the selection of r_1 or r_2, the follower, due to the pessimistic assumption, chooses the same resource, so to have the leader incur a cost as large as possible (of 2). By uniformly randomizing between the two resources, though, the leader can reduce her cost to $2\frac{1}{2} + \frac{1}{2} = 1.5$. □

Relying on Theorems 6.7 and 6.8, we can compute an SSPNE (respectively, WSPNE) by enumerating the leader's pure strategies and, for each of them, computing a followers' NE which results in the smallest (respectively, largest) leader's cost. Such NE can be computed by applying a simple greedy procedure which progressively assigns followers to resources. At each step, a single follower is assigned to the resource which is cheapest for her, given how the previously considered followers have been distributed over the resources. At a given step, among all the resources minimizing followers' cost the procedure selects one minimizing (respectively, maximizing) the leader's cost. An S/WSPNE is then obtained by picking any leader's pure strategy for which the leader's cost is the smallest.

The detailed procedure is described in Algorithm 6.1 where, for some $\mathcal{S} \subseteq R$ and $i \in \mathcal{S}$, the function S-PICK$(\mathcal{S}, i)$ (respectively, W-PICK$(\mathcal{S}, i)$) returns some resource $j^\star \in \mathcal{S}$, giving precedence to resources $j^\star \neq i$ (respectively, $j^\star = i$).

Algorithm 6.1 Algorithm for computing an S/WSPNE in SSSCG with monotonic costs.

function COMPUTE-S/W-SPNE
    for all $i \in R$ **do**
        $x_n[i] \leftarrow x_n \in \Delta_n : x_n^i = 1$
        $\nu[i,j] \leftarrow 0 \ \forall\, i,j \in R$
        while $\sum_{j \in R} \nu[i,j] < n$ **do**
            $\mathcal{S} \leftarrow \arg\min_{j \in R} \ c_{j,F}^{x_n[i]}(\nu[i,j]+1)$
            $j^\star \leftarrow$ S/W-PICK$(\mathcal{S}, i)$
            $\nu[i,j^\star] \leftarrow \nu[i,j^\star]+1$
        end while
        $c_n[i] \leftarrow c_{i,n}(\nu[i,i]+1)$
    end for
    $i^\star \leftarrow \arg\min_{i \in R} c_n[i]$
    return $x = (x_n[i^\star], \nu[i^\star, \cdot])$
end function

Let us remark that, in Algorithm 6.1, $x_n[\cdot]$, $\nu[\cdot,\cdot]$, and $c_n[\cdot]$ are the algorithm's variables and, for every $i \in R$, $\nu[i,j]$ denotes the number of followers selecting resource $j \in R$ in the NE which is reached when the leader's strategy is $x_n[i]$.

Theorem 6.9. *Algorithm 6.1 is correct and it runs in time $O(nr \log r)$.*

Proof. We rely on the pseudocode reported in Algorithm 6.1 to show its correctness. Thanks to Theorems 6.7 and 6.8, we only need to prove that, for every $i \in R$ and after the execution of the while loop, the followers configuration ν is such that, for all $j \in R$, $\nu^j = \nu[i,j]$ is an NE for $x_n[i]$ minimizing (or maximizing) the leader's cost. First, let us show that ν is an NE. Suppose, by contradiction, that it is not. Then, there exists $j \in$

$R : \nu_j > 0$ and $k \in R$ such that $c^{x_n[i]}_{j,F}(\nu^j) > c^{x_n[i]}_{k,F}(\nu^k + 1)$. Let $\bar{\nu}^k$ be the value of $\nu[i,k]$ during the step in which $\nu[i,j]$ is set to its final value ν^j. Clearly, $c^{x_n[i]}_{j,F}(\nu^j) > c^{x_n[i]}_{k,F}(\nu^k + 1) \geq c^{x_n[i]}_{k,F}(\bar{\nu}^k + 1)$, and the algorithm would have not incremented $\nu[i,j]$ during that step, a contradiction. Let us show now that $(x_n[i], \nu)$ is an S/WSPNE. In the remainder of the proof, we focus on the strong case (the weak one can be treated analogously). Suppose, by contradiction, that ν is not an NE minimizing the leader's cost for $x_n[i]$ (*i.e.*, not an SSPNE). Then, there exists another NE $\hat{\nu}$ for $x_n[i]$ such that $c_{i,n}(\hat{\nu}^i + 1) < c_{i,n}(\nu^i + 1)$. Given the monotonicity of the costs, $\hat{\nu}^i < \nu^i$ must hold. Therefore, there must exist some $j \neq i \in R$ such that $\hat{\nu}^j > \nu^j$. Let us consider the step in which $\nu[i,i]$ is set to ν^i, and let $\bar{\nu}^j$ be the value of $\nu[i,j]$ during that step. Note that $c^{x_n[i]}_{i,F}(\nu^i) < c^{x_n[i]}_{j,F}(\bar{\nu}^j + 1)$ must hold as, otherwise, the algorithm would have incremented $\nu[i,j]$ instead of $\nu[i,i]$. But, then, $c^{x_n[i]}_{j,F}(\bar{\nu}^j + 1) \leq c^{x_n[i]}_{j,F}(\nu^j + 1) \leq c^{x_n[i]}_{j,F}(\hat{\nu}^j)$, which implies $c^{x_n[i]}_{i,F}(\hat{\nu}^i + 1) \leq c^{x_n[i]}_{i,F}(\nu^i) < c^{x_n[i]}_{j,F}(\bar{\nu}^j + 1) \leq c^{x_n[i]}_{j,F}(\hat{\nu}^j)$, contradicting the fact that $\hat{\nu}$ is an NE for the given $x_n[i]$.

Since the while loop is executed exactly r times, each execution carries out n steps. Using efficient data structures, each step takes time $O(\log r)$. Thus, the overall running time is $O(nr \log r)$. $\square$

Next, we provide a characterization of S/WSPNEs in SSSCGs with monotonic costs under the additional assumption that leader's and followers' costs be equal, which may be of independent interest besides the computation of S/WSPNEs.

Theorem 6.10. *Given an SSSCG with monotonic costs and* $c_n = c_F = \{c_i\}_{i \in R}$*, any S/WSPNE* $x = (x_n, \nu)$ *with* x_n *pure is an NE.*

Proof. Let $x = (x_n, \nu)$ be an S/WSPNE with $x^{i^\star}_n = 1$ for some $i^\star \in R$. Clearly, given that $\nu \in \mathcal{E}(x_n)$, $c^{x_n}_i(\nu^i) \leq c^{x_n}_j(\nu^j + 1)$ holds for every $i \in R : \nu^i > 0$ and for every $j \in R$. Therefore, no follower has an incentive to change resource. Thus, it is sufficient to prove that the leader has no incentive to deviate from resource $i^\star$ unilaterally, *i.e.*, without assuming that the followers would react to her deviation (which is the case in the Stackelberg setting). If $\nu^{i^\star} > 0$, we have $c_{i^\star}(\nu^{i^\star}+1) = c^{x_n}_{i^\star}(\nu^{i^\star}) \leq c^{x_n}_j(\nu^j + 1) = c_j(\nu^j + 1)$ for every $j \neq i^\star \in R$, and it immediately follows that the leader does not deviate and x is an NE. The case in which $\nu^{i^\star} = 0$ is more involved. By contradiction, assume that x is not an NE. As a consequence, the leader must have an incentive to deviate to some resource $j \neq i^\star \in R$, *i.e.*, $c_{i^\star}(\nu^{i^\star} + 1) = c_{i^\star}(1) > c_j(\nu^j + 1)$. Let $\hat{x}_n$ with $\hat{x}^j_n = 1$ be the strategy the leader commits to. We prove (by contradiction) that, for every $\hat{\nu} \in \mathcal{E}(\hat{x}_n)$, $\hat{x} = (\hat{x}_n, \hat{\nu})$ provides the leader with a cost strictly smaller than $c_{i^\star}(1)$. Assume $c_j(\hat{\nu}^j + 1) \geq c_{i^\star}(1)$. Three cases are possible. In the first one, $\hat{\nu}^j < \nu^j$ and $c_{i^\star}(1) > c_j(\nu^j + 1) \geq c_j(\hat{\nu}^j + 1) \geq c_{i^\star}(1)$. In the second one, $\hat{\nu}^j = \nu^j$ and $c_j(\hat{\nu}^j + 1) \geq c_{i^\star}(1) > c_j(\nu^j + 1)$. In the third case, $\hat{\nu}^j > \nu^j$, which implies that there must be a resource $k \neq i^\star \in R$ such that $\hat{\nu}^k < \nu^k$, and $c_{i^\star}(1) > c_j(\nu^j + 1) \geq c_k(\nu^k) \geq c_k(\hat{\nu}^k + 1) \geq c_j(\hat{\nu}^j + 1) \geq c_{i^\star}(1)$. As all the cases lead to a contradiction, it must be $c_j(\hat{\nu}^j + 1) < c_{i^\star}(1)$. The proof is complete as, in $\hat{x}$, the leader's cost is $c_j(\hat{\nu}^j + 1) < c_{i^\star}(1)$, contradicting the fact that x is an S/WSPNE. □

6.3.2 On the Necessity of the Assumptions

We provide some examples showing why Algorithm 6.1 cannot be easily extended to more general settings—the reason being that Theorems 6.7 and 6.8 do not hold if the assumption of monotonicity is dropped.

First, let us analyze the general case of SSSCGs in which the costs need not be monotonic functions of the resource congestion:

Proposition 6.4. *There are SSSCGs in which, even if the cost functions of one player only are nonmonotonic, be it the leader or one of the followers, any S/WSPNE prescribes the leader to play a mixed strategy.*

Proof. Consider the following instance of SSSCG with non-monotonic followers' cost functions, where $R = \{r_1, r_2\}$ and $|F| = 1$.

y	$c_{r_1,n}$	$c_{r_1,F}$	$c_{r_2,n}$	$c_{r_2,F}$
1	1	2	1	2
2	2	1	2	1

The follower selects r_2 whenever $x_{r_1} \leq \frac{1}{2}$, while, if $x_n^{r_1} \geq \frac{1}{2}$, she chooses r_1. The leader's cost is $2 - x_n^{r_1}$ if $x_n^{r_1} \leq \frac{1}{2}$, and $1 + x_n^{r_1}$ if $x_n^{r_1} \geq \frac{1}{2}$. There is, thus, a unique S/WSPNEE that prescribes the leader to commit to x_n with $x_n^{r_1} = x_n^{r_2} = \frac{1}{2}$.

Consider now the following instance of SSSCG with non-monotonic leader's cost functions, where $R = \{r_1, r_2\}$ and $|F| = 1$.

y	$c_{r_1,n}$	$c_{r_1,F}$	$c_{r_2,n}$	$c_{r_2,F}$
1	2	1	2	1
2	0	2	0	2

The follower selects r_2 if $x_n^{r_1} \geq \frac{1}{2}$ and r_1 if $x_n^{r_1} \leq \frac{1}{2}$. The leader's cost is thus $2x_n^{r_1}$ if $x_n^{r_1} \geq \frac{1}{2}$ and $2-2x_n^{r_1}$ if $x_n^{r_1} \leq \frac{1}{2}$. There is, thus, a unique O/PSE which prescribes the leader to commit to x_n with $x_n^{r_1} = x_n^{r_2} = \frac{1}{2}$. □

Finally, we show that Theorems 6.7 and 6.8 do not hold for general SSCGs with different action spaces, even if the cost functions are monotonic:

Proposition 6.5. *There are SSCGs with different action spaces and monotonic cost functions where any S/WSPNE prescribes the leader to play a mixed strategy.*

Proof. Consider the following SSCG with $R = \{r_1, r_2, r_3\}$, two followers $F = \{p_1, p_2\}$, and $A_{p_1} = \{r_1, r_2\}$, $A_{p_2} = \{r_2, r_3\}$, $A_n = \{r_1, r_2\}$:

y	$c_{r_1,F}$	$c_{r_1,n}$	$c_{r_2,F}$	$c_{r_2,n}$	$c_{r_3,F}$
1	1	0	0	1	3
2	1	1	2	1	3
3	1	1	4	1	3

If the leader plays $x_n^{r_1} = 1$, there is a unique NE where follower p_1 plays r_1 and follower p_2 plays r_2. Indeed, p_2 incurs a cost of 0 and, thus, has no incentive to deviate, while p_1 would incur a cost of $2 > 1$ by deviating to r_2. Thus, the leader's cost is 1. The leader's cost is also 1 if she played $x_n^{r_2} = 1$, as p_2 would also choose r_2, while p_1 would choose r_1. Let us show that the leader can commit to a mixed strategy and incur a cost smaller than 1. Indeed, with $x_n^{r_1} = x_n^{r_2} = \frac{1}{2}$, there is a followers' NE where p_1 chooses r_2 and p_2 chooses r_3: p_1, incurring a cost of 1 (smaller or equal than any other cost), has no incentive to deviate, while p_2, currently incurring a cost of 3, by switching to r_2 would incur the same (expected) cost of 3 (i.e., a cost of 2 with probability $\frac{1}{2}$ and one of 4 with probability $\frac{1}{2}$), thus having no incentive to deviate. At that NE, the leader's cost is $0 \cdot \frac{1}{2} + 1 \cdot \frac{1}{2} = \frac{1}{2}$. □

6.3.3 Pure-Strategy Commitment in SSSCGs with Generic Costs

We propose, here, a simple polynomial-time algorithm for computing an S/WSPNE in SSSCGs with generic costs where the leader is restricted to pure-strategy commitments. It is based on a dynamic programming algorithm proposed in (Ieong et al., 2005) for the computation of an optimal NE in symmetric non-Stackelberg singleton congestion games. The original algorithm runs in $O(n^6 r^5)$. One can compute an S/WSPNE in r iterations, fixing, at each iteration, the action the leader would choose and calling the previous algorithm to compute an NE which either minimizes or maximizes the leader's cost. This takes, overall, $O(n^6 r^6)$.

We show, in the following, how to improve the complexity of the original algorithm to $O(n^4 r^3)$, thanks to which we can compute an S/WSPNE for the restricted case in $O(n^4 r^4)$. The algorithm is based on the same recursive formula shown in (Ieong et al., 2005), which we reintroduce, here, in a different and, possibly, clearer way.

Let $\mathcal{O}(h, B, M, V)$ be the cost of an optimal NE for a symmetric SCG without leadership restricted to h resources $\{1, 2, ..., h\} \subseteq R$ and B players, where M is the largest cost incurred by a player and V is the smallest cost a player would incur if she were to switch to another resource.

Proposition 6.6. *$\mathcal{O}(h, B, M, V)$ satisfies the following recursive equation:*

$$\mathcal{O}(h, B, M, V) = \min_{\substack{p \in \{0,...,B\} \\ m \in \mathbb{Z}^+,\, v \in \mathbb{Z}^+}} \mathcal{O}(h-1, p, m, v) + (B-p)\, c_h(B-p)$$

$$\begin{aligned} \text{s.t.} \quad & m \leq M && \text{(6.5a)} \\ & v \geq V && \text{(6.5b)} \\ & c_h(B-p) \leq M && \text{(6.5c)} \\ & c_h(B-p+1) \geq V && \text{(6.5d)} \\ & c_h(B-p) \leq v && \text{(6.5e)} \\ & c_h(B-p+1) \geq m. && \text{(6.5f)} \end{aligned}$$

Proof. We show that all the constraints are necessary for the definition of $\mathcal{O}(h, B, M, V)$ to be respected. If Constraint (6.5a) were not satisfied, $m > M$ would imply that there is at least a resource among those in $\{1, \ldots, h-1\}$ costing strictly more than M. If Constraint (6.5b) were not satisfied, $v < V$ would imply that the cost to deviate to a resource among those in $\{1, \ldots, h-1\}$ is strictly smaller than V. If Constraint (6.5c) were not satisfied, $c_h(B-p) > M$ would imply that M is smaller than the cost of the most expensive chosen resource. If Constraint (6.5d) were not satisfied, $c_h(B-p+1) < V$ would imply that V is larger than the cheapest cost a player would incur upon deviating to another resource. If Constraint (6.5e) were not satisfied, $c_h(B-p) > v$ would imply that each of the $B-p$ players who chose resource h would have an incentive to deviate to any of the resources in $\{1, \ldots, h-1\}$. If Constraint (6.5f) were not satisfied, $c_h(B-p+1) < m$ would imply that at least one of the p players who selected a resource in $\{1, \ldots, h-1\}$ (*i.e.*, all those incurring a cost of m) would have an incentive to deviate to resource h. $\square$

We now show how to simplify the recursive formula for $\mathcal{O}(h, B, M, V)$:

Theorem 6.11. *$\mathcal{O}(h, B, M, V)$ satisfies the following recursive equation:*

$$\begin{aligned}
&\mathcal{O}(h, B, M, V) = \\
&\quad = \min_{p \in \{0,\ldots,B\}} \mathcal{O}(h-1, p, m(p)^*, v(p)^*) + (B-p)\, c_h(B-p) \\
&\qquad\qquad \text{s.t.} \quad c_h(B-p) \leq M && \text{(6.6a)} \\
&\qquad\qquad\qquad\;\; c_h(B-p+1) \geq V, && \text{(6.6b)}
\end{aligned}$$

where $m(p)^* = \min\{M, c_h(B-p+1)\}$ and $v(p)^* = \max\{V, c_h(B-p)\}$.

Proof. Constraints (6.5a)–(6.5f) and (6.5b)–(6.5e) imply, respectively, $m \leq \min\{M, c_h(B-p+1)\}$ and $v \geq \max\{V, c_h(B-p)\}$. Hence, $m(p)^*$ and $v(p)^*$ are feasible for Problem (6.5). Notice that, if $m' > m$ and $v' < v$, the feasible region underlying $\mathcal{O}(h, p, m', v')$ contains the one underlying $\mathcal{O}(h, p, m, v)$, which implies $\mathcal{O}(h, p, m', v') \leq \mathcal{O}(h, p, m, v)$. The claim follows since $m(p)^*$ and $v(p)^*$ are, respectively, the largest and smallest values m and v can take. □

Corollary 6.11.1. *In symmetric non-Stackelberg singleton congestion games, an optimal NE can be found in $O(n^4r^3)$. In SSSCGs with the leader restricted to pure strategies, an S/WSPNE can be found in $O(n^4r^4)$.*

Proof. Since there are at most nr different values of $c_j(y)$, for all $j \in R$ and $y \in \mathbb{N}$, there are at most nr values of M and at most nr values of V. There are also exactly r values of h and exactly n of B. Hence, the dynamic programming table of $\mathcal{O}(h, B, M, V)$ contains $O(n^3r^3)$ entries. Due to Theorem (6.11), computing an entry of the table requires $O(n)$. Overall, an optimal NE is computed in $O(n^4r^3)$. For the case with leadership restricted to pure strategies, it suffices to run the algorithm for each resource the leader may choose, obtaining a complexity of $O(n^4r^4)$. □

6.4 Results on Non-singleton SCGs

We now study the problem of computing an SSPNE in general SCGs with non-singleton actions, proving it is intractable even when the game is symmetric, cost functions are monotonic, and players' actions are made of only two resources. Thus, non-singleton actions make the problem considerably harder than in the singleton case, which admits a polynomial-time algorithm in symmetric games with monotonic costs (see Theorem 6.9).

In particular, we show that the problem is NP-hard and *not* in Poly-APX, unless P = NP, by means of a reduction from 3-SAT. Intuitively, given $\epsilon > 0$, we map any 3-SAT instance to an SCG that admits an SSPNE x with $c_n^x = \epsilon$ if and only if 3-SAT is satisfiable, otherwise $c_n^x = 1$.

Theorem 6.12. *Computing an SSPNE in symmetric SCGs is* NP*-hard, even when cost functions are monotonic and players' actions have cardinality at most two.*

Proof. We provide a reduction from 3-SAT showing that a polynomial-time algorithm for finding an SSPNE in SCGs would allow us to solve any 3-SAT instance in polynomial time. Given a 3-SAT instance (C, V) and a number $0 < \epsilon < 1$, we build an SCG $\Gamma_\epsilon(C, V)$ admitting an SSPNE x with $c_n^x = \epsilon$ if and only if (C, V) is satisfiable.

Mapping. $\Gamma_\epsilon(C, U)$ is defined as follows:

- $N = F \cup \{n\}$ with $F = \{p_v \mid v \in V\} \cup \{p_\phi \mid \phi \in C\}$;
- $R = \{r_w\} \cup \{r_v, r_{\bar{v}}, r_{v,t} \mid v \in V\} \cup \{r_\phi \mid \phi \in C\}$;
- $A_p = \{a_v = \{r_v, r_{v,t}\}, a_{\bar{v}} = \{r_{\bar{v}}, r_{v,t}\} \mid v \in V\} \cup \{a_w = \{r_w\}\} \cup \{a_{\phi,l} = \{r_\phi, r_l\} \mid \phi \in C, l \in \phi\}$ for all $p \in N$.

Cost functions are specified in the following table, and, additionally, $c_{r_{\bar{v}},F} = c_{r_v,F}$ and $c_{i,n} = c_{i,F}$ for all $i \in R$.

y	$c_{r_\phi,F}$	$c_{r_v,F}$	$c_{r_{v,t},F}$	$c_{r_w,F}$
1	1	0	3	ϵ
$[2, \lvert C\rvert + \lvert V\rvert + 1]$	5	2	5	4

Clearly, $\Gamma_\epsilon(C, V)$ can be constructed in polynomial time, since $n = |C| + |V| + 1$, $r = |C| + 3|V| + 1$, and $|A_p| = 3|C| + 2|V| + 1$ for $p \in N$. We remark that $\Gamma_\epsilon(C, V)$ is symmetric, cost functions are monotonic, and each action has cardinality at most two. Moreover, the leader's cost is ϵ if and only if she is the only player using the singleton action a_w, otherwise her cost is at least 1, since other actions contain two resources.

If. Suppose that (C, V) is satisfiable, and let $T : V \mapsto \{0, 1\}$ be a truth assignment satisfying all clauses in C. Let $x_n \in \Delta_n : x_n^{a_w} = 1$. Using T, we can build $x = (x_n, a_{-n})$, with $a_{-n} \in \mathcal{E}(x_n)$, such that $c_n^x = \epsilon$. Since ϵ is the minimum leader's cost and the followers behave in favor of the leader, x is an SSPNE. In particular, for every $\phi \in C$, there must be a follower $p \in F$ such that $a_p = a_{\phi,l}$, where $l \in \phi$ evaluates to true under T. Clearly, one such literal $l \in \phi$ always exists. When there are many, take one minimizing

$\nu^{r_l}_{a_{-n}}$. Moreover, for every $v \in V$, if $T(v) = 1$, respectively $T(v) = 0$, there must be a follower $p \in F$ such that $a_p = a_{\bar{v}}$, respectively $a_p = a_v$. Thus, $\nu^{r_\phi}_{a_{-n}} = 1$ for all $\phi \in C$, and, similarly, $\nu^{r_{v,t}}_{a_{-n}} = 1$ for all $v \in V$. Additionally, $\nu^{r_w}_{a_{-n}} = 0$ as there are $|C| + |V|$ followers. Next, we show that $a_{-n} \in \mathcal{E}(x_n)$. First, followers $p \in F$ with $a_p = a_{\phi,l}$ experience a cost $c^x_p = c_{r_\phi,F}(\nu^{r_\phi}_{a_{-n}}) + c_{r_l,F}(\nu^{r_l}_{a_{-n}}) \leq 3$, since $\nu^{r_\phi}_{a_{-n}} = 1$. Thus, they do not have any incentive to deviate. If they switch to $a_{\phi',l'}$ (with $\phi' \neq \phi$), then they would pay at least 5 (as $\nu^{r_{\phi'}}_{a_{-n}} = 1$). Furthermore, they do not deviate to $a_{\phi,l'}$ (with $l' \neq l \in \phi$), as, if l' is false, then they would pay 3, while, when l' is true, they would incur a cost of $c_{r_\phi,F}(\nu^{r_\phi}_{a_{-n}}) + c_{r_{l'},F}(\nu^{r_{l'}}_{a_{-n}}) \geq c^x_p$ (as $\nu^{r_{l'}}_{a_{-n}} \geq \nu^{r_l}_{a_{-n}}$). If, instead, they deviate to a_v or $a_{\bar{v}}$, then their cost would be at least 5 (as $\nu^{r_{v,t}}_{a_{-n}} = 1$). Moreover, they do not switch to a_w, since they would pay 4. Followers $p \in F$ with $a_p = a_v$ has cost $c^x_p = c_{r_v,F}(\nu^{r_v}_{a_{-n}}) + c_{r_{v,t},F}(\nu^{r_{v,t}}_{a_{-n}}) = 3$ since $\nu^{r_v}_{a_{-n}} = 1$. Thus, they do not deviate, as they would pay at least 4. Similarly, followers $p \in F$ with $a_p = a_{\bar{v}}$ do not deviate. As a result, a_{-n} is an NE and, since $x^{a_w}_n = 1$ and $\nu^{r_w}_{a_{-n}} = 0$, it holds $c^x_n = \epsilon$.

Only if. Suppose there exists an SSPNE $x = (x_n, a)$ such that $c^x_n = \epsilon$. Thus, $x^{a_w}_n = 1$ and $\nu^{r_w}_{a_{-n}} = 0$. For $v \in V$, $\nu^{r_{v,t}}_{a_{-n}} \leq 1$, otherwise, if $\nu^{r_{v,t}}_{a_{-n}} \geq 2$, some followers would have an incentive to deviate to action a_w, paying $4 < 5$. Analogously, for $\phi \in C$, $\nu^{r_\phi}_{a_{-n}} \leq 1$. Since there are $|C| + |V|$ followers, $\nu^{r_{v,t}}_{a_{-n}} = 1$ for every $v \in V$, and $\nu^{r_\phi}_{a_{-n}} = 1$ for every $\phi \in C$. Thus, for every $v \in V$, there exists $p \in F$ such that either $a_p = a_v$ or $a_p = a_{\bar{v}}$, and no other follower selects actions a_v and $a_{\bar{v}}$. Define a truth assignment T such that $T(u) = 1$ if there is $p \in F$ with $a_p = a_{\bar{v}}$, while $T(v) = 0$ if there is $p \in F$ with $a_p = a_v$. Clearly, T is well-defined. Moreover, for every $\phi \in C$, there exists a unique follower $p \in F$ and a literal $l \in \phi$ such that $a_p = a_{\phi,l}$, as $\nu^{r_\phi}_{-n} = 1$. This implies that no follower plays a_l, otherwise her cost would be at least 5, and she would deviate to a_w, paying 4. Thus, if l is positive, there is $p \in F$ with $a_p = a_{\bar{v}}$, while, if it is negative, there is $p \in F$ with $a_p = a_v$. Therefore, T satisfies all clauses. $\square$

By letting $\epsilon = \frac{1}{2^I}$, where I is the game size, the reduction used for Theorem 6.12 also shows the following:

Theorem 6.13. *The problem of computing an SSPNE in symmetric SCGs is not in Poly-*APX *unless* P = NP*, even when costs are monotonic and players' actions have cardinality two.*

Proof. Given a 3-SAT instance (C, V), we build an SCG $\Gamma_\epsilon(C, V)$ as in the proof of Theorem 6.12. As previously shown, in an SSPNE x of $\Gamma_\epsilon(C, U)$, it holds $c^x_n = \epsilon$ if and only if (C, V) is satisfiable. Next, we prove that, if (C, V) is *not* satisfiable, then any SSPNE x has $c_n^x \geq 1$. Suppose, by contradiction, there exists an SSPNE $x = (x_n, a)$ with $c_n^x < 1$. This implies that $x_n^{a_w} > 0$ and $\nu_{a-n}^{r_w} = 0$, otherwise $c_n^x \geq 1$. Moreover, all the followers must experience a cost at most of 4, otherwise they would have an incentive to switch to a_w. Thus, for every $v \in V$, it must be the case that $\nu_{a-n}^{r_{v,t}} \leq 1$, otherwise, if $\nu_{a-n}^{r_{v,t}} \geq 2$, some followers would have a cost at least 5. Similarly, for every $\phi \in C$, it must be the case that $\nu_{a-n}^{r_\phi} \leq 1$. Following the same reasoning as in the proof of Theorem 6.12, we can build a truth assignment satisfying all clauses, a contradiction. Finally, let $\epsilon = \frac{1}{2^I}$, where I is the size of $\Gamma_\epsilon(C, V)$. Suppose there is a polynomial-time approximation algorithm $\mathcal{A}$ with approximation factor poly(I), *i.e.*, a polynomial function of I. If (C, V) is satisfiable, then $\mathcal{A}$ applied to $\Gamma_\epsilon(C, V)$ would return a solution with cost at most $\frac{1}{2^I}\text{poly}(I) < 1$, for I large enough. Thus, $\mathcal{A}$ would allow us to solve any 3-SAT instance in polynomial time, which is a contradiction unless P = NP holds. □

In conclusion, we provide some side results that deepen our analysis on how non-singleton actions impact on the complexity of finding an SSPNE in SCGs. The following theorem shows that our intractability results hold even in SCGs where only the followers have non-singleton actions.

Theorem 6.14. *The problem of computing an SSPNE in SCGs is* NP*-hard and not in Poly-*APX *unless* P = NP*, even when leader's actions are singletons, costs are monotonic, and followers are symmetric with actions of cardinality at most two.*

Proof. The result is obtained from the proofs of Theorems 6.12 and 6.13, by setting $A_n = \{a_w\}$ in the reduction. □

Let us observe that, since the SSPNEs of the games used in our reduction prescribe the leader to play a pure strategy, we have that the results in Theorems 6.12, 6.13, and 6.14 hold even if we restrict the leader to pure-strategy commitments.

In conclusion, we consider the case in which only the leader has non-singleton actions. It is easy to show that in SCGs with symmetric followers having singleton actions, an SSPNE can be found in polynomial time if we restrict the leader to play pure strategies. A polynomial-time algorithm enumerates the leader's pure strategies, and, for each of them, it computes an NE minimizing the leader's cost in the resulting followers' symmetric singleton congestion game, which can be done in polynomial time using dynamic programming, as shown by Ieong et al. (2005).

6.5 SSCGs with Multiple User Classes

In this section, we switch the attention to $\mathcal{T}$-class SSCGs, a generalization of symmetric SSCGs in which the players are partitioned into a finite number of classes. As shown by Theorem 6.3, finding an SSPNE in symmetric SSCGs with non-monotonic costs is NP-hard, while the problem becomes easy if: *(a)* we assume that the leader can only play pure strategies (Corollary 6.11.1), or *(b)* we force players' costs be monotonic (Theorem 6.9). Here, first we show that, under condition *(a)*, computing an SSPNE is easy also in $\mathcal{T}$-class SSCGs with a fixed number of classes. Next, we prove that, if condition *(a)* does not hold, then the problem is NP-hard in $\mathcal{T}$-class SSCGs even if we enforce *(b)* and there are only four followers' classes.

Let us start providing a polynomial-time algorithm for computing SSPNEs in $\mathcal{T}$-class SSCGs with a fixed number of classes and the leader restricted to pure strategies. We extend the dynamic programming method based on the recursive formula defined by Problem 6.5 and Corollary 6.11.1. Specifically, let $\mathcal{O}(h, B_1, \ldots, B_T, M_1, \ldots, M_T, V_1, \ldots, V_T)$ be the cost of an optimal NE for a $\mathcal{T}$-class singleton congestion game restricted to h resources $\{1, 2, \ldots, h\} \subseteq R$ and B_t players for each class $t \in \mathcal{T}$, where M_t is the largest cost experienced by a player of class t and V_t is the smallest cost a player of class t would get by switching to another resource.

Lemma 6.1. *$\mathcal{O}(h, B_1, \ldots, B_T, M_1, \ldots, M_T, V_1, \ldots, V_T)$ satisfies:*

$$\begin{aligned}
&\mathcal{O}(h, B_1, \ldots, B_T, M_1, \ldots, M_T, V_1, \ldots, V_T) = \\
&= \min_{\substack{p_t \in \{0,\ldots,B_t\}\ \forall t \in \mathcal{T} \\ m_t \in \{0,\ldots,M_t\}\ \forall t \in \mathcal{T} \\ v_t \in \{1,\ldots,V_t\}\ \forall t \in \mathcal{T}}} \mathcal{O}(h-1, p_1, \ldots, p_T, m_1, \ldots, m_T, v_1, \ldots, v_T) + b c_h(b)
\end{aligned}$$

$$\begin{aligned}
\text{s.t.}\quad & b = \sum_{t \in \mathcal{T}} (B_t - p_t) && && \text{(6.7a)}\\
& B_t = p_t && \forall t \in \mathcal{T} : h \notin A_t && \text{(6.7b)}\\
& m_t \leq M_t && \forall t \in \mathcal{T} && \text{(6.7c)}\\
& v_t \geq V_t && \forall t \in \mathcal{T} && \text{(6.7d)}\\
& c_h(b) \leq M_t && \forall t \in \mathcal{T} : B_t - p_t > 0 && \text{(6.7e)}\\
& c_h(b+1) \geq V_t && \forall t \in \mathcal{T} : h \in A_t && \text{(6.7f)}\\
& c_h(b) \leq v_t && \forall t \in \mathcal{T} : B_t - p_t > 0 && \text{(6.7g)}\\
& c_h(b+1) \geq m_t && \forall t \in \mathcal{T} : h \in A_t. && \text{(6.7h)}
\end{aligned}$$

Proof. We show that all the constraints are necessary. If Constrains (6.7b) were not satisfied, at least one player would play an action not available to her. If Constraints (6.7c) were not satisfied, there would exist a $t \in \mathcal{T}$ such that $m_t > M_t$, and, thus, there would be at least a resource in $\{1, \dots, h-1\}$ having cost larger than M_t for players of class t. If Constraints (6.7d) were not satisfied, there would exist a $t \in \mathcal{T}$ such that $v_t < V_t$, and, thus, players of class t would incur a cost strictly smaller than V_t when deviating to a resource in $\{1, \dots, h-1\}$. If Constraints (6.7e) were not satisfied, there would exist a $t \in \mathcal{T} : B_t - p_t > 0$ such that $c_h(b) > M_t$, and, thus, M_t would be smaller than the cost of the most expensive resource used by players of class t. If Constraints (6.7f) were not satisfied, there would exist a $t \in \mathcal{T} : h \in A_t$ such that $c_h(b+1) < V_t$, and, thus, players of class t would incur a cost strictly smaller than V_t upon deviating to another resource. If Constraints (6.7g) were not satisfied, there would exist a $t \in \mathcal{T} : B_t - p_t > 0$ such that $c_h(b) > v_t$ and, thus, at least one player of class t using resource h would have an incentive to deviate to another resource. If Constraints (6.7h) were not satisfied, there would exist a $t \in \mathcal{T} : h \in A_t$ such that $c_h(b+1) < m_t$ and at least one player of class t experiencing a cost of m_t would prefer to switch to resource h. $\square$

Thus, we can conclude the following:

Theorem 6.15. *In $\mathcal{T}$-class non-Stackelberg singleton congestion games, an optimal NE can be found in $O(n^{6T} r^{4T+1})$. In $\mathcal{T}$-class SSCGs, an SSPNE can be found in $O(n^{6T} r^{4T+2})$ if we restrict the leader to pure strategies.*

Proof. Since there are at most nr different values of costs $c_i(y)$ ($i \in R$, $y \in \{1, \dots, n\}$), for each $t \in \mathcal{T}$ there are at most nr values of M_t and V_t. There are also exactly r values of $i \in R$ and exactly n_t values of B_t for each $t \in \mathcal{T}$. Hence, $\mathcal{O}(h, B_1, \dots, B_T, M_1, \dots, M_T, V_1, \dots, V_T)$ has $O(n^{3T} r^{2T+1})$ entries. Computing an entry of the table requires $O(n^{3T} r^{2T})$. Overall, an optimal NE is computed in $O(n^{6T} r^{4T+1})$. For $\mathcal{T}$-class SSCGs with the leader restricted to play pure strategies, it suffices to run the algorithm for each $i \in A_n$, minimizing the cost of i. Since there are $O(r)$ leader's actions, the overall complexity is $O(n^{6T} r^{4T+2})$. $\square$

Now, we prove the hardness result, using a reduction from K-PARTITION, an NP-compete variant of PARTITION with an additional size constraint (see Definition 6.4).

Theorem 6.16. *Computing an SSPNE in $\mathcal{T}$-class SSCGs is* NP*-hard, even when cost functions are monotonic and $|\mathcal{T}| = 4$.*

Proof. Our reduction from K-PARTITION shows that a polynomial-time algorithm for finding an SSPNE in $\mathcal{T}$-class SSCGs would allow us to solve any K-PARTITION instance in polynomial time. Given a K-PARTITION instance $(\mathcal{S}, K)$, we build a game $\Gamma(\mathcal{S}, K)$ that admits an SSPNE x with $c_n^x \leq 2Y - \frac{Y}{K}$ if and only if $(\mathcal{S}, K)$ has answer *yes*, *i.e.*, there is $\mathcal{S}' \subseteq \mathcal{S}$ with $|\mathcal{S}'| = K$ and $\sum_{y_i \in \mathcal{S}'} y_i = Y$. We assume, w.l.o.g., that $y_i \leq Y$ for all $y_i \in \mathcal{S}$ (if not, $(\mathcal{S}, K)$ trivially has answer *no*).

Mapping. $\Gamma(\mathcal{S}, K)$ is defined as follows:

- $N = F \cup \{n\}$ and $\mathcal{T} = \{1, 2, 3, 4\}$, where $F = \bigcup_{t \in \mathcal{T}} F_t$ with $|F_1| = K$, $|F_2| = 2|\mathcal{S}|$, $|F_3| = 1$, and $|F_4| = 1$;
- $R = R_{\mathcal{S}} \cup \{r_w, r_x, r_y, r_z\}$ with $R_{\mathcal{S}} = \{r_i \mid y_i \in \mathcal{S}\}$;
- $A_1 = R_{\mathcal{S}} \cup \{r_w\}$, $A_2 = R_{\mathcal{S}} \cup \{r_z\}$, $A_3 = \{r_w, r_y\}$, $A_4 = \{r_x, r_y\}$, and $A_n = R_{\mathcal{S}} \cup \{r_y\}$.

Costs are specified in the table below, with $C_{r_y,F} = \frac{6K-2}{2K^2-K}$, $C_{r_i,F} = \left(1 - \frac{2YK}{y_i} + 2YK\right) \frac{2YK}{y_i}$, and $C_{r_i,n} = \frac{2Y(2Y-y_i)}{y_i}$.

y	$c_{r_i,F}$	$c_{r_w,F}$	$c_{r_x,F}$	$c_{r_y,F}$	$c_{r_z,F}$	$c_{r_i,n}$	$c_{r_y,n}$
1	0	$\frac{1}{K}$	$\frac{3}{K}$	$\frac{2}{K}$	$2YK$	$C_{r_i,n}$	0
2	$\frac{2YK}{y_i}$	1	$\frac{3}{K}$	$C_{r_y,F}$	$2YK$	$C_{r_i,n}$	Y^4
$[3, n]$	$C_{r_i,F}$	1	$\frac{3}{K}$	$C_{r_y,F}$	$2YK$	Y^4	Y^4

Clearly, $\Gamma(\mathcal{S}, K)$ can be constructed in polynomial time, since $n = K + 2|\mathcal{S}| + 3$, $r = |\mathcal{S}| + 4$, $|A_1| = |A_2| = |A_n| = |\mathcal{S}| + 1$, $|A_3| = |A_4| = 2$, and each cost can be encoded with a number of bits polynomial in the size of the instance $(\mathcal{S}, K)$. Notice that resource costs are monotonic.

If. Suppose that $(\mathcal{S}, K)$ has answer *yes*, and let $\mathcal{S}' \subseteq \mathcal{S}$ be such that $|\mathcal{S}'| = K$ and $\sum_{y_i \in \mathcal{S}'} y_i = Y$. Using $\mathcal{S}'$, we can recover $x = (x_n, \{\nu^t\}_{t \in \mathcal{T}})$ such that the followers' configurations $\{\nu^t\}_{t \in \mathcal{T}}$ represent an NE for x_n and $c_n^x = 2Y - \frac{Y}{K}$. Thus, in any SSPNE the leader's cost must be less than or equal to $2Y - \frac{Y}{K}$. In particular, for every $y_i \in \mathcal{S}'$, let $\nu^{1,r_i} = 1$, $\nu^{2,r_i} = 0$, and $x_n^{r_i} = \frac{y_i}{2YK}$. Instead, for $y_i \notin \mathcal{S}'$, let $\nu^{1,r_i} = 0$, $\nu^{2,r_i} = 2$, and $x_n^{r_i} = 0$. Moreover, we let $\nu^{1,r_w} = 0$, $\nu^{2,r_z} = 2K$, $\nu^{3,r_w} = 1$, $\nu^{3,r_y} = 0$, $\nu^{4,r_x} = 1$, $\nu^{4,r_y} = 0$, and $x_n^{r_y} = \frac{2K-1}{2K}$. It is easy to see that both $\{\nu^t\}_{t \in \mathcal{T}}$ and x_n are well-defined. Next, we prove that $\{\nu^t\}_{t \in \mathcal{T}}$ represent an NE for x_n. First, followers of class 1 experience a cost of $\frac{2YK}{y_i} \frac{y_i}{2YK} = 1$, and, thus, they do not have incentive to deviate to resource r_w, as they would still pay 1.

Similarly, they do not switch to another resource $r_i \in R_{\mathcal{S}}$, since, if $y_i \in \mathcal{S}'$, they would get a cost of $\frac{2YK}{y_i}(1 - \frac{y_i}{2YK}) + C_{r_i,F}\frac{y_i}{2YK} = 2YK > 1$, while, if $y_i \notin \mathcal{S}'$, they would pay $C_{r_i,F} > 1$. Moreover, followers of class 2 do not deviate, since, if they are selecting a resource $r_i \in R_{\mathcal{S}}$, then their current cost is $\frac{2YK}{y_i}$ and they would pay at least $2YK \geq \frac{2YK}{s_i}$ by deviating, while, if they are using r_z, then they experience a cost of $2YK$ and they would pay at least $2YK$ by switching to a resource $r_i \in R_{\mathcal{S}}$. Finally, the follower of class 3 pays $\frac{1}{K}$ and she does not deviate to r_y, as she would incur a cost at least of $\frac{2}{K}$, and, analogously, the follower of class 4 does not deviate since her cost would be $C_{r_y,F}\frac{2K-1}{2K} + \frac{2}{K}\frac{1}{2K} = \frac{3}{K}$ and she is paying $\frac{3}{K}$. In conclusion, $c_n^x = \sum_{y_i \in \mathcal{S}'} C_{r_i,n}\frac{y_i}{2YK} = \sum_{y_i \in \mathcal{S}'}\frac{2Y}{K} - \sum_{y_i \in \mathcal{S}'}\frac{y_i}{K} = 2Y - \frac{Y}{K}$, as $|\mathcal{S}'| = K$ and $\sum_{y_i \in \mathcal{S}'} y_i = Y$.

Only if. Suppose there exists an SSPNE $x = (x_n, \{\nu^t\}_{t \in \mathcal{T}})$ such that $c_n^x \leq 2Y - \frac{Y}{K}$. Using x, we build $\mathcal{S}' \subseteq \mathcal{S}$ such that $|\mathcal{S}'| = K$ and $\sum_{y_i \in \mathcal{S}'} y_i = Y$, showing that $(\mathcal{S}, K)$ has answer *yes*. First, it must be the case that $x_n^{r_y} > 0$ and $\nu^{r_y} = 0$, otherwise the leader's cost cannot be smaller than the minimum among costs $C_{r_i,n}$, which, since $y_i \leq Y$ for all $y_i \in \mathcal{S}$, is at least $\frac{2Y(2Y-Y)}{Y} = 2Y > 2Y - \frac{Y}{K}$. Thus, it must be $\nu^{3,r_y} = \nu^{4,r_y} = 0$ and $\nu^{3,r_w} = \nu^{4,r_x} = 1$. As a result, $x_n^{r_y} \geq 1 - \frac{1}{2K}$, otherwise the follower of class 4 would have an incentive to deviate to resource r_y, paying $C_{r_y,F}x_n^{r_y} + \frac{2}{K}(1 - x_n^{r_y}) < \frac{3}{K}$. This implies that $\sum_{r_i \in R_S} x_n^{r_i} \leq \frac{1}{2K}$. Moreover, $\nu^{1,r_w} = 0$, otherwise, if $\nu^{1,r_w} > 0$, the follower of class 3 would experience a cost of 1 and she would switch to r_y, paying at most $C_{r_y,F} < 1$, assuming, w.l.o.g., $K \geq 4$. Thus, there must be K different resources $r_i \in R_{\mathcal{S}}$ such that $\nu^{1,r_i} = 1$ and $\nu^{2,r_i} = 0$, since, if either $\nu^{1,r_i} > 1$ or $\nu^{2,r_i} > 0$, then $\nu^{r_i} > 1$ and the followers of class 1 selecting r_i would experience a cost greater than or equal to $\frac{2YK}{y_i} > 1$, thus having an incentive to deviate to r_w, paying 1. Let $R'_{\mathcal{S}} := \{r_i \in R_{\mathcal{S}} \mid \nu^{1,r_i} = 1\}$ (notice that $|R'_{\mathcal{S}}| = K$). It must be the case that $\nu^{2,r_i} < 3$ for all $r_i \notin R'_{\mathcal{S}}$, otherwise a follower of class 2 would have an incentive to deviate to r_z (as $C_{r_i,F} > 2YK$). Thus, since $|F_2| = 2|\mathcal{S}|$, there are at least $2K$ followers on r_z. Furthermore, $\nu^{2,r_i} > 0$ for all $r_i \notin R'_{\mathcal{S}}$, otherwise the followers of class 2 selecting r_z would have an incentive to switch to r_i, paying less than $\frac{2YK}{y_i} \leq 2YK$. Now, let us fix any $r_i \in R'_{\mathcal{S}}$. Say $x_n^{r_i} < \frac{y_i}{2YK}$, then the followers of class 2 using r_z would deviate to r_i, paying $\frac{2YK}{y_i}(1 - x_n^{r_i}) + C_{r_i,F}x_n^{r_i} < 2YK$. Moreover, say $x_n^{r_i} > \frac{y_i}{2YK}$, then the follower of class 1 on r_i would deviate to r_w, paying $1 < \frac{2YK}{y_i}x_n^{r_i}$. As a result, $x_n^{r_i} = \frac{y_i}{2YK}$ for every $r_i \in R'_{\mathcal{S}}$. Since $\sum_{r_i \in R_{\mathcal{S}}} x_n^{r_i} \leq \frac{1}{2K}$, we also have $\sum_{r_i \in R'_{\mathcal{S}}} x_n^{r_i} = \frac{1}{2YK}\sum_{r_i \in R'_{\mathcal{S}}} y_i \leq \frac{1}{2K}$, implying $\sum_{r_i \in R'_{\mathcal{S}}} y_i \leq Y$. It must also be the case that $x_n^{r_i} = 0$ for all $r_i \notin R'_{\mathcal{S}}$. If not, then there would be $r_j \in R'_{\mathcal{S}}$ with $\nu^{2,r_j} \in \{1,2\}$ and $x_n^{r_j} > 0$, which implies that $c_n^x = Y^4 x_n^{r_j} + \sum_{r_i \neq r_j \in R_{\mathcal{S}}} C_{r_i,n} x_n^{r_i} > \sum_{r_i \in R'_{\mathcal{S}}} \frac{2Y - y_i}{K} = 2Y - \frac{1}{K} \sum_{r_i \in R'_{\mathcal{S}}} y_i \geq 2Y - \frac{Y}{K}$, a contradiction. Finally, $c_n = \sum_{r_i \in R'_{\mathcal{S}}} C_{r_i,n} x_n^{r_i} = 2Y - \frac{1}{K} \sum_{r_i \in R'_{\mathcal{S}}} y_i \leq 2Y - \frac{Y}{K}$, which implies $\sum_{r_i \in R'_{\mathcal{S}}} y_i \geq Y$. Thus, $\sum_{r_i \in R'_{\mathcal{S}}} y_i = Y$. Letting $\mathcal{S}' \coloneqq \{y_i \in \mathcal{S} \mid r_i \in R'_{\mathcal{S}}\}$, we have $|\mathcal{S}'| = K$ and $\sum_{y_i \in \mathcal{S}'} y_i = Y$. □

6.6 MILP Formulations

In this last section, we show how the problem of computing an SSPNE in SCGs can be formulated as an MILP, providing different formulations for different classes of games. Our goal is to provide methods which work well in practice, even though their worst-case running time is exponential. [2] The proposed formulations are experimentally evaluated in Chapter 7.

Specifically, in Subsection 6.6.1, we provide two MILP formulations for the problem of computing an SSPNE in SSCGs and SSSCGs for which the problem is intractable (see Subsections 6.2.1 and 6.2.2). Then, in Subsection 6.6.2, we present other two formulations: the first one is specifically tailored for $\mathcal{T}$-class SSCGs, while the second one can be adopted in the general setting of SCGs (even with non-singleton actions).

6.6.1 MILPs for Computing SSPNEs in SSCGs and SSSCGs

We start from SSSCGs, for which the MILP formulation is simpler, and then extend the result to the more general case of SSCGs.

Computing an SSPNE in SSSCGs with Generic Costs

For the ease of notation, let $V \coloneqq \{1, \ldots, n-1\}$ be the set of possible congestion levels induced by the followers on a resource. Let, for every resource $i \in R$ and value $v \in V$, the binary variable y_{iv} be equal to 1 if and only if $\nu^i = v$, *i.e.*, if and only if v followers select resource $i \in R$. We use these variables to achieve a binarized representation of the followers' configuration $\nu \in \mathbb{N}^r$, namely, $\nu^i = \sum_{v \in V} v\, y_{iv}$ for all $i \in R$. Let, for each $i \in R$, $\alpha_i \in [0,1]$ be equal to x_n^i. Let also, for each $i \in R$ and $v \in V$, the auxiliary variable z_{iv} be equal to the bilinear term $y_{iv}\alpha_i$.

[2] We recall that, while we do not directly propose algorithms for the computation of WSPNEs, their computation can be carried out with the general method proposed in Section 4.4 for general SGs in normal form.

The complete MILP formulation reads:

$$\min \sum_{i \in R} \sum_{v \in V} c_{i,n}(v+1)\, z_{iv} \tag{6.8a}$$

$$\text{s.t.} \sum_{v \in V} y_{iv} \leq 1 \qquad \forall i \in R \tag{6.8b}$$

$$\sum_{i \in R} \sum_{v \in V} v\, y_{iv} = n - 1 \tag{6.8c}$$

$$\sum_{v \in V} \Big(y_{jv} c_{j,F}(v+1) + z_{jv} \Big(c_{j,F}(v+2) - c_{j,F}(v+1) \Big) \Big) \geq$$

$$\sum_{v \in V} \Big(y_{iv} c_{i,F}(v) + z_{iv} \Big(c_{i,F}(v+1) - c_{i,F}(v) \Big) \Big) \forall i \neq j \in R \tag{6.8d}$$

$$z_{iv} \leq \alpha_i \qquad \forall i \in R, v \in V \tag{6.8e}$$

$$z_{iv} \leq y_{iv} \qquad \forall i \in R, v \in V \tag{6.8f}$$

$$z_{iv} \geq \alpha_i + y_{iv} - 1 \qquad \forall i \in R, v \in V \tag{6.8g}$$

$$z_{iv} \geq 0 \qquad \forall i \in R, v \in V \tag{6.8h}$$

$$\sum_{i \in R} \alpha_i = 1 \tag{6.8i}$$

$$\alpha_i \geq 0 \qquad i \in R \tag{6.8j}$$

$$y_{iv} \in \{0, 1\} \qquad \forall i \in R, v \in V. \tag{6.8k}$$

Function (6.8a) represents the leader's expected cost (to be minimized). Constraints (6.8b) ensure that at most one variable y_{iv} be equal to 1 for each resource $i \in R$, thus guaranteeing that the congestion level of each resource be uniquely determined (note that $\sum_{v \in V} y_{iv} = 0$ if no followers select resource $i \in R$). Constraints (6.8c) guarantee that the followers' configuration be well-defined, *i.e.*, that $\sum_{i \in R} \nu^i$ be equal to $n-1$ (the number of followers). Constraints (6.8d) force the followers' configuration defined by the y_{iv} variables to be an NE for the leader's strategy identified by the α_i variables. This is because $\displaystyle\sum_{v \in V} \Big(y_{iv} c_{i,F}(v) + z_{iv} \Big(c_{i,F}(v+1) - c_{i,F}(v) \Big) \Big)$ (recall that $z_{iv} = y_{iv}\alpha_i$) is equal to the cost incurred by the followers selecting $i \in R$, while $\displaystyle\sum_{v \in V} \Big(y_{jv} c_{j,F}(v+1) + z_{jv} \Big(c_{j,F}(v+2) - c_{j,F}(v+1) \Big) \Big)$ (recall that $z_{jv} = y_{jv}\alpha_j$) is equal to the cost they would incur after deviating to resource $j \in R$. Let us remark that Constraints (6.8d) are trivially satisfied if $y_{iv} = 0$ for all $v \in V$. This is correct as, if no followers choose resource $i \in R$, no equilibrium conditions need to be enforced.

Constraints (6.8e)–(6.8h) are McCormick envelope constraints McCormick (1976) which guarantee $z_{iv} = y_{iv}\alpha_i$ whenever $y_{iv} \in \{0, 1\}$. Finally, we remark that Formulation (6.8) features $r(2n-1)$ variables, nr of which binary, and $r(r-1) + r(3n-2) + 2$ constraints.

We now extend Formulation (6.8) to general SSCGs. For the ease of notation, let, for $i \in R$, $\bar{v}_i := |\{p \in F \mid i \in A_p\}|$ be the maximum number of followers selecting i, and let $V(i) := \{1, \dots, \bar{v}_i\}$ be the set of possible congestion levels for i. For every $p \in F$ and $i \in A_p$, let the binary variable x_{pi} be equal to 1 if and only if $a_p = i$. All the variables in Formulation (6.8) are used with the same meaning.

The complete MILP formulation reads:

$$\min \sum_{i \in R} \sum_{v \in V(i)} c_{i,n}(v+1)\, z_{iv} \tag{6.9a}$$

$$\text{s.t.} \sum_{i \in A_p} x_{pi} = 1 \qquad \forall p \in F \tag{6.9b}$$

$$\sum_{v \in V(i)} y_{iv} \leq 1 \qquad \forall i \in R \tag{6.9c}$$

$$\sum_{v \in V(i)} v\, y_{iv} = \sum_{p \in F : i \in A_p} x_{pi} \qquad \forall i \in R \tag{6.9d}$$

$$\begin{aligned} &\sum_{v \in V(j)} \Big(y_{jv} c_{j,F}(v+1) + z_{jv} \Big(c_{j,F}(v+2) - c_{j,F}(v+1) \Big) \Big) \geq \\ &\sum_{v \in V(i)} \Big(y_{iv} c_{i,F}(v) + z_{iv} \Big(c_{i,F}(v+1) - c_{i,F}(v) \Big) \Big) \end{aligned} \qquad \forall p \in F, i \neq j \in A_p \tag{6.9e}$$

$$z_{iv} \leq \alpha_i \qquad \forall i \in R, v \in V(i) \tag{6.9f}$$

$$z_{iv} \leq y_{iv} \qquad \forall i \in R, v \in V(i) \tag{6.9g}$$

$$z_{iv} \geq \alpha_i + y_{iv} - 1 \qquad \forall i \in R, v \in V(i) \tag{6.9h}$$

$$z_{iv} \geq 0 \qquad \forall i \in R, v \in V(i) \tag{6.9i}$$

$$\sum_{i \in R} \alpha_i = 1 \tag{6.9j}$$

$$\alpha_i \geq 0 \qquad \forall i \in R \tag{6.9k}$$

$$\alpha_i = 0 \qquad \forall i \in R \setminus A_n \tag{6.9l}$$

$$x_{pi} \in \{0, 1\} \qquad \forall p \in F, i \in A_p \tag{6.9m}$$

$$y_{iv} \in \{0, 1\} \qquad \forall i \in R, v \in V(i). \tag{6.9n}$$

Objective Function (6.9a), Constraints (6.9c), and Constraints (6.9e)–(6.9k) have the same meaning as their counterparts in Formulation (6.8). Constraints (6.9b) ensure that each follower selects exactly one resource. Constraints (6.9d) guarantee that the followers' configuration be correctly defined, *i.e.*, for each $i \in R$, $\nu^i = \sum_{v \in V} v\, y_{iv}$ is equal to $\sum_{p \in F} x_{pi}$, which is the number of followers who select resource i. Notice that, differently from the previous formulation, Constraints (6.9e) are enforced for each follower $p \in F$ here, and only for pairs of resources $i, j \in R$ follower p has access to. Note also that, via Constraints (6.9l), α_i is forced to be 0 for all the resources $i \in R$ the leader has no access to.

We observe that Formulation (6.9) features $\sum_{p \in F} |A_p| + 2 \sum_{i \in R} \bar{v}_i + r = O(r(3n+1))$ variables, $\sum_{p \in F} |A_p| + \sum_{i \in R} \bar{v}_i = O(2rn)$ of which binary, and $n + 2r + 3 \sum_{i \in R} \bar{v}_i + \sum_{p \in F} |A_p| \left(|A_p| - 1\right) = O(n + 2r + 3nr + nr(r-1))$ constraints.

Computing a Pure-Strategy SSPNE in SSSCGs and SSCGs with Generic Costs

Formulations (6.8) and (6.9) can be easily modified for the case in which the leader's commitment is forced to be in pure strategies. This can be achieved by imposing the variables α_i to be binary. Notice that, when both α_i and y_{iv} are binary variables, z_{iv} becomes binary as well due to the McCormick constraints. The resulting formulations are ILPs.

One may wonder on the practical advantages of introducing an ILP formulation for computing a pure-strategy OSE in SSSCGs since, in Section 6.3.3, we have shown that this can be done in $O(n^4 r^4)$ by dynamic programming. As we will see comment on in Section 7.3, preliminary experiments show that, due to the high order of complexity of the dynamic programming algorithm, it is, in practice, more efficient to solve this formulation with state-of-the-art ILP algorithms than running the dynamic programming algorithm, even if solving Formulation (6.8) with branch-and-bound (and its variants, such as branch-and-cut) may take an exponential amount of computing time in the worst case.

6.6.2 MILPs for Computing SSPNEs in $\mathcal{T}$-Class SSCGs and SCGs

Let us first focus on $\mathcal{T}$-class SSCGs. For the ease of presentation, let $V(i) := \{1, \dots, \bar{v}_i\}$ for $i \in R$, with $\bar{v}_i := \sum_{t \in \mathcal{T} : i \in A_t} n_t$. Moreover, let $V(t) := \{1, \dots, n_t\}$ for $t \in \mathcal{T}$. For every class $t \in \mathcal{T}$, resource $i \in A_t$, and value $v \in V(t)$, let us introduce the binary variable q_{tiv}, which is equal to 1 if and only if v followers of class t select i. These variables represent followers' configurations. Specifically, for $t \in \mathcal{T}$, $\nu^t \in \mathbb{N}^r$ is such that $\nu^{t,i} = \sum_{v \in V(t)} v\, q_{tiv}$ for all $i \in R$. All the variables already defined in Formulation (6.8) are used with the same meaning. Finally, let $M > \max\{c_{i,F}(v) \mid i \in R, v \in \{1, \ldots, \bar{v}_i + 2\}\}$.

The complete MILP formulation reads as follows:

$$\min \quad \sum_{i \in R} \sum_{v \in V(i)} c_{i,n}(v+1)\, z_{iv} \tag{6.10a}$$

$$\text{s.t.} \sum_{v \in V(t)} q_{tiv} \leq 1 \qquad \forall t \in \mathcal{T}, i \in A_t \tag{6.10b}$$

$$\sum_{v \in V(i)} y_{iv} \leq 1 \qquad \forall i \in R \tag{6.10c}$$

$$\sum_{i \in A_t} \sum_{v \in V(t)} v\, q_{tiv} = n_t \qquad \forall t \in \mathcal{T} \tag{6.10d}$$

$$\sum_{t \in \mathcal{T}: i \in A_t} \sum_{v \in V(t)} v\, q_{tiv} = \sum_{v \in V(i)} v\, y_{iv} \qquad \forall i \in R \tag{6.10e}$$

$$\begin{aligned} &\sum_{v \in V(j)} \Big(y_{jv} c_{j,F}(v+1) + z_{jv} \Big(c_{j,F}(v+2) - c_{j,F}(v+1) \Big) \Big) \geq \\ &\sum_{v \in V(i)} \Big(y_{iv} c_{i,F}(v) + z_{iv} \Big(c_{i,F}(v+1) - c_{i,F}(v) \Big) \Big) + \\ &\qquad - M \Big(1 - \sum_{v \in V(t)} q_{tiv} \Big) \qquad \forall t \in \mathcal{T}, i \neq j \in A_t \end{aligned} \tag{6.10f}$$

$$z_{iv} \leq \alpha_i \qquad \forall i \in R, v \in V(i) \tag{6.10g}$$

$$z_{iv} \leq y_{iv} \qquad \forall i \in R, v \in V(i) \tag{6.10h}$$

$$z_{iv} \geq \alpha_i + y_{iv} - 1 \qquad \forall i \in R, v \in V(i) \tag{6.10i}$$

$$z_{iv} \geq 0 \qquad \forall i \in R, v \in V(i) \tag{6.10j}$$

$$\sum_{i \in R} \alpha_i = 1 \tag{6.10k}$$

$$\alpha_i \geq 0 \qquad \forall i \in R \tag{6.10l}$$

$$\alpha_i = 0 \qquad \forall i \notin A_n \tag{6.10m}$$

$$q_{tiv} \in \{0,1\} \qquad \forall t \in \mathcal{T}, i \in A_t, v \in V(t) \tag{6.10n}$$

$$y_{iv} \in \{0,1\} \qquad \forall i \in R, v \in V(i). \tag{6.10o}$$

Function (6.10a) is the leader's expected cost. Constraints (6.10b) ensure that at most one variable q_{tiv} be equal to 1 for each class $t \in \mathcal{T}$ and resource $i \in A_t$, and, thus, the number of followers of class t on each resource is uniquely determined (note that $\sum_{v \in V(t)} q_{tiv} = 0$ if no follower of class t selects resource i). Constraints (6.10c) ensure that at most one variable y_{iv} be equal to 1 for each resource $i \in R$, which guarantees that the congestion level of each resource is uniquely determined. Constraints (6.10d) guarantee that the followers' configuration be well-defined, *i.e.*, for all $t \in \mathcal{T}$, exactly n_t followers of class t are present. Constrains (6.10e) ensure that the congestion level on resource $i \in R$ be equal to the sum of the congestion levels induced by all classes. Constraints (6.10f) force the followers' configurations defined by the q_{tiv} variables be an NE for the leader's strategy identified by the α_i variables. This follows from the fact that, being $z_{iv} = y_{iv}\alpha_i$ and $z_{jv} = y_{jv}\alpha_j$, the right-hand side is the cost incurred by the followers of class $t \in \mathcal{T}$ selecting resource $i \in A_t$, while the left-hand side is the cost they would incur after deviating to $j \neq i \in A_t$. Note that, for each $t \in \mathcal{T}$, the constrains are active only if there is at least one follower of class t selecting i. Constraints (6.10g)–(6.8h) are McCormick envelope constraints McCormick (1976) guaranteeing $z_{iv} = y_{iv}\alpha_i$ if $y_{iv} \in \{0, 1\}$.

Next, we extend Formulation (6.10) to deal with general SCGs. Letting $\bar{v}_i := |\{p \in F \mid \exists a_p \in A_p : i \in a_p\}|$ be the maximum number of followers who can select resource $i \in R$, we define $V(i) := \{1, \dots, \bar{v}_i\}$. For every follower $p \in F$ and action $a_p \in A_p$, we introduce the binary variable $x_{p\,a_p}$, which is equal to 1 if and only if player p plays a_p. Moreover, for every leader's action $a_n \in A_n$, we let $\alpha_{a_n} \in [0, 1]$ be equal to $x_n^{a_n}$. All the variables already defined in Formulation (6.10) are used here with the same meaning. Finally, we also need to define the following constant $M > \sum_{i \in R} \max\{c_{i,F}(v) \mid i \in R, v \in \{1, \dots, \bar{v}_i + 2\}\}$.

The complete MILP formulation is provided in Problem (6.11), whose objective and constraints should be interpreted as follows. Function (6.11a), Constraints (6.11c), and Constraints (6.11e)–(6.11k) have the same meaning as their counterparts in Formulation (6.10). Note that, in this case, $z_{iv} = y_{iv} \sum_{a_n \in A_n : i \in a_n} \alpha_{a_n}$, where the summation represents the probability x_n^i and x_n is identified by the α_{a_n} variables. Constraints (6.11b) ensure that each follower selects exactly one action. Constraints (6.11d) guarantee that the followers' configuration represented by the variables y_{iv} be well-defined. Notice that Constraints (6.11e) are enforced for each follower $p \in F$ here, and they are active only for the action $a_p \in A_p$ that she plays. Finally, let us remark that, differently from the other formulations, Constraints (6.11e) (*i.e.*, the equilibrium constraints) must be enforced for each follower $p \in F$ and pair of actions $a_p \neq a'_p \in A_p$. Notice that Constraints (6.11e) do not account for the costs of resources $i \in a_p \cup a'_p$, since they do not change when deviating from a_p to a'_p.

$$\min \quad \sum_{i \in R} \sum_{v \in V(i)} c_{i,n}(v+1)\, z_{iv} \tag{6.11a}$$

$$\text{s.t.} \sum_{a_p \in A_p} x_{p\,a_p} = 1 \qquad \forall p \in F \tag{6.11b}$$

$$\sum_{v \in V(i)} y_{iv} \leq 1 \qquad \forall i \in R \tag{6.11c}$$

$$\sum_{v \in V(i)} v\, y_{iv} = \sum_{p \in F} \sum_{a_p \in A_p : i \in a_p} x_{p\,a_p} \qquad \forall i \in R \tag{6.11d}$$

$$\begin{aligned}\sum_{i \in a'_p \setminus a_p} \sum_{v \in V(i)} \Big(y_{iv} c_{i,F}(v+1) + z_{iv}\Big(c_{i,F}(v+2) - c_{i,F}(v+1)\Big)\Big) \geq \\ \sum_{i \in a_p \setminus a'_p} \sum_{v \in V(i)} \Big(y_{iv} c_{i,F}(v) + z_{iv}\Big(c_{i,F}(v+1) - c_{i,F}(v)\Big)\Big) + \\ - M\Big(1 - x_{p\,a_p}\Big) \qquad \forall p \in F, a_p \neq a'_p \in A_p\end{aligned} \tag{6.11e}$$

$$z_{iv} \leq \sum_{a_n \in A_n : i \in a_n} \alpha_{a_n} \qquad \forall i \in R, v \in V(i) \tag{6.11f}$$

$$z_{iv} \leq y_{iv} \qquad \forall i \in R, v \in V(i) \tag{6.11g}$$

$$z_{iv} \geq \sum_{a_n \in A_n : i \in a_n} \alpha_{a_n} + y_{iv} - 1 \qquad \forall i \in R, v \in V(i) \tag{6.11h}$$

$$z_{iv} \geq 0 \qquad \forall i \in R, v \in V(i) \tag{6.11i}$$

$$\sum_{a_n \in A_n} \alpha_{a_n} = 1 \tag{6.11j}$$

$$\alpha_{a_n} \geq 0 \qquad \forall a_n \in A_n \tag{6.11k}$$

$$x_{p\,a_p} \in \{0,1\} \qquad \forall p \in F, a_p \in A_p \tag{6.11l}$$

$$y_{iv} \in \{0,1\} \qquad \forall i \in R, v \in V(i). \tag{6.11m}$$

CHAPTER 7

Experimental Results on Stackelberg Games with Multiple Followers

This chapter concludes the first part of our work by experimentally evaluating the computational tools developed in the previous chapters for finding equilibria in SGs with a single and multiple followers.

Section 7.1 provides the results of extensive experiments carried out using algorithms for finding WSPNEs in normal-form SGs (see Chapter 4). Then, Section 7.2 shows experimental results related to the WSPNE-finding algorithm introduced for OLTSPGs (see Chapter 5), whereas Section 7.3 provides an experimental evaluation of the MILP formulations developed for computing SSPNEs in SCGs (see Chapter 6).

7.1 Normal-Form Stackelberg Games

We carry out an experimental evaluation of the equilibrium-finding algorithms introduced in Chapter 4, comparing the following methods:

- `QCQP`: the QCQP Formulation (4.5) solved with the state-of-the-art spatial-branch-and-bound solver BARON 14.3.1 (Sahinidis, 2014). As global optimality cannot be guaranteed by BARON if the feasible region of the problem is not bounded (Sahinidis, 2014), the solutions obtained with `QCQP` are not necessarily optimal.

- `MILP`: the MILP formulation derived according to Theorem 4.5 with dual variables artificially bounded by M, solved with the state-of-the-art MILP solver Gurobi 7.0.2.
- `BnB-sup`: the branch-and-bound algorithm we proposed, run for computing $\sup_{x_n \in \Delta_n} f(x_n)$. The algorithm is coded in Python 2.7, relying on Gurobi 7.0.2 as MILP solver.
- `BnB-`α: the branch-and-bound algorithm we proposed, run to find an α-approximate strategy whenever there is no $x_n \in \Delta_n$ at which the value of the supremum is attained.

For `MILP` and `Bnb-`α, we report the results for different values of M and α. `BnB-sup` and `BnB-`α are initialized with an outcome which results in an SSPNE for some leader's strategy. Specifically, we add it to S^+ in the starting node with empty S^- and to S^- in the starting node with empty S^+. The next node to explore is always selected according to a *best-bound* rule.

We generate a testbed of random normal-form SGs with payoffs independently drawn from a uniform distribution over the interval $[1, 100]$, using GAMUT (Nudelman et al., 2004). The results are then normalized to the interval $[0, 1]$ for the sake of presentation. The testbed contains games with $n = 3, 4, 5$ players (*i.e.*, with $2, 3, 4$ followers), $m \in \{4, 6, \ldots, 20, 25, \ldots, 70\}$ actions when $n = 3$, and $m \in \{3, 4, \ldots, 14\}$ actions when $n = 4, 5$. We generate 30 instances per pair of n and m.

We report the following figures, aggregated over the 30 instances per game with the same values of n and m:

- *Time*: average computing time, in seconds (up to the time limit).
- *LB*: average value of the best feasible solution found (only considered for instances where a feasible solution is found).
- *Gap*: average additive gap measured as UB $-$ LB, where UB is the upper bound returned by the algorithm. [1]
- *Opt*: percentage of instances solved to optimality (reported only for `BnB-sup`, as `QCQP` and `MILP` are not guaranteed to produce optimal solutions).

[1] When solving `QCQP` and `MILP`, *Gap* corresponds to the gap "internal" to the solution method. Since `QCQP` and `MILP` impose artificial restrictions (present by design in `MILP` and introduced automatically by the solver in `QCQP`), such value is, in general, not valid for the original, unrestricted problem. This is not the case for `BnB-sup` and `BnB-`α, for which *Gap* is a correct estimate of the difference between the best found LB and the value of the supremum (overestimated by UB).

- *Feas*: percentage of instances for which a feasible solution has been found (reported only for `QCQP` and `MILP` as an alternative to *Opt*).

The experiments are run on a UNIX machine with a total of 32 cores working at 2.3 GHz, equipped with 128 GB of RAM. The computations are carried out on a single thread, with a time limit of 3600 seconds per instance.

7.1.1 Experimental Results with Two Followers

Table 7.1 reports the results on games with two followers ($n = 3$) and $m \leq 30$, comparing `QCQP`, `MILP` (with $M = 10, 100, 1000$), `BnB-sup`, and `BnB-`α (with $\alpha = 0.001, 0.01, 0.1$).

`QCQP` can be solved only for instances with up to $m = 18$ due to BARON running out of memory on larger games. With $m \leq 18$, feasible solutions are found, on average, in 91% of the cases, but their quality is quite poor (the additive gap is equal to 0.34 on average). The time limit is reached on almost each instance, even those with $m = 4$, with the sole exception of those with $m = 18$, on which the solver halts prematurely due to memory issues.

`MILP` performs much better than `QCQP`, handling instances with up to $m = 30$ actions per player. $M = 100$ seems to be the best choice, for which we obtain, on average, LBs of 0.68 and gaps of 0.28, with a computing time slightly smaller than 2600 seconds. For $M = 1000$, the number of feasible solutions found increases from 94% to 97%, but LBs and gaps become slightly worse, possibly due to the fact that MILP solvers are typically quite sensitive to the magnitude of "big M" coefficients (which, if too large, can lead to large condition numbers, resulting in numerical issues).

`BnB-sup` substantially outperforms `QCQP` and `MILP`, finding not just feasible solutions but *optimal* ones for *every* game instance with $m \leq 25$ and solving to optimality 47% of the instances with $m = 30$. The average computing time is of 359 seconds, and it reduces to 126 if we only consider the instances with $m \leq 25$ (all solved to optimality). `BnB-sup` shows that the supremum of the leader's utility is very large on the games in our testbed, equal to 0.96 on average on the instances with $m \leq 25$ for which the supremum is computed exactly.

The time taken by `BnB-`α to find an α-approximate strategy is unaffected by the value of α. Since, in its implementation, `BnB-`α requires a relaxed outcome configuration on which the value of the supremum has been attained, we have run it only on instances with $m \leq 25$ (on which the supremum has always been computed).

	QCQP				MILP $M=10$				MILP $M=100$				MILP $M=1000$				BnB-sup				BnB-α $\alpha=0.001$		BnB-α $\alpha=0.01$		BnB-α $\alpha=0.1$	
m	Time	LB	Gap	Feas	Time	LB	Gap	Feas	Time	LB	Gap	Feas	Time	LB	Gap	Feas	Time	LB	Gap	Opt	Time	LBe	Time	LBe	Time	LBe
4	3600	0.81	0.19	100	2	0.83	0.00	100	1	0.86	0.00	100	1	0.85	0.00	100	1	0.86	0.00	100	1	0.86	1	0.85	0	0.76
6	3600	0.80	0.20	100	761	0.90	0.01	100	137	0.92	0.00	100	173	0.92	0.00	100	2	0.92	0.00	100	3	0.92	2	0.91	2	0.82
8	3600	0.70	0.30	100	1788	0.93	0.01	100	1419	0.94	0.01	100	1760	0.94	0.01	100	5	0.95	0.00	100	9	0.95	9	0.94	9	0.85
10	3600	0.67	0.33	100	2672	0.90	0.07	97	2161	0.96	0.02	100	2116	0.95	0.02	100	7	0.97	0.00	100	17	0.97	16	0.96	17	0.87
12	3600	0.63	0.37	100	3456	0.84	0.13	100	3184	0.90	0.08	100	3117	0.87	0.11	100	15	0.97	0.00	100	39	0.97	39	0.96	32	0.87
14	3600	0.57	0.43	97	3600	0.64	0.36	80	3585	0.69	0.31	100	3591	0.66	0.33	100	20	0.98	0.00	100	72	0.98	79	0.97	78	0.88
16	3600	0.45	0.55	77	3600	0.33	0.66	50	3600	0.62	0.38	93	3600	0.60	0.41	100	53	0.98	0.00	100	226	0.98	248	0.97	230	0.88
18	3243	0.58	0.42	50	3600	0.32	0.68	57	3600	0.53	0.47	83	3600	0.54	0.46	100	160	0.99	0.00	100	449	0.98	432	0.98	488	0.88
20	–	–	–	–	3600	0.31	0.69	73	3600	0.41	0.59	93	3600	0.43	0.57	100	222	0.99	0.00	100	1048	0.99	1056	0.98	1075	0.87
25	–	–	–	–	3600	0.32	0.68	73	3600	0.32	0.68	73	3600	0.33	0.67	73	777	0.99	0.00	100	3271	0.99	3112	0.98	3146	0.89
30	–	–	–	–	3600	0.35	0.64	97	3600	0.35	0.64	97	3600	0.35	0.65	97	2687	0.96	0.04	47	–	–	–	–	–	–

Table 7.1: *Experimental results for normal-form SGs with $n = 3$ players. The figures are averaged over games with the same values of m.*

	BnB-sup			
m	Time	LB	Gap	Opt
35	3573	0.80	0.21	3
40	3560	0.63	0.37	0
45	3600	0.50	0.50	0
50	3600	0.49	0.51	0
55	3600	0.53	0.47	0
60	3600	0.49	0.51	0
65	3600	0.50	0.50	0
70	3600	0.50	0.50	0

Table 7.2: *Results obtained with* `BnB-sup` *for normal-form SGs with* $n = 3$ *players and* $35 \leq m \leq 70$.

Table 7.2 reports further results obtained with `BnB-sup` for games with $n = 3$ and up to $m = 70$ actions per player. As the table shows, while some optimal solutions can still be found for $m = 35$, optimality is lost for game instances with $m \geq 40$. Nevertheless, `BnB-sup` still manages to find feasible solutions for instances with up to $m = 70$, obtaining solutions with an average LB of 0.55 and an average additive gap of 0.44. Under the conservative assumption that games with $35 \leq m \leq 70$ admit suprema of value close to 1 (which is empirically true when $m \leq 30$), `BnB-sup` provides, on average, solutions that are less than 50% off of optimal ones.

7.1.2 Experimental Results with More Followers and Observations

Results obtained with `BnB-sup` with more than two followers ($n = 4, 5$) are reported in Table 7.3 for $m \leq 14$. For the sake of comparison, we also report the results obtained for the same values of m and $n = 3$ that are contained in Tables 7.1 and 7.2.

	BnB-sup $n = 3$			BnB-sup $n = 4$			BnB-sup $n = 5$		
m	Time	Gap	Opt	Time	Gap	Opt	Time	Gap	Opt
4	0	0.00	100	3	0.00	100	8	0.00	100
6	2	0.00	100	17	0.00	100	137	0.00	100
8	5	0.00	100	126	0.00	100	2953	0.11	53
10	7	0.01	100	955	0.00	100	3461	0.46	13
12	15	0.00	100	2784	0.06	60	3600	0.53	0
14	20	0.01	100	3600	0.50	0	3600	0.52	0

Table 7.3: *Results obtained with* `BnB-sup` *for normal-form SGs with* $n = 4, 5$ *players and* $m = 4, 6, 8, 10, 12, 14$. *For comparison, the results for* $n = 3$ *are also reported.*

As the table illustrates, computing the value of the supremum of the leader's utility becomes very hard already for $m = 12$ with $n = 4$, for which the algorithm manages to find optimal solution in only 60% of the cases. For $m = 14$ and $n = 4$, no instance is solved to optimality within the time limit. For $n = 5$, the problem becomes hard already for $m = 8$, where only 53% of the instances are solved to optimality. With $m = 12$ and $n = 5$, no instances at all are solved to optimality.

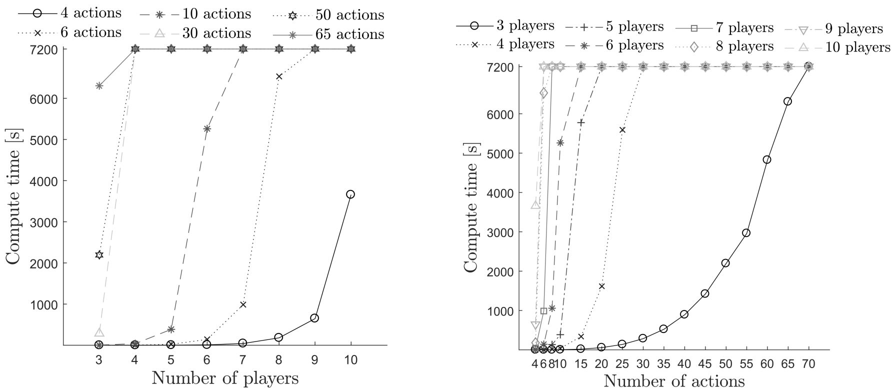
We do not report results on game instances with $n = 4, 5$ and $m > 14$ as such games are so large that, on them, `BnB-sup` incurs memory problems when solving the MILP subproblems.

In spite of the problem of computing a WSPNE being a nonconvex bilevel program, with our branch-and-bound algorithm we can find solutions with an additive optimality gap ≤ 0.01 for three-player games with up to $m = 20$ actions (containing three payoffs matrices with 8000 entries each), which are comparable, in size, to those solved in previous works which solely tackled the problem of computing a single NE maximizing the social welfare, see, *e.g.*, (Sandholm et al., 2005).

7.2 Stackelberg Polymatrix Games

We ran Algorithm 5.1 (*i.e.*, the algorithm finding an WSPNE in OLTSPGs) on a testbed of randomly generated game instances, evaluating the running time as a function of the number of players n and the number of actions per player m. Specifically, for each pair (n, m), times are averaged over 20 game instances, with $n \in \{3, \ldots, 10\}$ and $m \in \{4, 6, \ldots 10, 15, \ldots, 70\}$. Game instances have been randomly generated, with each payoff uniformly and independently drawn from the interval $[0, 100]$. All the experiments are run on a UNIX machine with a total of 32 cores working at 2.3 GHz, and equipped with 128 GB of RAM. Each game instance is solved on a single core, within a time limit of 7200 seconds. The algorithm is implemented in Python 2.7, while all LPs are solved with GUROBI 7.0, using the Python interface. Figures 7.1a and 7.1b show two plots of the average computing times, as a function of n and m, respectively.

We observe that, as expected, the computing time increases exponentially in the number of players n, while, once n is fixed, the growth is polynomial in the number of actions m. Specifically, the algorithm is able to solve within the time limit instances with three players, up to within $m = 65$, while, as the number of players increases, the scalability w.r.t. m decreases considerably, *e.g.*, with ten players, the algorithm can solve games with at most $m = 4$.



(a) *Average computing times (in seconds), as a function of the number of players n.*

(b) *Average computing times (in seconds), as a function of the number of actions m.*

Figure 7.1: *Average computing times (in seconds) of Algorithm 5.1.*

7.3 Stackelberg Congestion Games

In this section, we experimentally evaluate the MILP formulations for the problem of computing SSPNEs in SCGs proposed in Section 6.6.

Since Algorithm 6.1 has a very low complexity—$O(nr \log r)$—its efficiency is clear and it does not need to be established via computational experiments. As to the dynamic programming algorithm proposed in Section 6.3.3 for SSSCGs with generic costs when the leader is restricted to pure-strategy commitments, preliminary tests have shown that this algorithm takes several hours to solve instances which are solved only in a matter of seconds with a state-of-the-art ILP algorithm applied to Formulation (6.8). This happens in, *e.g.*, instances with 10 resources and 25 followers, on which the dynamic programming algorithm takes more than 11000 seconds while with the ILP formulation we can solve them in less than a second. Analogous considerations apply to the dynamic programming algorithm in Section 6.5. For these reasons, in the remainder of the section we solely focus on the mathematical programming formulations.

7.3.1 MILP Formulations for SSCGs and SSSCGs

Notice that games with monotonic cost functions and identical action spaces can be solved efficiently using Algorithm 6.1 (which we proposed in Subsection 6.3.1). For this reason, we focus on games with generic cost functions and/or different action spaces, assessing how state-of-the-art branch-and-bound methods behave when solving our formulations on instances of increasing size. We experiment with Formulations (6.8) and (6.9) on a testbed of randomly generated game instances of two classes:

- *SSSCGs*: we assume a number of followers in $\{20, 40, 60, 80, 100\}$, with r resources in the range $\{10, 20, 30, 40, 50\}$ and players' costs randomly generated by sampling from $\{1, \dots, (n-1)\, r\}$ with a uniform probability. [2]

- *SSCGs*: we assume a number of followers in $\{20, 40, 60, 80, 100\}$, with $r = 30$ resources and a number of actions $|A_p|$ per player in the range $\{7, 15, 22\}$, generated by sampling uniformly at random without replacement; the players' costs are sampled from $\{1, \dots, (n-1)\, r\}$ with uniform probability.

We also test our MILP formulations on the worst-case game instances generated by following the reductions of Theorems 6.1 and 6.3:

- *SSSCGs*: instances built following the reduction of Theorem 6.3 starting from K-PARTITION instances with $|S| \in \{50, 100, 150, 250, 300\}$ integer numbers with values sampled from $\{1, \dots, 100\}$.

- *SSCGs*: instances built following the reduction of Theorem 6.1 using random 3-SAT instances with $|V| \in \{3, 5, 7, 9, 11, 13\}$ variables and $|C| = k|V|$ clauses, where $k \approx 4.26$ is the phase-transition parameter which typically characterizes hard-to-solve 3-SAT instances (Cheeseman et al., 1991).

We generate 15 instances per combination of the parameters. All the experiments are run on a UNIX machine with a total of 32 cores working at 2.3 GHz, equipped with 128 GB of RAM. Each game instance is solved on a single core within a time limit of 7200 seconds. We use Python 2.7, solving the MILP formulations with GUROBI 7.0.

We use, as baseline for the comparisons, a simple algorithm which, starting from a randomly generated assignment of the players to the resources, simulates best-response dynamics halting after a time limit of 10 minutes. When ties arise, *i.e.*, whenever there are two or more players who are not playing their best response, we select a player lexicographically and make her switch to playing her (currently) best response. We refer to this algorithm as a *best response dynamics heuristic* since it is not exact when ap-

[2]The value $(n-1)\, r$ is chosen as, when looking for pure-strategy NEs, cost functions taking $(n-1)\, r$ different values are sufficient to represent every possible SCG.

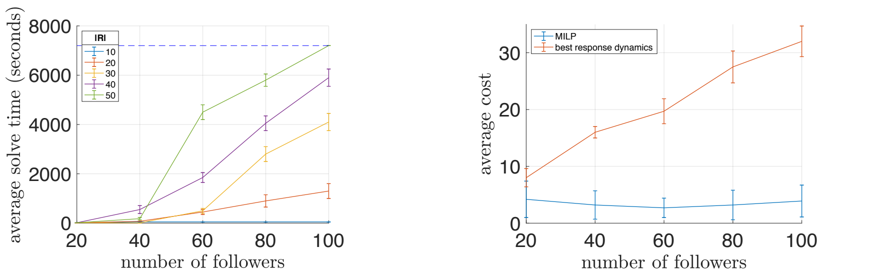
plied to the intractable cases of SSCGs and SSSCGs. [3] On average, within the time limit of 10 minutes we observe a number of deviations to a best response of the order of 10^5. Let us recall that the method always produces, by design, pure-strategy NEs. [4]

Figures 7.2 and 7.4a report the results for SSSCGs with generic costs. Figure 7.2a displays the average computing time required by Formulation (6.8), as a function of the number of followers and for different numbers of resources. One can see that, with Formulation (6.8), an optimal solution is always found within the time limit of 7200 seconds in all the instances. This suggests that, even if the problem is hard in the worst case, an optimal solution can be found in a reasonable amount of time on randomly generated instances. Figure 7.4a displays the results for worst-case instances. Surprisingly, within the time limit of 7200 seconds we are able to solve games with up to 302 resources and 1202 followers. Thus, while the instances generated by our reduction from K-PARTITION are the hardest ones asymptotically, they are solved more easily than randomly generated instances of the same size. Figure 7.2b reports, as a function of the number of followers, the average leaders' cost of the solutions obtained with Formulation (6.8), compared to the average cost obtained with the best-response dynamics heuristic. As the figure shows, the difference in leader's utility between solutions found with the two methods can be quite large as the number of followers increases, up to a factor of 6 with $n = 100$, showing a clearly growing trend.

Figure 7.3 and 7.4b report the results for SSCGs with generic costs and 30 resources. Figure 7.3a reports the average computing time required by Formulation (6.9) to find an SSPNE, as a function of the number of followers and for a different number of actions available to each player. Similarly to the case of SSSCGs, the chart shows that with Formulation (6.9) we can find an optimal solution within the time limit of 7200 seconds in all the instances. This suggests that, even if the problem is hard in the worst case, also for SSCGs one can find an optimal solution in a reasonable amount of computing time on randomly generated instances. The chart also shows, though, that the time required to solve this class of problems is much larger

[3] We also evaluated different heuristic algorithms combining Algorithm 6.1 together with best response dynamics, *e.g.*, using the solution returned by Algorithm 6.1 as a starting point for best-response dynamics instead of using randomly generated starting points. However, these approaches exhibited worse empirical performances than the best-response dynamics heuristic for all the settings which we have considered, including the more symmetric ones, showing that the algorithm does not benefit from the degree of symmetry of the instance.

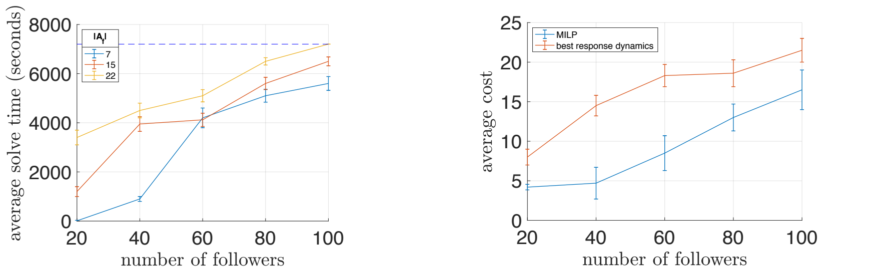
[4] Notice that, for games with identical action spaces, one could think of using the dynamic programming algorithm presented in Section 6.3.3 to find a pure-strategy SSPNE as heuristic approximation of a mixed-strategy SSPNE. However, as we mentioned above, the dynamic programming algorithm does not scale well enough in practice.



(a) *Average computing time required by Formulation* (6.8)*, as a function of the number of followers and for different numbers of actions available to each player.*

(b) *Average leader's cost obtained with Formulation* (6.8) *and with the best-response dynamics heuristic as a function of the number of followers, with* 30 *resources.*

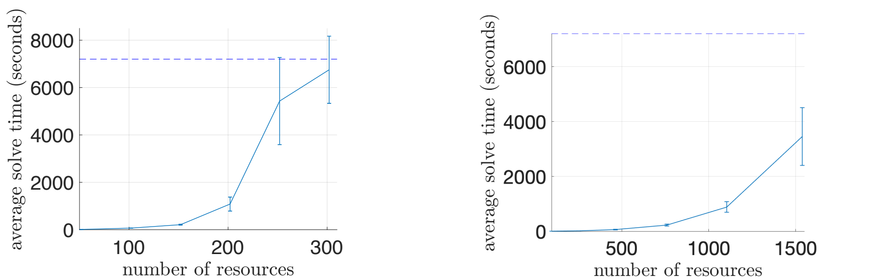
Figure 7.2: *Results for the computation of an SSPNE in SSSCGs with generic costs.*



(a) *Average computing time required by Formulation* (6.9)*, as a function of the number of followers and for different numbers of actions available to each player.*

(b) *Average leader's cost obtained with Formulation* (6.9) *and with the best-response dynamics heuristics, as a function of the number of followers and with 15 actions per player.*

Figure 7.3: *Results for the computation of an SSPNE in SSCGs with generic costs and 30 resources.*



(a) *Average computing time required by Formulation* (6.8) *for instances based on* K-PARTITION *(Theorem 6.3), as a function of the number of resources.*

(b) *Average computing time required by Formulation* (6.9) *for instances based on* 3-SAT *(Theorem 6.1), as a function of the number of resources.*

Figure 7.4: *Results for the computation of an SSPNE in worst-case instances built on the base of our inapproximability reductions.*

than the time required to solve their SSSCGs counterparts. Figure 7.4b displays results for worst-case instances generated using our reduction from 3-SAT. As for SSSCGs, these instances are not harder than random ones for the instance sizes used in our experimental setting. In particular, within the time limit of 7200 seconds, we are able to solve instances with up to 1538 resources and 2983 followers. Figure 7.3b reports, for games with 15 actions per player, the average leader's cost of the solutions obtained with the MILP Formulation (6.9) and with the best-response dynamics heuristic, as a function of the number of followers. Differently from the case of SSSCGs, we observe that for SSCGs the heuristic returns solutions which, empirically, appear to be within a constant approximation factor of the optimal ones which is never larger than 5.

Overall, the results suggest the practical viability of our MILP formulations for finding provably optimal solutions also for games where a simple best-response heuristic provides poor-quality solutions.

Surprisingly, the results that we have obtained for the worst-case instances are comparable to those for random games, empirically showing that, for the games that we study, random instances are not easier to solve than structured ones, differently from what is often observed in other cases (see, *e.g.*, (Sandholm et al., 2005) for the case of normal-form games).

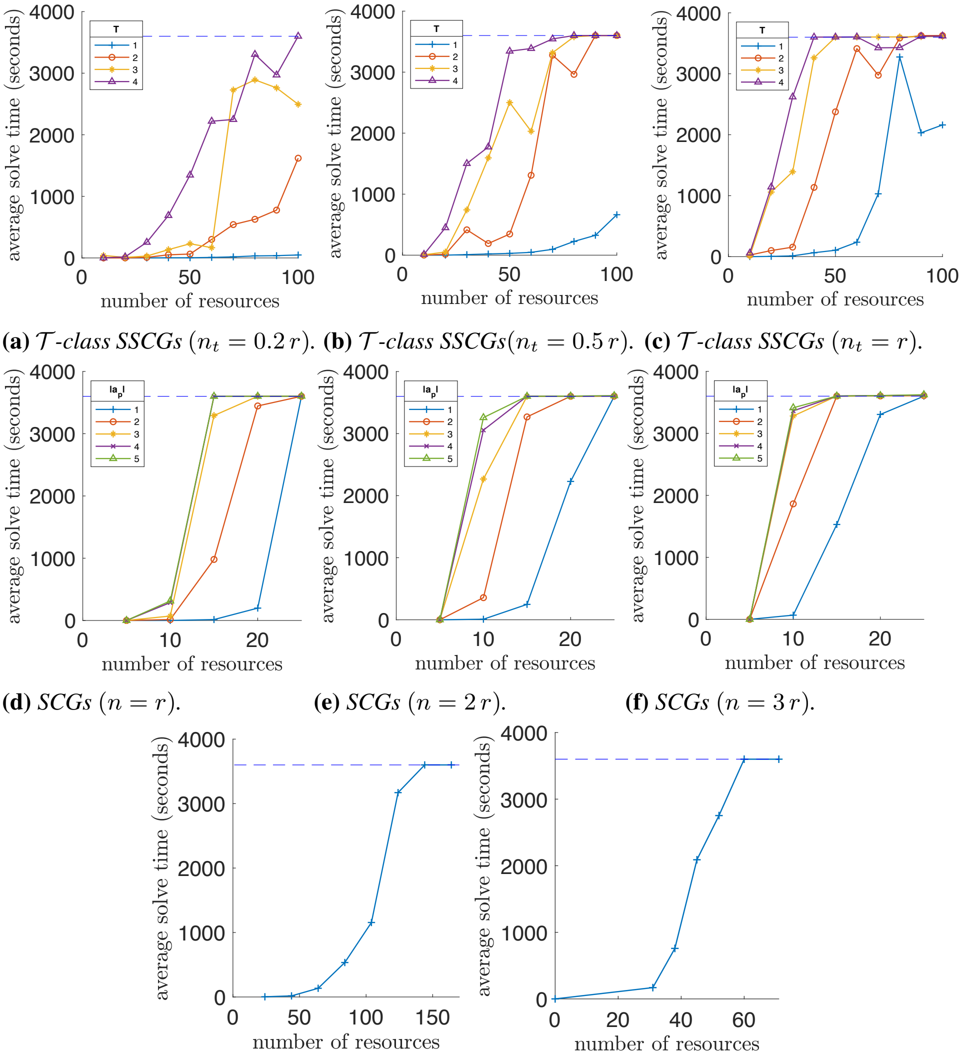
7.3.2 Experimental Evaluation on $\mathcal{T}$-Class SSCGs and SCGs

We test the MILP Formulations (6.10) and (6.11) on randomly generated games, which represent instances of average-case complexity, and games based on the reductions provided in the proofs of Theorems 6.12 and 6.16, which, instead, represent worst-case complexity instances.

All the experiments are run on a UNIX machine with a total of 32 cores working at 2.3 GHz, equipped with 128 GB of RAM. Each instance is solved on a single core within a time limit of 3600 seconds. We use GUROBI 7.0 (with Python interface) as MILP solver.

Random Game Instances

For $\mathcal{T}$-class SSCGs, we generate random games with a number of resources $r \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ and $T \in \{1, 2, 3, 4\}$ classes, with $n_t \in \{0.2\,r, 0.5\,r, r\}$ followers per class $t \in \mathcal{T}$ and $|A_t| = 0.5\,r$ actions per class $t \in \mathcal{T}$. Cost functions are randomly generated by sampling uniformly from $\{1, \dots, nrT\}$. For general SCGs, we generate instances with $r \in \{5, 10, 15, 20, 25\}$ resources and $n \in \{r, 2\,r, 3\,r\}$ followers, with $|a_p| \in \{1, 2, 3, 4, 5\}$ resources per action $a_p \in A_p$ and $|A_p| = 0.5\,r$ actions



(a) *$\mathcal{T}$-class SSCGs* ($n_t = 0.2\,r$). **(b)** *$\mathcal{T}$-class SSCGs*($n_t = 0.5\,r$). **(c)** *$\mathcal{T}$-class SSCGs* ($n_t = r$).

(d) *SCGs* ($n = r$). **(e)** *SCGs* ($n = 2\,r$). **(f)** *SCGs* ($n = 3\,r$).

(g) *Worst-case $\mathcal{T}$-class SSCGs.* **(h)** *Worst-case SCGs.*

Figure 7.5: *Computing times of Formulations* (6.10) *and* (6.11) *on randomly generated game instances and worst-case instances.*

per player $p \in N$. Cost functions are randomly generated by sampling uniformly from $\{1, \ldots, nr\}$. We build a testbed with 20 game instances per combination of the parameters.

Figure 7.5b displays the average computing times for Formulation (6.10) with $0.5\,r$ followers per class. The formulation scales quite well in practice. Symmetric games ($T = 1$) are quickly solved up to $r = 100$. Moreover, we are able to solve within the time limit games with up to four classes, 40 resources, and 160 players (40 players per class). Let us notice that the dynamic programming algorithm presented in Theorem 6.15 can be employed in this setting to find an SSPNE, if we restrict the leader to play pure strategies. However, preliminary experiments show that its scalability is extremely limited with respect to that of our formulation, as it finds a solution within the time limit only for games with less than 10 resources, while our formulation scales on much bigger games and it also works for mixed-strategy commitments.

Figure 7.5e shows the average computing times for Formulation (6.11) with $2\,r$ followers. We can conclude that, as expected, game instances with non-singleton actions are much harder to solve than singleton games. Here, the largest game instances we can solve within the time limit feature actions of cardinality two, 15 resources, and 30 players.

Finally, we test Formulation (6.11) on instances built according to the reduction in Theorem 6.12. Specifically, we generate these games from random 3-SAT instances with $|V| \in \{4, 5, 6, 7, 8, 9\}$ variables and $|C| = k|V|$ clauses, where $k \approx 4.26$ is the phase transition parameter determining generally hard-to-solve 3-SAT instances (Cheeseman et al., 1991). We test 10 random instances for each number of variables. Furthermore, we experiment Formulation (6.10) on instances based on the reduction in Theorem 6.16. We generate these games from random K-PARTITION instances with $|\mathcal{S}| \in \{20, 40, \ldots, 160\}$ integers, $y_i \in [2, 100]$ for all $y_i \in \mathcal{S}$, and $K = \frac{|\mathcal{S}|}{2}$. We test 10 random instances for each value of $|\mathcal{S}|$.

Figures 7.5g and 7.5h show the computing times for $\mathcal{T}$-class SSCGs and SCGs, respectively. Surprisingly, the results we obtain are comparable to those for random games, empirically showing that, for the games we study, random instances are not easier to solve than structured ones, as instead it is the case, *e.g.*, in normal-form games (Sandholm et al., 2005).

Part II

Stackelberg Games with Multiple Leaders

CHAPTER 8

Stackelberg Games with Multiple Leaders: Be a Leader or Become a Follower

In this chapter, we introduce a new way to address the problem of computing the strategies to commit to in SGs with *multiple leaders and followers*. Then, in Chapter 9, we study the computational properties of our model.

Section 8.1 introduces our approach, and, specifically, some reasonable properties of the leaders' commitments leading to the definition of different solution concepts. Then, Section 8.2 analyzes the game-theoretic properties of our model, while Section 8.3 investigates the relationship between our solution concepts and other commonly-studied equilibrium notions.

8.1 Multi-Leader-Follower Stackelberg Games

We address SGs with multiple leaders and followers, *i.e.*, games (Γ, L, F) with Γ being any (underlying) finite game, and L, respectively F, being the set of the leaders, respectively the followers. The key components of our model are the following. First, we allow the leaders to decide whether to participate in the commitment or to defect from it by taking on the role of followers. This is modeled by the *agreement stage* of the SG, whose result is the formation of an agreement involving a subset of the leaders. Second, in the spirit of CEs, we introduce a correlation device that, after the agreement, draws recommendations and privately communicates them to the players. Following the work by Conitzer and Korzhyk (2011), we assume that the leaders involved in the agreement commit to play their recommendations, while the followers obey to the usual incentive constraints of CEs (see Equation (2.3)). The correlation device may adopt different distributions depending on the sequence of defections that determined the agreement, and these distributions are publicly known. Our goal is to design the device, so as to achieve some desirable properties of the commitment, which we formally describe in the rest of the section.

Before going into our main definitions, we introduce some useful notation. Given a subset of players $P \subseteq N$, we denote with Π_P the collection of ordered subsets of P, including the empty set $\varnothing$. Given $\pi \in \Pi_P$ and $p \in P \setminus \pi$, we let πp be the ordered set obtained by appending p at the end of π. Moreover, let us recall that, given a subset of players $P \subseteq N$, we denote with $\mathcal{X}^{\text{CE}}_P \subseteq \mathcal{X}$ the set of correlated distributions which satisfy Equation (2.3) only for the players in P. Thus, $\mathcal{X}^{\text{CE}} := \mathcal{X}^{\text{CE}}_N$ defines the set of CEs of the underlying game.

We use $\mathbf{x} = [x_\pi]$ to denote a vector of correlated distributions $x_\pi \in \mathcal{X}^{\text{CE}}_{\pi \cup F}$, one per ordered subset of leaders $\pi \in \Pi_L$, while $\mathbf{X} = \times_{\pi \in \Pi_L} \mathcal{X}^{\text{CE}}_{\pi \cup F}$ is the set of all such vectors. In words, $\pi \in \Pi_L$ represents a sequence of leaders' defections in the agreement stage, while $\mathbf{x}$ defines the publicly known correlated distributions adopted by the correlation device, with x_π being the one used when the sequence of defections is π.

Definition 8.1 (Multi-Leader-Follower SG)**.** *Given a vector of distributions $\mathbf{x} = [x_\pi] \in \mathbf{X}$, an SG (Γ, L, F) is structured in the following two stages:*

- ***Agreement.*** *It goes on in rounds. In a given round, each leader, in turn, decides between* OPT-IN *and* OPT-OUT.[1] *All the decisions are perfectly observable. If a player chooses* OPT-OUT*, then she leaves the set of leaders becoming a follower, and a new round starts. The stage ends when, during a round, all the remaining leaders decide to* OPT-IN*. The result is the ordered subset $\pi \in \Pi_L$ of leaders who decided to* OPT-OUT. [2]

[1] We assume that the leaders are asked to take a decision according to some ordering, *e.g.*, $p \in L$ decides before $q \in L$ if $p < q$.

[2] The agreement stage is finite as there are at most $|L|$ rounds and each round involves at most $|L|$ decisions. Moreover, our results do not rely on the protocol implemented in the agreement stage. Others could be adopted, with the only requirement that they must record in which order the leaders do OPT-OUT.

- ***Play.*** *The correlation device draws some $s \in S$ according to the publicly known correlated distribution x_π. Then, each player is privately told her recommendation and the underlying game Γ is played, with the leaders in $L \setminus \pi$ sticking to their recommendations.*

The agreement stage of an SG can be represented as an extensive-form game involving the leaders. In such game, the players play in turn, according to some fixed order, with only two actions available at each decision point: OPT-IN and OPT-OUT. When a player chooses OPT-OUT, then she never plays anymore. The game ends after a sequence of OPT-IN actions performed by all leaders who have not selected OPT-OUT yet. Thus, each leaf of the game corresponds to the ordered subset $\pi \in \Pi_L$ representing the sequence of leaders who performed OPT-OUT on the path to the leaf. Players' payoffs are defined by $u_p(x_\pi)$ for $p \in L$. See Figure 8.1b for an example of sequential-game-representation of the agreement stage.

Next, we introduce some desirable properties that the distributions of the correlation device should satisfy. In the following definitions, we assume that an SG (Γ, L, F) is given.

First, we introduce *stability*. In words, we require that the leaders in L do not have any incentive to become followers. We introduce two different notions of stability, as follows.

Definition 8.2 (Stability)**.** *Given $\mathbf{x} = [x_\pi] \in \mathbf{X}$, for any $\pi \in \Pi_L$, x_π is* stable *if, for every $p \in L \setminus \pi$, $u_p(x_\pi) \geq u_p(x_{\pi p})$. Moreover:*

- $\mathbf{x}$ *is* stable *if $x_\varnothing$ is stable;*
- $\mathbf{x}$ *is* perfectly stable *if x_π is stable for every $\pi \in \Pi_L$.*

We denote with $\mathbf{X}^{\mathrm{S}} \subseteq \mathbf{X}$ and $\mathbf{X}^{\mathrm{PS}} \subseteq \mathbf{X}$ the sets of stable and perfectly stable distributions, respectively.

The rationale behind stability is that of NE. Indeed, $\mathbf{x} \in \mathbf{X}$ is stable if and only if each leader playing OPT-IN is an NE of the extensive-form game representing the agreement stage. Intuitively, this is because, if $\mathbf{x} \in \mathbf{X}$ is stable, each leader must not have any incentive to play OPT-OUT given that the other leaders always play OPT-IN. Instead, the rationale behind perfect stability is that of subgame perfection. Indeed, $\mathbf{x} \in \mathbf{X}$ is perfectly stable if and only if each leader playing OPT-IN is a subgame perfect equilibrium of the agreement stage. The reason is that perfect stability requires that playing OPT-IN is optimal at any decision point of the sequential game.

The second property that we look for is *efficiency*. We require that the correlated distributions of the correlation device are *Pareto optimal* with respect to the utility functions of the leaders who decided to OPT-IN. Given

$\mathbf{X}' \subseteq \mathbf{X}$, for $\pi \in \Pi_L$, we use $\mathcal{P}_{L\setminus\pi}(\mathbf{X}')$ to denote the set of Pareto optimal correlated distributions in the set $\{x'_\pi \mid \mathbf{x}' = [x'_\pi] \in \mathbf{X}'\}$, where the objectives are the functions u_p, for $p \in L \setminus \pi$. Formally:

Definition 8.3 (Efficiency)**.** *Given* $\mathbf{x} = [x_\pi] \in \mathbf{X}' \subseteq \mathbf{X}$*, for any* $\pi \in \Pi_L$*,* x_π *is* efficient *on the set* $\mathbf{X}'$ *if* $x_\pi \in \mathcal{P}_{L\setminus\pi}(\mathbf{X}')$*. Moreover:*

- $\mathbf{x}$ *is* efficient *on* $\mathbf{X}'$ *if* $x_\varnothing$ *is efficient on* $\mathbf{X}'$*;*
- $\mathbf{x}$ *is* perfectly efficient *on* $\mathbf{X}'$ *if* x_π *is efficient on* $\mathbf{X}'$ *for every* $\pi \in \Pi_L$*.*

We introduce three different solution concepts for our SGs, which we refer to as *Stackelberg correlated equilibria* (SCEs). They differ on the types of stability and efficiency that they prescribe. Formally:

Definition 8.4 (Stackelberg Correlated Equilibria)**.** *Given a multi-leader-follower SG* (Γ, L, F)*,* $\mathbf{x} \in \mathbf{X}$ *is an:*

- SCE *if it is efficient on the set* $\mathbf{X}^{\mathrm{S}}$*;*
- SCE with perfect agreement (SCE-PA) *if it is efficient on the set* $\mathbf{X}^{\mathrm{PS}}$*;*
- SCE with perfect agreement and perfect efficiency (SCE-PAPE) *if it is perfectly efficient on the set* $\mathbf{X}^{\mathrm{PS}}$*.*

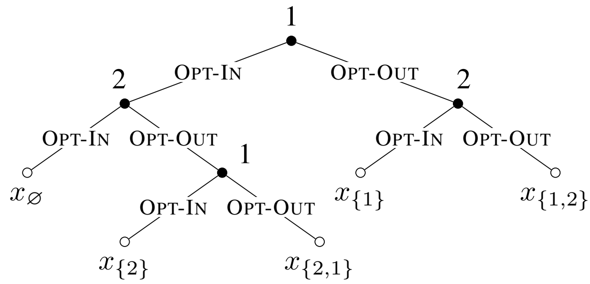
We denote with $\mathbf{X}^{\mathrm{SCE}}$, $\mathbf{X}^{\mathrm{SCE\text{-}PA}}$, and $\mathbf{X}^{\mathrm{SCE\text{-}PAPE}}$ the sets of SCEs, SCE-PAs, and SCE-PAPEs, respectively.

Example 8.1. *Consider the normal-form SG in Figure 8.1, where* $L = \{1, 2\}$ *and* $F = \varnothing$*.* [3] *Let* $\mathbf{x} = [x_\pi]$ *be such that* $x_\varnothing(s_{1,1}, s_{2,1}) = 1$*,* $x_{\{2\}}(s_{1,5}, s_{2,1}) = 1$*, and* $x_\pi(s_{1,1}, s_{2,2}) = 1$ *for all the other* $\pi \in \Pi_L$*. Clearly,* $x_\pi \in \mathcal{X}^{\mathrm{CE}}_\pi$ *for all* $\pi \in \Pi_L$*. Moreover, being* $x_\varnothing$ *stable and Pareto optimal,* $\mathbf{x}$ *is an SCE. Observe that, if player 2 performs* OPT-OUT*,* $\mathbf{x}$ *prescribes an irrational behavior to player 1, as* $u_1(x_{\{2\}}) = 0$*, while she gets* 1 *by doing* OPT-OUT*. Thus,* $\mathbf{x}$ *is not perfectly stable, as playing* OPT-IN *must be optimal at any decision point of the agreement stage. For instance,* $\mathbf{x}' = [x'_\pi]$ *with* $x'_\varnothing(s_{1,2}, s_{2,1}) = 1$ *and* $x'_\pi(s_{1,3}, s_{2,2}) = 1$ *for every other* $\pi \in \Pi_L$ *is an SCE-PA. However, notice that* $\mathbf{x}'$ *is not an SCE-PAPE since* $x'_{\{2\}}$ *does not maximize player 1's utility. Instead, it is easy to verify that* $\mathbf{x}'' = [x''_\pi]$ *with* $x''_\varnothing(s_{1,4}, s_{2,1}) = 1$*,* $x''_{\{2\}}(s_{1,3}, s_{2,1})$*, and* $x''_\pi(s_{1,4}, s_{2,2}) = 1$ *for all the other* $\pi \in \Pi_L$ *is an SCE-PAPE.*

[3] In this chapter and the following one, when representing normal-form games we adopt the convention that $A = S$, using the equivalence between action profiles in the normal form and pure strategy profiles.

	$s_{2,1}$	$s_{2,2}$
$s_{1,1}$	$5,0$	$1,2$
$s_{1,2}$	$4,1$	$1,2$
$s_{1,3}$	$2,1$	$1,1$
$s_{1,4}$	$3,2$	$1,3$
$s_{1,5}$	$0,0$	$0,0$

(a) *Underlying game.*



(b) *Extensive-form game representing the agreement stage.*

Figure 8.1: *Example of two-player normal-form SG with $L = \{1, 2\}$ and the extensive-form game representing its agreement stage.*

8.2 On the Existence of SCEs

We investigate the existence of our solution concepts in general SGs. We show that SCEs and SCE-PAs always exist, while we provide an SG where there is no SCE-PAPE.

The fundamental step for proving our existence results (Theorem 8.1) is to show that *(i)* $\mathbf{X}^{\text{S}}$ and $\mathbf{X}^{\text{PS}}$ are polytopes, and *(ii)* they are non-empty. The latter point is a direct consequence of the fact that all vectors $\mathbf{x} = [x_\pi] \in \mathbf{X}$ with $x_\pi = x$ for some CE $x \in \mathcal{X}^{\text{CE}}$ are perfectly stable.

First, we prove a useful property of stable distributions.

Lemma 8.1. *The sets $\mathbf{X}^{\text{S}}$ and $\mathbf{X}^{\text{PS}}$ are polytopes.*

Proof. $\mathbf{X} \subseteq \mathbb{R}^{|\Pi_L| \cdot |S|}$ is the set of vectors $\mathbf{x} = [x_\pi]$ such that $x_\pi \in \mathcal{X}^{\text{CE}}_{\pi \cup F}$ for all $\pi \in \Pi_L$. Each $\mathcal{X}^{\text{CE}}_{\pi \cup F}$ is defined by the linear constraints of Equation (2.3), thus $\mathbf{X}$ is a polytope. Moreover, if $\mathbf{x} \in \mathbf{X}^{\text{PS}} \subseteq \mathbf{X}$, x_π is stable for all $\pi \in \Pi_L$, *i.e.*, $u_p(x_\pi) \geq u_p(x_{\pi p})$ for all $p \in L \setminus \pi$. Thus, being these constraints linear, $\mathbf{X}^{\text{PS}}$ is a polytope. A similar argument holds for $\mathbf{X}^{\text{S}}$. □

Theorem 8.1. *Every SG admits an SCE and an SCE-PA.*

Proof. Given an SG (Γ, L, F), let $x \in \mathcal{X}^{\text{CE}}$ and $\mathbf{x} = [x_\pi] \in \mathbf{X}$ be such that $x_\pi = x$ for all $\pi \in \Pi_L$. We prove that $\mathbf{x} \in \mathbf{X}^{\text{PS}}$. First, for each $\pi \in \Pi_L$, $x_\pi \in \mathcal{X}^{\text{CE}}_{\pi \cup F}$, since $\mathcal{X}^{\text{CE}} \subseteq \mathcal{X}^{\text{CE}}_{\pi \cup F}$. Moreover, each x_π is stable, since $u_p(x_\pi) = u_p(x_{\pi p})$ for all $p \in L \setminus \pi$. This shows that $\mathbf{X}^{\text{PS}} \neq \varnothing$. Finally, being $\mathbf{X}^{\text{PS}}$ a polytope by Lemma 8.1, there exists $\mathbf{x} = [x_\pi] \in \mathbf{X}^{\text{PS}}$ such that $x_\varnothing \in \mathcal{P}_L(\mathbf{X}^{\text{PS}})$. Thus, $\mathbf{X}^{\text{SCE-PA}} \neq \varnothing$. A similar reasoning holds for the sets $\mathbf{X}^{\text{S}}$ and $\mathbf{X}^{\text{SCE}}$. □

Finally, we provide an example of SGs with no SCE-PAPE.

	$s_{2,1}$	$s_{2,2}$
$s_{1,1}$	$0,2,0$	$2,0,0$
$s_{1,2}$	$0,2,0$	$1,2,1$

$s_{3,1}$

	$s_{2,1}$	$s_{2,2}$
$s_{1,1}$	$0,2,0$	$2,1,1$
$s_{1,2}$	$0,2,0$	$0,0,0$

$s_{3,2}$

	$s_{2,1}$	$s_{2,2}$
$s_{1,1}$	$0,0,1$	$0,0,1$
$s_{1,2}$	$0,0,1$	$0,0,1$

$s_{3,3}$

Table 8.1: *Three-player normal-form SG with no SCE-PAPE (players 1, 2, and 3 select rows, columns, and matrices, respectively).*

Proposition 8.1. *There are SGs with no SCE-PAPE.* [4]

Proof. Consider the SG in Table 8.1, where $L = \{1,2,3\}$ and $F = \varnothing$. Suppose, by contradiction, that there exists $\mathbf{x} = [x_\pi] \in \mathbf{X}^{\text{SCE-PAPE}}$. First, for every x_π with player 3 in π, $u_3(x_\pi) = 1$ (otherwise $x_\pi \notin \mathcal{X}^{\text{CE}}_{\pi\cup F}$, as player 3 always gets 1 by deviating to $s_{3,3}$). Let us consider the sequences of OPT-OUT defined by the ordered subsets $\{1,2\}$ and $\{2,1\}$. Given that the definition of stability requires $u_3(x_{\{1,2\}}) \geq u_3(x_{\{1,2,3\}}) = 1$ and $u_3(x_{\{2,1\}}) \geq u_3(x_{\{2,1,3\}}) = 1$, we have that $x_{\{1,2\}}$ and $x_{\{2,1\}}$ must place strictly positive probability only on strategy profiles $(s_{1,2}, s_{2,2}, s_{3,1})$, $(s_{1,1}, s_{2,2}, s_{3,2})$, and those recommending $s_{3,3}$ to player 3. Moreover, player 1 cannot be told to play $s_{1,2}$, as she would have an incentive to deviate to $s_{1,1}$. The same holds for player 2 and strategy $s_{2,2}$. As a result, $x_{\{1,2\}}$ and $x_{\{2,1\}}$ must always recommend $s_{3,3}$ to player 3. Now, let us take the sequence of OPT-OUT defined by $\{1\}$. By stability of $x_{\{1\}}$, it must hold $u_3(x_{\{1\}}) \geq u_3(x_{\{1,3\}}) = 1$ and $u_2(x_{\{1\}}) \geq u_2(x_{\{1,2\}}) = 0$. Hence, given $x_{\{1\}} \in \mathcal{X}^{\text{CE}}_{\{1\}}$, in order to satisfy $x_{\{1\}} \in \mathcal{P}_{L\setminus\{1\}}(\mathbf{X}^{\text{PS}})$, $x_{\{1\}}$ must always recommend the strategy profile $(s_{1,1}, s_{2,2}, s_{3,2})$, where player 1 gets a utility of 2. Similarly, for the sequence defined by $\{2\}$, $x_{\{2\}}$ must always recommend $(s_{1,2}, s_{2,2}, s_{3,1})$ and, thus, player 2 receives a utility of 2. Thus, for stability, $x_\varnothing$ must satisfy $u_1(x_\varnothing), u_2(x_\varnothing) \geq 2$, which is clearly impossible. □

As a result, in the rest of this work we focus on SCEs and SCE-PAs. We remark that the non-existence of SCE-PAPEs implies that, under the requirements of perfect stability and perfect efficiency, there cannot be an agreement involving all the leaders. This does not rule out the possibility that some subsets of leaders can still reach an agreement. However, these cases are much more involved, as the actual group of leaders reaching an agreement inevitably depends on the rules of the protocol implemented in the agreement stage.

[4]Let us remark that this holds even if, for the efficiency of x_π, we require that $x_\pi \in \mathcal{P}_L(\mathbf{X}')$, *i.e.*, we use as objectives the utility function of all the leaders, including those who performed OPT-OUT.

8.3 SCEs and Other Solution Concepts

First, we show that the optimal correlated strategies to commit to introduced by Conitzer and Korzhyk [2011] are a special case of SCEs. Intuitively, in single-leader SGs, efficiency is equivalent to the maximization of leader's utility, while stability does not enforce additional constraints on the commitment. Formally:

Theorem 8.2. *Given an SG* $(\Gamma, \{1\}, N \setminus \{1\})$*, it holds* $\mathbf{X}^{\text{SCE}} = \mathbf{X}^{\text{SCE-PA}} = \mathbf{X}^{\text{SCE-PAPE}}$ *and, given some* $\mathbf{x} = [x_\pi] \in \mathbf{X}^{\text{SCE}}$*,* $x_\varnothing$ *is an optimal correlated strategy to commit to.*

Proof. Since the SG has only one leader (player 1), stability and perfect stability are equivalent, and, thus, $\mathbf{X}^{\text{S}} = \mathbf{X}^{\text{PS}}$. As a result, $\mathbf{X}^{\text{SCE}} = \mathbf{X}^{\text{SCE-PA}}$. Moreover, for the same reasons, also efficiency and perfect efficiency are equivalent, and $\mathbf{X}^{\text{SCE-PA}} = \mathbf{X}^{\text{SCE-PAPE}}$. Note that requiring Pareto optimality is the same as maximizing the leader's utility function u_1. Let $\mathbf{x} = [x_\pi] \in \mathbf{X}^{\text{SCE}}$ and assume, by contradiction, that $x_\varnothing$ is not an optimal correlated strategy to commit to. This would imply that there exists another $\hat{x} \in \mathcal{X}^{\text{CE}}_{N\setminus\{1\}}$ such that $u_1(\hat{x}) \geq u_1(x_\varnothing)$. However, replacing $x_\varnothing$ with $\hat{x}$ in $\mathbf{x}$ would give us another $\hat{\mathbf{x}} \in \mathbf{X}^{\text{S}}$ (stability constraints are trivially satisfied). This would contradict the efficiency of $\mathbf{x}$. □

Given the relation between optimal correlated strategies to commit to and SEs in single-leader single-follower SGs, we have the following:

Corollary 8.2.1. *Given an SG* $(\Gamma, \{1\}, \{2\})$*, any* $\mathbf{x} = [x_\pi] \in \mathbf{X}^{\text{SCE}}$ *is such that* $u_1(x_\varnothing)$ *is the leader's utility in an SE.*

8.3.1 SCEs and non-Stackelberg Correlation

Now, we analyze how our solution concepts relate to other non-Stackelberg solutions involving correlation. Specifically, we focus on CEs and CCEs. We recall that $\mathcal{X}^{\text{CE}}$ denotes the set of CEs of the underlying game, whereas, in the following, we let $\mathcal{X}^{\text{CCE}} \subseteq \mathcal{X}$ be the set of correlated distributions defining CCEs of the game.

In our analysis, we compare CEs and CCEs with the correlated distributions $x_\varnothing$ resulting from our solution concepts in general SGs. Given an SG (Γ, L, F), we define $\mathcal{X}^{\text{S}} \subseteq \mathcal{X}$ and $\mathcal{X}^{\text{PS}} \subseteq \mathcal{X}$ as the sets of $x_\varnothing$ such that $\mathbf{x} = [x_\pi] \in \mathbf{X}^{\text{S}}$ and $\mathbf{x} \in \mathbf{X}^{\text{PS}}$, respectively. Our goal is to investigate the relationships involving the sets $\mathcal{X}^{\text{S}}$ and $\mathcal{X}^{\text{PS}}$ with the sets of CEs and CCEs of the underlying game Γ, namely $\mathcal{X}^{\text{CE}}$ and $\mathcal{X}^{\text{CCE}}$.

Figure 8.2 depicts these relationships.

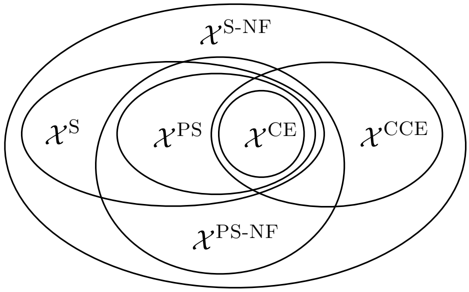


Figure 8.2: *Relations among $\mathcal{X}^{\text{S}}$, $\mathcal{X}^{\text{PS}}$, $\mathcal{X}^{\text{CE}}$, $\mathcal{X}^{\text{CCE}}$, $\mathcal{X}^{\text{S-NF}}$, $\mathcal{X}^{\text{PS-NF}}$.*

Let us remark that the relations $\mathcal{X}^{\text{CE}} \subseteq \mathcal{X}^{\text{CCE}}$, $\mathcal{X}^{\text{CE}} \subseteq \mathcal{X}^{\text{PS}}$, and $\mathcal{X}^{\text{PS}} \subseteq \mathcal{X}^{\text{S}}$ hold by definition, while it is easy to show that $\mathcal{X}^{\text{CE}} \subseteq \mathcal{X}^{\text{PS}}$ (see the proof of Theorem 8.1).

First, we look at the connection between (perfectly) stable distributions and CCEs. Given the relation between SEs and SCEs (see Corollary 8.2.1) in single-leader single-follower SGs, the following result holds as a direct consequence of (Von Stengel and Zamir, 2010, Remark 13).

Proposition 8.2. *There are SGs where $\mathcal{X}^{\text{CCE}} \not\subseteq \mathcal{X}^{\text{S}}$.*

Moreover, not all perfectly stable distributions are CCEs.

Proposition 8.3. *There are SGs where $\mathcal{X}^{\text{PS}} \not\subseteq \mathcal{X}^{\text{CCE}}$.*

Proof. Consider the SG in Table 8.2a, where $L = \{1, 2\}$ and $F = \varnothing$. Since $s_{1,1}$ and $s_{2,1}$ are strictly dominated, there is a unique CCE $x \in \mathcal{X}^{\text{CCE}}$ with $x(s_{1,2}, s_{2,2}) = 1$. Let $\mathbf{x} = [x_\pi] \in \mathbf{X}$ be such that $x_\varnothing(s_{1,1}, s_{2,1}) = 1$ and $x_\pi(s_{1,2}, s_{2,2}) = 1$ for all $\pi \neq \varnothing \in \Pi_L$. Notice that each x_π with $\pi \neq \varnothing$ satisfies the incentive constraints of Equation (2.3) for every player, and, thus, $x_\pi \in \mathcal{X}^{\text{CE}}_\pi$. Moreover, for each leader $p \in L$, $u_p(x_\varnothing) = 2$ and $u_p(x_\pi) = 1$ for all $\pi \in \Pi_L \setminus \{\varnothing\}$. Thus, each x_π is stable and $\mathbf{x} \in \mathbf{X}^{\text{PS}}$. □

Next, we analyze the relationships with the sets $\mathcal{X}^{\text{S-NF}}$ and $\mathcal{X}^{\text{PS-NF}}$, which are defined as $\mathcal{X}^{\text{S}}$ and $\mathcal{X}^{\text{PS}}$, but for the SG $(\Gamma, N, \varnothing)$ where each player is a leader. Our goal is to study the impact of players' roles in SGs having the same underlying finite game. The following result shows that enlarging the set of leaders can only introduce new stable distributions.

Theorem 8.3. $\mathcal{X}^{\text{S}} \subseteq \mathcal{X}^{\text{S-NF}}$ *and* $\mathcal{X}^{\text{PS}} \subseteq \mathcal{X}^{\text{PS-NF}}$.

	$s_{2,1}$	$s_{2,2}$
$s_{1,1}$	$2,2$	$0,3$
$s_{1,2}$	$3,0$	$1,1$

(a) *Example of two-player normal-form SG where $\mathcal{X}^{\text{PS}} \nsubseteq \mathcal{X}^{\text{CCE}}$.*

	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$	$s_{2,4}$
$s_{1,1}$	$0,0$	$-2,4$	$1,-8$	$1,-2$
$s_{1,2}$	$1,-8$	$0,0$	$-2,4$	$1,-2$
$s_{1,3}$	$-2,4$	$1,-8$	$0,0$	$1,-2$

(b) *Example of two-player normal-form SG where $\mathcal{X}^{\text{CCE}} \nsubseteq \mathcal{X}^{\text{PS-NF}}$.*

Table 8.2: *Two-player normal-form SGs used in the proofs of Propositions 8.3 and 8.5.*

Proof. We only prove the result for $\mathcal{X}^{\text{PS}}$, as similar arguments hold for $\mathcal{X}^{\text{S}}$. Given any SG (Γ, L, F), for every perfectly stable $\mathbf{x} = [x_\pi] \in \mathbf{X}^{\text{PS}}$ of (Γ, L, F), we show that there exists a perfectly stable $\mathbf{x}' = [x'_\pi] \in \mathbf{X}^{\text{PS}}$ of $(\Gamma, N, \varnothing)$, such that $x_\varnothing = x'_\varnothing$. Let us define $x'_\pi = x_{\pi \cap L}$, for all $\pi \in \Pi_N$. Clearly, it holds $x'_\pi \in \mathcal{X}^{\text{CE}}_\pi$, as $x'_\pi \in \mathcal{X}^{\text{CE}}_{(\pi\cap L)\cup F} \subseteq \mathcal{X}^{\text{CE}}_\pi$. For every player $p \in L$ and $\pi \in \Pi_N$ such that $p \notin \pi$, we have $u_p(x'_\pi) = u_p(x_{\pi\cap L})$ and $u_p(x'_{\pi p}) = u_p(x_{\pi p \cap L})$. Thus, given that $\mathbf{x} \in \mathbf{X}^{\text{PS}}$, $\mathbf{x}'$ satisfies the stability constraints for the players in L. Now, in order to show that $\mathbf{x}' \in \mathbf{X}^{\text{PS}}$, it is sufficient to prove that players in F do not have an incentive to OPT-OUT in $(\Gamma, N, \varnothing)$. This is the case as, for $p \in F$ and $\pi \in \Pi_N$ with $p \notin \pi$, we have $x'_{\pi p} = x'_\pi$. □

Furthermore, we can also provide examples showing that:

Proposition 8.4. *There are SGs where $\mathcal{X}^{\text{PS-NF}} \nsubseteq \mathcal{X}^{\text{S}}$.*

Proof. Consider the SG in Table 8.2a, where $L = \{1\}$ and $F = \{2\}$. There is an $\mathbf{x} = [x_\pi] \in \mathbf{X}^{\text{PS-NF}}$ of $(\Gamma, N, \varnothing)$ in which $x_\varnothing(s_{1,1}, s_{2,1}) = 1$ (see the proof of Proposition 8.3). Let $\mathbf{x}' = [x'_\pi] \in \mathbf{X}^{\text{S}}$ of (Γ, L, F). Since $x'_\varnothing \in \mathcal{X}^{\text{CE}}_{\{2\}}$ and $s_{2,1}$ is strictly dominated, it must be $x'_\varnothing(s_{1,1}, s_{2,1}) = 0$. □

Proposition 8.5. *There are SGs where $\mathcal{X}^{\text{CCE}} \nsubseteq \mathcal{X}^{\text{PS-NF}}$.*

Proof. Consider the SG in Table 8.2b, where $L = N = \{1, 2\}$. There is a CCE $x \in \mathcal{X}^{\text{CCE}}$ with $x(s_{1,1}, s_{2,1}) = x(s_{1,2}, s_{2,2}) = x(s_{1,3}, s_{2,3}) = \frac{1}{3}$. We show that there is no $\mathbf{x} = [x_\pi] \in \mathbf{X}^{\text{PS}}$ with $x_\varnothing = x$. By contradiction, assume there exists such $\mathbf{x}$. Given that $u_1(x_\varnothing) = 0$, it should be the case that $u_1(x_{\{1\}}) \leq 0$, by stability of $x_\varnothing$. Take the incentive constraints of player 1 (Equation (2.3)). Since there must be no incentive to deviate from $s_{1,1}$ to $s_{1,2}$, it holds

$$x_{\{1\}}(s_{1,1}, s_{2,3}) \geq \frac{1}{3} x_{\{1\}}(s_{1,1}, s_{2,1}) + \frac{2}{3} x_{\{1\}}(s_{1,1}, s_{2,2}).$$

Similar conditions also hold for the deviation from $s_{1,2}$ to $s_{1,3}$ and that from $s_{1,3}$ to $s_{1,1}$. Thus, we can write:

$$x_{\{1\}}(s_{1,2}, s_{2,1}) \geq \frac{1}{3}x_{\{1\}}(s_{1,2}, s_{2,2}) + \frac{2}{3}x_{\{1\}}(s_{1,2}, s_{2,3}),$$
$$x_{\{1\}}(s_{1,3}, s_{2,2}) \geq \frac{1}{3}x_{\{1\}}(s_{1,3}, s_{2,3}) + \frac{2}{3}x_{\{1\}}(s_{1,3}, s_{2,1}).$$

As a result, we can conclude that, if $x_{\{1\}}$ only recommends player 2 to play $s_{2,1}$, $s_{2,2}$, and $s_{2,3}$, then $u_2(x_{\{1\}}) < -2$. However, if player 2 decides to OPT-OUT, then she would get at least -2, as $x_{\{1,2\}} \in \mathcal{X}^{\text{CE}}$ and player 2 is guaranteed to get -2 by playing $s_{2,4}$. Thus, being $x_{\{1\}}$ stable, it must be the case that player 2 is always recommended to play $s_{2,4}$ in $x_{\{1\}}$. Thus, $u_1(x_{\{1\}}) = 1$, which is a contradiction. □

Finally, we prove that the stable distributions for the SG without followers encompass those defining CCEs.

Theorem 8.4. $\mathcal{X}^{\text{CCE}} \subseteq \mathcal{X}^{\text{S-NF}}$.

Proof. Let $x \in \mathcal{X}^{\text{CCE}}$ be a CCE of a given finite game Γ. We prove that the SG $(\Gamma, N, \varnothing)$ admits a stable distribution $\mathbf{x} = [x_\pi] \in \mathbf{X}^{\text{S}}$ with $x_\varnothing = x$. In order to do so, for every leader $p \in N$, we let $x_{\{p\}}$ be such that $u_p(x_{\{p\}}) \leq u_p(x)$, as shown in the following. Let us fix a player $p \in N$ and let $\hat{s}_p \in S_p$ be such that, for every $s'_p \in S_p$:

$$\sum_{s \in S} x(s) \left(u_p(\hat{s}_p, s_{-p}) - u_p(s'_p, s_{-p})\right) \geq 0, \tag{8.1}$$

i.e., $\hat{s}_p$ is the best player p's strategy against the correlated distribution x. Given that $x \in \mathcal{X}^{\text{CCE}}$:

$$\sum_{s \in S} x(s) \left(u_p(s) - u_p(\hat{s}_p, s_{-p})\right) \geq 0. \tag{8.2}$$

We define $x_{\{p\}}$ as follows:

- $x_{\{p\}}(\hat{s}_p, s_{-p}) = \sum_{s_p \in S_p} x(s_p, s_{-p})$ for all $s_{-p} \in S_{-p}$;
- $x_{\{p\}}(s_p, s_{-p}) = 0$ for all $s_p \neq \hat{s}_p \in S_p, s_{-p} \in S_{-p}$.

Given how $x_{\{p\}}$ is defined and Equation (8.1), we have that the incentive constraints of player p (Equation (2.3)) are satisfied, and, thus, $x_{\{p\}} \in \mathcal{X}^{\text{CE}}_{\{p\}}$. Moreover, Equation (8.2) implies that $u_p(x_{\{p\}}) \leq u_p(x)$, which concludes the proof. □

	$s_{2,1}$	$s_{2,2}$
$s_{1,1}$	k, k	$0, k+1$
$s_{1,2}$	$k+1, 0$	$1, 1$

Table 8.3: *Two-player normal-form SG (with $k > 0$) where the leaders' social welfare of an SCE-PA is arbitrary larger than in any CE.*

Observe that, when one looks for equilibria maximizing a linear function of leaders' utilities (*e.g.*, the leaders' social welfare), larger sets result in better solutions. [5] Moreover, we can provide examples where the difference in terms of leaders' social welfare between two solution concepts can be arbitrarily large. For instance, the following holds. [6]

Proposition 8.6. *There are SGs (Γ, L, F) with leaders' social welfare in SCE-PAs arbitrarily larger than in any CE of Γ.*

Proof. Consider the SG in Table 8.3, where $L = \{1, 2\}$ and $F = \varnothing$. Since strategies $s_{1,1}$ and $s_{2,1}$ are strictly dominated, the only CE is $x \in \mathcal{X}^{\text{CE}}$ with $x(s_{1,2}, s_{2,2}) = 1$. Let $\mathbf{x} = [x_\pi] \in \mathbf{X}$ be such that $x_\varnothing(s_{1,1}, s_{2,1}) = 1$ and $x_\pi(s_{1,2}, s_{2,2}) = 1$ for all $\pi \neq \varnothing \in \Pi_L$. It is easy to check that $\mathbf{x} \in \mathbf{X}^{\text{SCE-PA}}$. Moreover, the social welfare of the CE is 2, while the social welfare of the SCE-PA is $2k$. $\square$

[5] Let us remark that, since $\mathbf{X}^{\text{S}}$ and $\mathbf{X}^{\text{PS}}$ are polytopes (see Lemma 8.1), maximizing a linear function of leaders' utilities over the sets $\mathbf{X}^{\text{S}}$ and $\mathbf{X}^{\text{PS}}$ also provides Pareto optimality, and, thus, efficiency over the corresponding set.

[6] Similar results hold for the other pairs of solution concepts.

CHAPTER 9

Computing Stackelberg Correlated Equilibria

In this chapter, we study the computational complexity of computing equilibria in the multi-leader-follower SGs introduced in the previous chapter.

Section 9.1 studies the complexity of finding SCEs, assuming to have access to an oracle solving an auxiliary problem, called *stability oracle*. Then, in the following Section 9.2, we show which classes of games admit a polynomial-time stability oracle.

9.1 Computational Complexity of SCEs

We study the computational complexity of SCEs and SCE-PAs in general SGs. We distinguish between the problem of finding *an* equilibrium and that of computing an *optimal* equilibrium, *i.e.*, one maximizing a specific given linear function of leaders' utilities, such as the leader's social welfare. We introduce the following formal definitions (problems f-SCE-PA and o-SCE-PA(λ) are defined analogously for SCE-PAs).

Definition 9.1 (f-SCE)**.** *Given an SG* (Γ, L, F)*, find an SCE.*

Definition 9.2 (o-SCE(λ)). *Given an SG (Γ, L, F) and $\lambda = [\lambda_p] \in [0,1]^{|L|}$, find an SCE $\mathbf{x} = [x_\pi] \in \mathbf{X}^{\text{SCE}}$ maximizing the objective function $f_\lambda = \sum_{p \in L} \sum_{s \in S} \lambda_p u_p(s) x_\varnothing(s)$.*

Let us remark that, in general, the size of a vector $\mathbf{x} \in \mathbf{X}$ is factorial in the number of players. Thus, in the following, we assume that there is some compact representation for $\mathbf{x}$. [1]

We establish a tight connection between our problems and an auxiliary one, which is a generalization of the problem of finding an optimal CE. In the rest of the section, we assume to have access to an oracle solving this auxiliary problem, which we call *stability oracle*. In Section 9.2, we then investigate for which games the oracle can be efficiently implemented.

Definition 9.3. *A* stability oracle $\mathcal{O}(\Gamma, c, L, \{x_p\}_{p \in L' \subseteq L})$ *is an algorithm that, given a finite game Γ, a coefficients vector $c = [c_p] \in [-1,1]^n$, a set of leaders $L \subseteq N$, and a collection of correlated distributions $x_p \in \mathcal{X}$ for $p \in L' \subseteq L$, returns an $x \in \mathcal{X}^{\text{CE}}_{N \setminus L}$ maximizing $\sum_{p \in N} \sum_{s \in S} c_p u_p(s) x(s)$ subject to the stability constraints, i.e., $u_p(x) \geq u_p(x_p)$ for all $p \in L'$.* [2]

In the following, we are interested in games where the stability oracle runs in polynomial time. Thus, we assume that $\mathcal{O}$ always returns a correlated distribution with size polynomial in the size of the game. [3] We also consider the decision form of the stability oracle, which reads as follows:

Definition 9.4. *The* decision form *of a stability oracle $\mathcal{O}$ is an algorithm $\mathcal{O}^{\text{D}}(x, L, \{x_p\}_{p \in L' \subseteq L})$ that, given $x \in \mathcal{X}$, $L \subseteq N$, and $x_p \in \mathcal{X}$ for $p \in L' \subseteq L$, answers* YES *if $x \in \mathcal{X}^{\text{CE}}_{N \setminus L}$ and x satisfies the stability constraints, and* NO *otherwise.*

In the following, given $L \subseteq N$ and $\lambda = [\lambda_p] \in [0,1]^{|L|}$, we let $c_\lambda = [c_{\lambda,p}] \in [0,1]^n$ be such that $c_{\lambda,p} = \lambda_p$ if $p \in L$, while $c_{\lambda,p} = 0$ if not. Moreover, given $p \in N$, we let $c_p = [c_{p,q}] \in [0,1]^n$ be such that $c_{p,p} = -1$ and $c_{p,q} = 0$ for all $q \in N \setminus \{p\}$. Note that c_λ is the coefficients vector of the objective f_λ, while c_p corresponds to minimizing u_p.

9.1.1 Computing SCEs

We show that, in games admitting a polynomial-time stability oracle, an optimal SCE can be computed in polynomial time. Intuitively, o-SCE(λ)

[1] As we see next, for all our positive results we can safely assume that there is a compact representation for $\mathbf{x} \in \mathbf{X}$ (*e.g.*, $\mathbf{x}$ only requires a polynomial number of polynomially-sized distributions).

[2] Note that, given a finite game G, $\mathcal{O}(\Gamma, c, \varnothing, \varnothing)$ returns an optimal CE $x \in \mathcal{X}^{\text{CE}}$ for the objective function defined by $c \in [0,1]^n$.

[3] Indeed, this assumption is not restrictive, as all the games we study in Section 9.2 admit a poly-time oracle $\mathcal{O}$ with this property.

is solved by $\mathbf{x} = [x_\pi]$ computed as: $x_{\{p\}} = \mathcal{O}(\Gamma, c_p, L \setminus \{p\}, \varnothing)$ for $p \in L$, $x_\varnothing = \mathcal{O}(\Gamma, c_\lambda, L, \{x_{\{p\}}\}_{p \in L})$, and $x_\pi = \mathcal{O}(\Gamma, c_\lambda, \varnothing, \varnothing)$ for every other ordered subset $\pi \in \Pi_L$. Formally:

Theorem 9.1. *Given an SG* (Γ, L, F) *and* $\lambda \in [0,1]^{|L|}$, o-SCE(λ) *can be solved with* $|L| + 2$ *queries to an oracle* $\mathcal{O}$.

Proof. We build an $\mathbf{x} = [x_\pi] \in \mathbf{X}^{\text{SCE}}$ that maximizes f_λ by invoking a stability oracle $\mathcal{O}$ multiple times. For every $p \in L$, we define $x_{\{p\}} = \mathcal{O}(G, c_p, L \setminus \{p\}, \varnothing)$. Moreover, we let $x_\varnothing = \mathcal{O}(\Gamma, c_\lambda, L, \{x_{\{p\}}\}_{p \in L})$ and $x_\pi = \mathcal{O}(\Gamma, c_\lambda, \varnothing, \varnothing)$ for every $\pi \in \Pi_L$ with $|\pi| \geq 2$. Clearly, we need $|L| + 2$ calls to $\mathcal{O}$. First, $x_\pi \in \mathcal{X}^{\text{CE}}_{\pi \cup F}$ for every $\pi \in \Pi_L$, by definition of $\mathcal{O}$. For the same reason, we have $u_p(x_\varnothing) \geq u_p(x_{\{p\}})$ for all $p \in L$. Thus, we can conclude that $\mathbf{x} \in \mathbf{X}^{\text{S}}$. Let f_λ be the value of the objective for $\mathbf{x}$. We show that f_λ is maximized over $\mathbf{X}^{\text{S}}$. Being f_λ a linear combination of leader's utility functions, we immediately get that $x_\varnothing \in \mathcal{P}_L(\mathbf{X}^{\text{S}})$, and $\mathbf{x} \in \mathbf{X}^{\text{SCE}}$. By contradiction, suppose that there exists an $\mathbf{x}' = [x'_\pi] \in \mathbf{X}^{\text{S}}$ with objective function value $f'_\lambda > f_\lambda$. This implies that there exists a leader $p \in L$ with $u_p(x'_{\{p\}}) < u_p(x_{\{p\}})$, otherwise the solution $x_\varnothing$ returned by $\mathcal{O}(\Gamma, c_\lambda, L, \{x_{\{p\}}\}_{p \in L})$ would not be optimal. This is a contradiction, since $x_{\{p\}}$ minimizes player p's utility on the set $\mathcal{X}^{\text{CE}}_{\{p\} \cup F}$, and $x'_{\{p\}} \in \mathcal{X}^{\text{CE}}_{\{p\} \cup F}$. $\square$

Corollary 9.1.1. *Given an SG* (Γ, L, F)*, if there is a poly-time oracle* $\mathcal{O}$*, then* o-SCE(λ) *can be solved in polynomial time.*

9.1.2 Computing SCE-PAs

First, we provide a positive result: one can find *an* SCE-PA with polynomially many invocations to a stability oracle. It is sufficient to compute $\mathbf{x} = [x_\pi]$ where $x_{\{p\}} = \mathcal{O}(\Gamma, c_p, \varnothing, \varnothing)$ for $p \in L$ and, additionally, $x_\varnothing = \mathcal{O}(\Gamma, c_\lambda, L, \{x_{\{p\}}\}_{p \in L})$. Thus:

Theorem 9.2. *Given an SG* (Γ, L, F)*,* f-SCE-PA *can be solved with* $|L|+1$ *queries to an oracle* $\mathcal{O}$.

Proof. Using $\mathcal{O}$, we construct an $\mathbf{x} \in \mathbf{X}^{\text{SCE-PA}}$. Let $x_{\{p\}} = \mathcal{O}(\Gamma, c_p, \varnothing, \varnothing)$, *i.e.*, $x_{\{p\}}$ is a CE that minimizes player p's utility. Moreover, we define $x_\varnothing = \mathcal{O}(\Gamma, c_\lambda, L, \{x_{\{p\}}\}_{p \in L})$ for some $\lambda \in (0,1]^{|L|}$. By setting, for every leader $p \in L$, $x_\pi = x_{\{p\}}$ for all $\pi \in \Pi_L$ where p is the first to OPT-OUT, we have $\mathbf{x} \in \mathbf{X}^{\text{PS}}$. Clearly, we only require $|L| + 1$ queries to $\mathcal{O}$. Now, we prove that $x_\varnothing \in \mathcal{P}_L(\mathbf{X}^{\text{PS}})$, and, thus, $\mathbf{x} \in \mathbf{X}^{\text{SCE-PA}}$. By contradiction, suppose that it is not the case, *i.e.*, there exists an $\mathbf{x}' = [x'_\pi] \in \mathbf{X}^{\text{PS}}$ with $u_p(x'_\varnothing) \geq u_p(x_\varnothing)$ for all $p \in L$ and $u_q(x'_\varnothing) > u_q(x_\varnothing)$ for some leader

$q \in L$. By stability of $x_\varnothing$, we have that $u_p(x'_\varnothing) \geq u_p(x_\varnothing) \geq u_p(x_{\{p\}})$ for every $p \in L$. Thus, $x'_\varnothing$ satisfies $u_p(x'_\varnothing) \geq u_p(x_{\{p\}})$ for every leader $p \in L$ (stability), and

$$\sum_{p \in N} \sum_{s \in S} c_{\lambda,p} u_p(s) x'_\varnothing(s) > \sum_{p \in N} \sum_{s \in S} c_{\lambda,p} u_p(s) x_\varnothing(s),$$

which implies that $x'_\varnothing$ verifies the constraints for a solution to the porblem solved by $\mathcal{O}(\Gamma, c_\lambda, L, \{x_{\{p\}}\}_{p \in L})$, while providing an objective grater than that of $x_\varnothing$. This contradicts the correctness of $\mathcal{O}$. □

Corollary 9.2.1. *Given an SG (Γ, L, F), if there is a poly-time oracle $\mathcal{O}$, then* f-SCE-PA *can be solved in polynomial time.*

Now, we switch to the problem of computing an optimal SCE-PA, showing that it cannot be solved efficiently, even with access to a polynomial-time stability oracle. Specifically, we prove a stronger negative result: even the easier problem of verifying the perfect stability of a given $\mathbf{x} \in \mathbf{X}$ is computationally intractable. Our statement is based on a reduction from the coNP-complete problem of deciding whether a given formula in *disjunctive normal form* (DNF) is a tautology or not (Arora and Barak, 2009).

Theorem 9.3. *Given an SG (Γ, L, F) and $\mathbf{x} \in \mathbf{X}$, verifying whether $\mathbf{x} \in$ or $\notin \mathbf{X}^{\text{PS}}$ is not in* P *unless* NP = coNP*, even with access to a polynomial-time decision-form oracle $\mathcal{O}^{\text{D}}$.*

Proof. Given a DNF formula Φ, we build an SG and an $\mathbf{x} \in \mathbf{X}$ such that $\mathbf{x} \in \mathbf{X}^{\text{PS}}$ if and only if Φ is a tautology. Thus, if one could verify the perfect stability of $\mathbf{x}$ in polynomial time, then there would be a polynomial-time checkable certificate for the coNP-complete problem of determining whether a DNF formula is a tautology or not Arora and Barak (2009). This would imply NP = coNP. Moreover, given how the SG is built, the result holds even if we get access to a polynomial-time decision oracle $\mathcal{O}^{\text{D}}$.

Construction. Given a DNF formula Φ, let V denote the set of variables appearing in Φ. We construct an SG (Γ, L, F) involving a leader for each variable and a single follower, *i.e.*, $L = \{p_v \mid v \in V\}$ and $F = \{p_f\}$. Moreover, we let $S_{p_f} = \{s_v \mid v \in V\}$ be the set of follower's strategies, one per variable, while the leaders share the strategies $S_{p_v} = \{s_{\text{T}}, s_{\text{F}}\}$, corresponding to truth values . As a result, any strategy profile $s \in S$ corresponds to a truth assignment τ^s defined by leaders' strategies. We write $\Phi(\tau^s) = \text{T}$ if τ^s satisfies Φ, while $\Phi(\tau^s) = \text{F}$ otherwise. We also denote with $\#\text{F}(\tau^s)$ the number of false variables in τ^s. Table 9.1 reports the leaders' utilities, while the follower's one is always 0. Then, we build

	$\Phi(\tau^s) = \mathrm{T}$		$\Phi(\tau^s) = \mathrm{F}$	
	$s_{p_f} = s_v$	$s_{p_f} \neq s_v$	$\exists v : s_{p_v} = s_{\mathrm{F}}$	$\forall v : s_{p_v} = s_{\mathrm{T}}$
s_{T}	0	$\#\mathrm{F}(\tau^s)$	$\lvert V\rvert$	-1
s_{F}	$\#\mathrm{F}(\tau^s) - 1$	$\lvert V\rvert$	$\lvert V\rvert$	0

Table 9.1: *Leader p_v's ($v \in V$) utilities in the SG for the reduction of Theorem 9.3. On rows, there are p_v's strategies s_{T} and s_{F}, whereas, on columns, we report the four possible cases for $s \in S$. $\#\mathrm{F}(\tau^s)$ denotes the number of variables set to false by τ^s.*

an $\mathbf{x} = [x_\pi] \in \mathbf{X}$ with $x_\varnothing(s) = 1$ for some $s \in S$ such that $s_{p_v} = s_{\mathrm{T}}$ for every $v \in V$. Furthermore, for every $v \in V$ and $\pi \in \Pi_{L \setminus \{p_v\}}$, we let $x_{\pi p_v}(s) = 1$ for $s \in S$ with $s_p = s_{\mathrm{F}}$ for every $p \in \pi p_v$, $s_p = s_{\mathrm{T}}$ for every $p \in L \setminus \pi p_v$, and $s_{p_f} = s_v$. Let us remark that our SG admits a polynomial-time decision oracle $\mathcal{O}^{\mathrm{D}}(x, L, \{x_p\}_{p \in L' \subseteq L})$, since it can be queried in polynomial time only on polynomially-sized distributions.

If. We prove that, if Φ is a tautology, then $\mathbf{x} \in \mathbf{X}^{\mathrm{PS}}$. For every $\pi \in \Pi_L$, x_π recommends all the leaders in π to play s_{F}. Moreover, being Φ a tautology, strategy s_{F} (weakly) dominates s_{T} (as it is always the case that $\Phi(\tau^s) = \mathrm{T}$). Thus, $x_\pi \in \mathcal{X}^{\mathrm{CE}}_{\pi \cup F}$. Note that, for every $v \in V$ and $\pi \in \Pi_{L \setminus \{p_v\}}$, $u_{p_v}(x_\pi) = \#\mathrm{F}(\tau^s) = |\pi|$, while, if p decides to OPT-OUT, she is recommended to play s_{F} and, being $s_{p_f} = s_v$, she gets the same utility. As a result, all distributions x_π are stable.

Only if. We prove that, if Φ is not a tautology, then $\mathbf{x} \notin \mathbf{X}^{\mathrm{PS}}$. Let $s \in S$ be such that $\phi(\tau^s) = \mathrm{F}$. Two cases are possible. If $s_{p_v} = s_{\mathrm{T}}$ for every $v \in V$, then $x_\varnothing$ is not stable as the leaders would have an incentive to OPT-OUT (since they get at least $0 > -1$). If this is not the case, then there exist $s, s' \in S$ such that $\Phi(\tau^s) = \mathrm{T}$ and $\Phi(\tau^{s'}) = \mathrm{F}$, where $x_\pi(s) = 1$ and $x_{\pi p_v}(s') = 1$ for some $v \in V$ and $\pi \in \Pi_{L \setminus \{p_v\}}$. In this case, $u_{p_v}(x_\pi) = \#\mathrm{F}(\tau^s) = |\pi|$ and $u_{p_v}(x_{\pi p_v}) = |V| > |\pi|$. Thus, x_π is not stable, as leader p_v would have an incentive to OPT-OUT. □

Corollary 9.3.1. *Given an SG (Γ, L, F) and $\mathbf{x} \in \mathbf{X}$, verifying whether $\mathbf{x} \in \mathbf{X}$ is an SCE-PA maximizing the social welfare is not in* P *unless* NP = coNP*, even with access to a polynomial-time decision-form oracle $\mathcal{O}^{\mathrm{D}}$.*

Proof. We can modify the proof of Theorem 9.3 so that, when Φ is a tautology, $\mathbf{x} \in \mathbf{X}$ is the only perfectly stable distribution maximizing the social welfare. In order to do this, it is enough to add a leader with a single action and utility $|V|^2$ if $s_{p_v} = s_{\mathrm{T}}$ for all $v \in V$, while 0 otherwise. □

As a byproduct of Theorem 9.1 we have that, when looking for optimal SCEs, one can restrict the attention to those $\mathbf{x} \in \mathbf{X}$ admitting a representation whose size is polynomial in the size of the game. For Theorem 9.2, the same holds when searching for an SCE-PA. However, Theorem 9.3 implies that optimal SCE-PAs require an exponential number of different distributions. Moreover, even when $\mathbf{x} \in \mathbf{X}$ can be easily represented in a compact form (as in the proof of Theorem 9.3), we cannot check in polynomial time whether $\mathbf{x} \in \mathbf{X}^{\text{PS}}$ or not.

This poses a new intriguing question: can we restrict the attention to $\mathbf{x} \in \mathbf{X}$ whose size is less than factorial in the number of players? We show that the answer is positive. It is sufficient to consider $\mathbf{x} \in \mathbf{X}$ whose size is exponential in the number of players, as only the unordered set of defecting leaders and the last of them who decided to OPT-OUT matter.

Theorem 9.4. *Given an SG* (Γ, L, F) *and* $\mathbf{x} = [x_\pi] \in \mathbf{X}^{\text{PS}}$*, there is an* $\mathbf{x}' = [x'_\pi] \in \mathbf{X}^{\text{PS}}$ *s.t.* $x'_\varnothing = x_\varnothing$ *and* $x'_{\pi p} = x'_{\pi' p}$ *for every* $p \in L$ *and* $\pi, \pi' \in \Pi_{L \setminus \{p\}}$ *defining the same set.*

Proof. Let us take some $\mathbf{x} \in \mathbf{X}^{\text{PS}}$. For every $p \in L$ and $\pi \in \Pi_{L\setminus\{p\}}$, we define $x'_{\pi p} = x_{\pi' p}$ where $x_{\pi' p}$ minimizes $u_p(x_{\pi' p})$ over all $\pi' \in \Pi_{L\setminus\{p\}}$ such that π and π' define the same set. Moreover, let $x'_\varnothing = x_\varnothing$. Clearly, $x'_\pi \in \mathcal{X}^{\text{CE}}_{\pi \cup F}$ for all $\pi \in \Pi_L$ (as each x'_π is set equal to an $x_{\pi'}$ such that π and π' correspond to the same set of leaders who performed OPT-OUT). Moreover, it is easy to check that $\mathbf{x}'$ is perfectly stable, as follows. Let us consider some $p \in L$ and $\pi \in \Pi_{L\setminus\{p\}}$. By definition, for every $q \in L \setminus \pi p$, it holds $u_q(x'_{\pi p}) = u_q(x_{\pi' p})$, for some $\pi' \in \Pi_{L\setminus\{p\}}$. Moreover, $u_q(x_{\pi' p}) \geq u_q(x_{\pi' pq})$ by stability of $\mathbf{x}$, and $u_q(x_{\pi' pq}) \geq u_q(x'_{\pi'' pq})$ for some $\pi'' \in \Pi_{L\setminus\{p\}}$. Finally, by definition of $\mathbf{x}'$, we have that $u_q(x'_{\pi'' pq}) = u_q(x'_{\pi pq})$, which shows that $u_q(x'_{\pi p}) \geq u_q(x'_{\pi pq})$. Since this holds for any $p \in L$ and $\pi \in \Pi_{L\setminus\{p\}}$, we conclude that $\mathbf{x}' \in \mathbf{X}^{\text{PS}}$. □

Theorem 9.4 allows us to reduce the number of queries to a stability oracle that are necessary to find an optimal SCE-PA.

Theorem 9.5. *Given an SG* (Γ, L, F) *and* $\lambda \in [0,1]^{|L|}$, o-SCE-PA(λ) *can be solved with* $|L| 2^{|L|-1} + 1$ *queries to* $\mathcal{O}$.

Proof. We build an $\mathbf{x} = [x_\pi] \in \mathbf{X}^{\text{SCE-PA}}$ that maximizes f_λ by using a stability oracle $\mathcal{O}$. For every $p \in L$ and $\pi \in \Pi_{L\setminus\{p\}}$ with $\pi p = L$, we let $x_{\pi p} = \mathcal{O}(\Gamma, c_p, \varnothing, \varnothing)$. Otherwise, whenever $\pi p \neq L$, letting $\pi' = \pi p$, we set $x_{\pi'} = \mathcal{O}(\Gamma, c_p, L \setminus \pi', \{x_{\pi' q}\}_{q \in L \setminus \pi'})$. Moreover, it holds that $x_\varnothing = \mathcal{O}(\Gamma, c_\lambda, L, \{x_{\{p\}}\}_{p \in L})$. Notice that $x_{\pi p} = x_{\pi' p}$ for every $p \in L$ and $\pi, \pi' \in$

	$s_{2,1}$	$s_{2,2}$
$s_{1,1}$	$1,2,0$	$0,1,0$
$s_{1,2}$	$1,2,0$	$0,1,0$

$s_{3,1}$

	$s_{2,1}$	$s_{2,2}$
$s_{1,1}$	$2,1,0$	$2,1,0$
$s_{1,2}$	$1,0,0$	$1,0,0$

$s_{3,2}$

	$s_{2,1}$	$s_{2,2}$
$s_{1,1}$	$2,2,0$	$2,2,0$
$s_{1,2}$	$2,2,0$	$0,0,10$

$s_{3,3}$

Table 9.2: *Three-player normal-form SG showing that, when searching for an optimal SCE-PA, it is necessary to consider the last leader who performed* OPT-OUT *(players 1, 2, and 3 select rows, columns, and matrices, respectively).*

$\Pi_{L\setminus\{p\}}$ with π and π' defining the same set. Thus, the number of queries to $\mathcal{O}$ is $\sum_{i=1}^{|L|} |L|\binom{|L|-1}{i-1}+1 = |L|2^{|L|-1}+1$. Clearly, by definition of $\mathcal{O}$, all the incentive constraints of Equation (2.3) are satisfied. Furthermore, it is easy to check that x_π is stable for every $\pi \in \Pi_L$. As a result, we can conclude that $\mathbf{x} \in \mathbf{X}^{\text{PS}}$. Now, we prove that $\mathbf{x}$ maximizes the objective f_λ over the set $\mathbf{X}^{\text{PS}}$. This also proves the efficiency of $\mathbf{x}$, and, thus, $\mathbf{x} \in \mathbf{X}^{\text{SCE-PA}}$. By contradiction, suppose there exists another $\mathbf{x}' = [x'_\pi] \in \mathbf{X}^{\text{SCE-PA}}$ with objective value $f'_\lambda > f_\lambda$. Three cases are possible:

- there exist $p \in L$ and $\pi \in \Pi_{L\setminus\{p\}}$ with $\pi p = L$ such that $u_p(x'_{\pi p}) < u_p(x_{\pi p})$;
- there exist $p \in L$ and $\pi \in \Pi_{L\setminus\{p\}}$ with $\pi p \neq L$ such that $u_p(x'_{\pi p}) < u_p(x_{\pi p})$ and, letting $\pi' = \pi p$, $u_q(x'_{\pi' q}) \geq u_q(x_{\pi' q})$ for all $q \in L \setminus \pi'$;
- $u_p(x'_{\{p\}}) \geq u_p(x_{\{p\}})$ for all $p \in L$.

All the three cases contradict the correctness of $\mathcal{O}$. □

Finally, we can provide an example showing that Theorem 9.5 is tight, which leads to the following proposition.

Proposition 9.1. *Solving* o-SCE-PA(λ) *requires to take into account the last player who performed* OPT-OUT, *while focusing only on the set of defecting leaders is not sufficient.*

Proof. Consider the SG in Table 9.2, with $L = \{1,2,3\}$ and $F = \varnothing$. There exists an $\mathbf{x} = [x_\pi] \in \mathbf{X}^{\text{SCE-PA}}$ such that $x_\varnothing(s_{1,2}, s_{2,2}, s_{3,3}) = 1$, and the same holds for $x_{\{1\}}(s_{1,1}, s_{2,2}, s_{3,1})$, $x_{\{2,1\}}(s_{1,2}, s_{2,2}, s_{3,1})$, $x_{\{2\}}(s_{1,2}, s_{2,1}, s_{3,2})$, and $x_{\{1,2\}}(s_{1,2}, s_{2,1}, s_{3,2})$. Moreover, for every $\pi \in \Pi_L$ including player 3, $x_\pi(s_{1,1}, s_{2,1}, s_{3,3}) = 1$. Notice that x_π depends on the last player who decides to OPT-OUT since $x_{\{1,2\}} \neq x_{\{2,1\}}$. We show that there is no $\mathbf{x}' = [x'_\pi] \in \mathbf{X}^{\text{SCE-PA}}$ where x_π does not depend on the last leader to

OPT-OUT and $x'_{\varnothing}(s_{1,2}, s_{2,2}, s_{3,3}) = 1$. Assume, by contradiction, that there exists such $\mathbf{x}'$. If player 1 performs OPT-OUT, she will get more than 0, unless only the strategy profiles $(s_{1,2}, s_{2,2}, s_{3,3})$, $(s_{1,1}, s_{2,2}, s_{3,1})$, and $(s_{1,2}, s_{2,2}, s_{3,1})$ are recommended by $x'_{\{1\}}$. The other strategy profiles providing player 1 with a utility of 0 cannot be recommended, otherwise incentive constraints of Eq. (2.3) are not satisfied. As a result, only strategy profiles $(s_{1,1}, s_{2,1}, s_{3,2})$ and $(s_{1,1}, s_{2,2}, s_{3,2})$ are recommended in $x'_{\{1,2\}}$ (otherwise player 2 would have an incentive to OPT-OUT). Instead, consider the case in which player 2 performs OPT-OUT. Since players 1 and 2 are symmetric, $x'_{\{2,1\}}$ can only recommend strategy profiles $(s_{1,1}, s_{2,1}, s_{3,1})$ and $(s_{1,2}, s_{2,1}, s_{3,1})$. Thus, $x'_{\{2,1\}}$ must be different from $x'_{\{1,2\}}$, a contradiction. □

9.2 Stability Oracle for Compact Games

We study which classes of games admit a polynomial-time stability oracle $\mathcal{O}$, focusing on those with polynomial type. [4]

Inspired by the classical approaches for finding CEs in games with polynomial type (Papadimitriou and Roughgarden, 2008; Jiang and Leyton-Brown, 2011, 2015), we solve $\mathcal{O}(\Gamma, c, L, \{x_p\}_{p \in L' \subseteq L})$ in polynomial time using the ellipsoid method. This requires that a suitably defined separation problem ($\mathsf{Sep}(z,t)$) can be computed in polynomial time. Our main result is that $\mathsf{Sep}(z,t)$ can be reduced to the *weighted deviation-adjusted social welfare problem* ($\mathsf{w\text{-}DaSW}(y, v, t)$) introduced by Jiang and Leyton-Brown (2011) for finding an optimal (according to some linear function of players' utilities) CE. This establishes a strict connection between the problem solved by our stability oracle and that of computing optimal CEs. As a consequence, given the results of Jiang and Leyton-Brown (2011), $\mathcal{O}$ can be computed in polynomial time for all the compact games where finding an optimal CE is computationally tractable. Thus:

Theorem 9.6. *The following games admit a polynomial-time stability oracle $\mathcal{O}$: anonymous games, symmetric games, and bounded-treewidth graphical and polymatrix games.*

Finally, our results also imply that the polynomial-time stability oracle $\mathcal{O}$ always outputs a polynomially-sized correlated distribution (see Corollary 9.7.1).

Next, we provide a complete proof of Theorem 9.6.

[4] We remark that, for normal-form games, a polynomial-time stability oracle $\mathcal{O}$ can be implemented by using a variation of the LP for finding optimal CEs (Shoham and Leyton-Brown, 2008).

For the ease of presentation, we treat $x \in \mathcal{X}$ as an $|S|$-dimensional vector. Moreover, given $c = [c_p] \in [-1,1]^n$, let $w = [w_s] \in \mathbb{R}^{|S|}$ be a vector with $w_s = \sum_{p \in N} c_p u_p(s)$, and, given a collection of correlated distributions $\{x_p\}_{p \in L' \subseteq L}$, let $b_p = \sum_{s \in S} u_p(s) x_p(s)$ for every $p \in L' \subseteq L$.

The solutions returned by $\mathcal{O}(\Gamma, c, L, \{x_p\}_{p \in L' \setminus L})$ are the optimal solutions to the following LP:

$$\mathfrak{P} \; : \begin{cases} \max & w^T x \\ \text{s.t.} & Ux \geq 0 \\ & \mathbf{1}^T x = 1, \; x \geq 0, \end{cases}$$

where U is a matrix of dimensions $C \times |S|$ (with $C = \sum_{p \in N \setminus L} |S_p|^2 + |L|$) encoding the coefficients of the incentive constraints of Equation (2.3) for the players in $N \setminus L$, and those of the additional stability constraints, *i.e.*, for every $p \in L$

$$\sum_{s \in S} (u_p(s) - b_p)\, x(s) \geq 0.$$

We denote with U_s the column of U corresponding to $s \in S$.

We can write the dual of problem $\mathfrak{P}$ as:

$$\mathfrak{D} \; : \begin{cases} \min & t \\ \text{s.t.} & U^T z + w \leq t\mathbf{1} \\ & z \geq 0, \end{cases}$$

where $z = [z^p_{s_p, s'_p}; z_p] \in \mathbb{R}^C$ is a vector of dual variables: $z^p_{s_p, s'_p}$ for all $p \in N \setminus L$ and $s_p, s'_p \in S_p$, and z_p for all $p \in L$.

A separation problem for $\mathfrak{D}$ asks whether a given pair (z, t) is feasible, and if not, it calls for a hyperplane separating (z, t) from the feasible set. Following Jiang and Leyton-Brown (2011), we focus on a restricted form of separation, requiring a violated constraint for infeasible points. Formally:

Definition 9.5 ($\mathsf{Sep}(z, t)$). *Given a pair (z, t) such that $z \geq 0$, determine if there exists an $s \in S$ such that $(U_s)^T z + w_s > t$; if so output such an s.*

Notice that, for every $s \in S$,

$$(U_s)^T z = \sum_{p \in N \setminus L} \sum_{s'_p \in S_p} z^p_{s_p, s'_p} \left(u_p(s) - u_p(s'_p, s_{-p}) \right) + \\ + \sum_{p \in L} z_p \left(u_p(s) - b_p \right).$$

The following holds:

Theorem 9.7. *If* $\mathsf{Sep}(z,t)$ *can be solved in polynomial time, then* $\mathcal{O}$ *can be computed in polynomial time.*

Proof. Clearly, a polynomial-time algorithm for $\mathsf{Sep}(z,t)$ can be used as separation oracle in the ellipsoid method, solving $\mathfrak{D}$ in polynomial time. By duality, the optimal objective for $\mathfrak{D}$ is the value $w^T x$ of a solution $x \in \mathcal{X}$ for $\mathcal{O}$. Since we required that separating hyperplanes be constraints for $\mathfrak{D}$, they can be used to compute such solution x. □

Corollary 9.7.1. $\mathcal{O}$ *returns a polynomially-sized* $x \in \mathcal{X}$.

Proof. This is a direct consequence of the fact that the ellipsoid method, as applied in Theorem 9.7, generates a polynomial number of violated constraints. □

Now, we introduce some definitions from (Jiang and Leyton-Brown, 2011). Given a finite game Γ, we let $y = [y^p_{s_p,s'_p}] \in \mathbb{R}^{C'}$ (with $C' = \sum_{p\in N} |S_p|^2$) be a vector indexed by $p \in N$ and $s_p, s'_p \in S_p$. Moreover, we let $v = [v_p] \in \mathbb{R}^n$ be a vector indexed by $p \in N$.

Definition 9.6. *Given a finite game* Γ*, a vector* $y \in \mathbb{R}^{C'}$ *such that* $y \geq 0$*, and a vector* $v \in \mathbb{R}^n$*, the* weighted deviation-adjusted utility *for player* $p \in N$ *in* $s \in S$ *is:*

$$\hat{u}^p_s(y,v) = v_p u_p(s) + \sum_{s'_p \in S_p} y^p_{s_p,s'_p} \left(u_p(s) - u_p(s'_p, s_{-p}) \right),$$

and the weighted deviation-adjusted social welfare *is defined as* $\hat{w}_s(y,v) = \sum_{p\in N} \hat{u}^p_s(y)$.

The following is the formal definition of *weighted deviation-adjusted social welfare problem.* [5]

Definition 9.7 ($\mathsf{w\text{-}DaSW}(y,v,t)$)**.** *Given a triplet* (y,v,t) *such that* $y \geq 0$*, determine if there exists an* $s \in S$ *such that* $\hat{w}_s(y,v) > t$*; if so output such an* s.

Our main result is the following:

Theorem 9.8. $\mathsf{Sep}(z,t)$ *reduces to* $\mathsf{w\text{-}DaSW}(y,v,t)$.

[5] The version proposed by Jiang and Leyton-Brown (2011) adds the additional constraints that $v_p \geq 0$ and $\sum_{p\in N} v_p = 1$.

Proof. Given (z, t) with $z \geq 0$, asking $(U_s)^T z + w_s > t$ is equivalent to asking

$$\sum_{p \in N \setminus L} \sum_{s'_p \in S_p} y^p_{s_p, s'_p} \left(u_p(s) - u_p(s'_p, s_{-p}) \right) + \\ + \sum_{p \in L} (c_p + y_p)\, u_p(s) + \sum_{p \in N \setminus L} c_p u_p(s) - \sum_{p \in L} y_p b_p > t.$$

In turn, this is equivalent to solving w-DaSW$(\hat{y}, \hat{v}, \hat{t})$ with:

- $\hat{y}^p_{s_p, s'_p} = 0$ for all $p \in L$, $s_p, s'_p \in S_p$;
- $\hat{y}^p_{s_p, s'_p} = y^p_{s_p, s'_p}$ for all $p \in N \setminus L$, $s_p, s'_p \in S_p$;
- $\hat{t} = t + \sum_{p \in L} y_p b_p$;
- $\hat{v}_p = c_p + y_p$ for all $p \in L$;
- $\hat{v}_p = c_p$ for all $p \in N \setminus L$.

This concludes the proof. □

In conclusion, the results in (Jiang and Leyton-Brown, 2011) together with Theorem 9.8 prove Theorem 9.6.

Part III

Trembling-Hand Perfection in Stackelberg Games

CHAPTER 10

Trembling-Hand Perfection in Extensive-Form Stackelberg Games

In this chapter, we initiate the study of trembling-hand perfection in the context of extensive-form SGs, *i.e.*, we extend the Stackelberg paradigm to games in extensive form where the players might tremble, taking off-equilibrium actions with low-but-non-zero probability. In particular, here, we introduce a general approach to refine SEs in extensive-form SGs. Then, in the following Chapter 11, we focus on a specific kind of refinement that is based on the idea of quasi perfection introduced by Van Damme (1984).

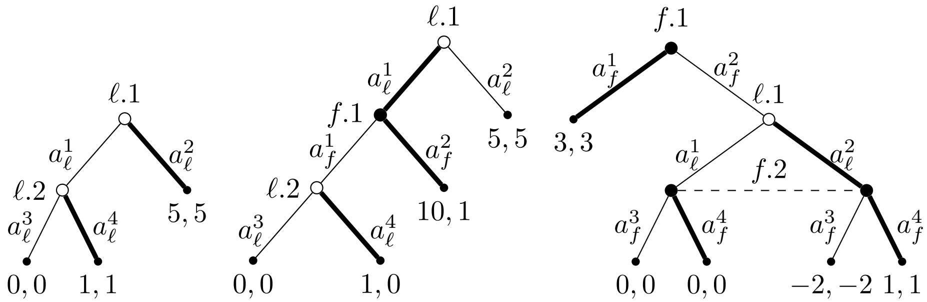
Initially, in Section 10.1, we provide some motivating examples showing that the refinement of classical solution concepts is needed also in two-player extensive-form SGs. Section 10.2 presents a general methodology to refine SEs by applying trembling-hand perfection in extensive-form SGs, introducing some gadgets, called *perturbation schemes*, which rely on the sequence form. Then, Section 10.3 shows our main result, *i.e.*, that the set of SEs is complete with respect to the limit points induced by perturbation schemes. Finally, Section 10.4 provides some computational complexity results about refinements of SEs.

10.1 Motivating Examples

In extensive-form games, classical equilibrium notions, such as, *e.g.*, NEs, may prescribe the players suboptimal actions off the equilibrium path. In the specific case of NEs, these weaknesses are amended by introducing equilibrium refinements based on trembles (see the discussion at the end of Subsection 2.3.1). Here, we show that the same problems (*i.e.*, non-optimality off the equilibrium path) arise in the context of *extensive-form Stackelberg games* (EFSGs). Then, in the following sections, we introduce trembles in EFSGs so as to refine SEs and overcome their weaknesses.

In the games in Figure 10.1, SEs, including as special cases SSEs and WSEs, may be suboptimal in presence of trembles. In particular, an SE may be suboptimal due to a leader's mistake (see Figure 10.1a), a follower's mistake (see Figure 10.1b), or both (see Figure 10.1c).

Moreover, the robust SE of Kroer et al. (2018) does not guard against trembles either. Consider the game in Fig. 10.1b: the leader should commit to (a_ℓ^1, a_ℓ^4) in order to get utility 10. However, (a_ℓ^1, a_ℓ^3) also achieves utility 10, but it is a worse strategy when trembles may happen. Adding payoff uncertainty on the $(1,0)$-payoff node such that the follower has a utility function where she picks a_f^1 does not solve this problem. The robust solution would pick a_ℓ^2 initially, since the worst-case follower picks a_f^1. In contrast, our perturbed SEs will uniquely identify (a_ℓ^1, a_ℓ^4) as the solution.



(a) *Any strategy at $\ell.2$ is optimal in an SE, while only a_ℓ^4 is optimal off the equilibrium path (i.e., if the leader trembles at $\ell.1$).*

(b) *Any leader's strategy at node $\ell.2$ is optimal in an SE, while only action a_ℓ^4 is optimal off the equilibrium path (notice that, here, $\ell.2$ is reached only if the follower trembles, playing a_f^1, at node $f.1$).*

(c) *Any strategy profile at nodes $\ell.1$ and $f.2$ is optimal in an SE, while only (a_ℓ^2, a_f^4) is optimal off the equilibrium path (notice that, in this example, both the leader and the follower may tremble at nodes $f.1$ and $\ell.1$, respectively).*

Figure 10.1: *Examples of EFSGs in which SEs may prescribe suboptimal actions off the equilibrium path.*

10.2 Game Perturbation Schemes

As discussed in Subsection 2.3.1, many of the most important NE refinement concepts are based on the idea that the player and/or the opponent makes mistakes at every decision point (*i.e.*, information set) with some small, vanishing probability. In this section, we introduce a more general family of such perturbations, of which the typical prior schemes, resulting in extensive-form perfection and quasi perfection, are subfamilies. Then, in the following sections, we develop a theory for the whole family and the subfamilies in the context of EFSGs. [1]

In the following definition, X is any sequence-form strategy polytope, *i.e.*, a set of valid realization plans, where no distinction is made based on to which player the polytope belongs.

Definition 10.1 (ϵ-Perturbation Scheme). *An* ϵ-perturbation scheme *for a strategy polytope* X *is a function* $\epsilon \mapsto X(\epsilon)$ *defined over* $\epsilon \in (0, 1]$ *with:*

- $X(\epsilon) \subseteq X(\epsilon')$ *for all* $\epsilon \geq \epsilon'$, *and*
- $\mathrm{cl}(\bigcup_{\epsilon\in(0,1]} X(\epsilon)) = X$.

The closure operation $\mathrm{cl}(\cdot)$ assumes that a topology is defined for the space containing X. We will always assume that the strategy polytopes X live in a Euclidean space where the usual metric induces open balls $B_\delta(\bar{x}) = \{x : \|x - \bar{x}\| < \delta\}$. The classical extensive-form-perfect and quasi-perfect perturbations (which we define formally in the following subsection) are two notable subfamilies of ϵ-perturbation schemes.

As a direct consequence of the conditions in Definition 10.1, every point in X is eventually "reached" by $X(\epsilon)$ when ϵ is small enough:

Lemma 10.1. *Given* $\bar{x} \in X$ *and* $\delta > 0$*, there exists* $\hat{x} \in X$ *and* $\bar{\epsilon} \in (0, 1]$ *such that* $\hat{x} \in X(\epsilon) \cap B_\delta(\bar{x})$ *for all* $\epsilon \leq \bar{\epsilon}$.

A *perturbed* EFSG is now simply an EFSG augmented with a perturbation scheme for each player:

Definition 10.2 (Perturbed EFSG). *A* perturbed EFSG *is defined as an EFSG, together with two* ϵ*-perturbation schemes* $\epsilon \mapsto R_\ell(\epsilon)$ *and* $\epsilon \mapsto R_f(\epsilon)$ *for the leader's and the follower's strategy polytope, respectively.*

Given a perturbed EFSG $(\Gamma, \epsilon \mapsto R_\ell(\epsilon), \epsilon \mapsto R_f(\epsilon))$, we denote by $\Gamma(\epsilon)$ the EFSG obtained from Γ by letting the leader and follower strategy polytopes be $R_\ell(\epsilon)$ and $R_f(\epsilon)$, respectively.

[1] Our perturbation family applies to any strategy polytope, not just EFSGs, and not even just the sequence form. That said, in the following, we assume that the game is in sequence form.

10.2.1 Extensive-Form-Perfect and Quasi-Perfect Perturbations

Now, we analyze perturbed EFSGs with extensive-form-perfect and quasi-perfect perturbations. These are EFSGs which are augmented with particular ϵ-perturbation schemes with specific structure.

In an extensive-form-perfect ϵ-perturbation scheme, each player takes into account the possibility that all the players, including herself, may make mistakes in the future. Players are constrained to placing at least a minimum probability α on every action at each information set, and those lower bounds α are functions of ϵ that go to zero as $\epsilon \to 0$. A formal definition, using sequence-form strategy polytopes, follows.

Definition 10.3 (Extensive-Form-Perfect ϵ-Perturbation Scheme). *An* extensive-form-perfect ϵ-perturbation scheme *for a sequence-form strategy polytope R_p of player $p \in N$ is an ϵ-perturbation scheme $\epsilon \mapsto R_p^{\mathrm{EFP}}(\epsilon)$ where a realization plan r_p belongs to $R_p^{\mathrm{EFP}}(\epsilon)$ if:*

- $r_p(\sigma_p) \geq \alpha(\epsilon, \sigma_p) r_p(\sigma'_p)$ *for* $\sigma_p, \sigma'_p \in \Sigma_p : \sigma_p = \sigma'_p a$ *for some* $a \in A_p$*;*
- $\alpha(\epsilon, \sigma_p) \geq 0$ *and* $\lim_{\epsilon \to 0^+} \alpha(\epsilon, \sigma_p) = 0$ *for all* $\sigma_p \in \Sigma_p$*;*
- $\sum_{\sigma_p \in \Sigma_p | \sigma_p = \sigma'_p a} \alpha(\epsilon, \sigma_p) \leq 1$ *for all* $\sigma'_p \in \Sigma_p$.

In a quasi-perfect ϵ-perturbation scheme, each player takes into consideration only the possibility of opponent's errors, assuming she will not make mistakes in future. This is modeled by requiring that sequences σ_p be played with probabilities at least $\xi_p(\epsilon, \sigma_p)$. In words, the lower-bounds on sequence probabilities enjoy the following properties: *(i)* they are polynomials in the variable ϵ; *(ii)* they approach zero as ϵ goes to zero; and *(iii)* $\xi_p(\epsilon, \sigma_p(I)a)$ approaches zero faster than $\xi_p(\epsilon, \sigma_p(I))$. Formally:

Definition 10.4 (Quasi-Perfect ϵ-Perturbation Scheme). *A* quasi-perfect ϵ-perturbation scheme *for a sequence-form strategy polytope R_p of player $p \in N$ is an ϵ-perturbation scheme $\epsilon \mapsto R_p^{\mathrm{QP}}(\epsilon)$ where a realization plan r_p belongs to $R_p^{\mathrm{QP}}(\epsilon)$ if $r_p(\sigma_p) \geq \xi_p(\epsilon, \sigma_p)$ for every $\sigma_p \in \Sigma_p$, and, additionally, $\xi_p : (0, 1] \times \Sigma_p \to \mathbb{R}^+$ is a function such that:*

1. *$\xi_p(\epsilon, \sigma_p)$ is a polynomial in ϵ, for all $\sigma_p \in \Sigma_p$;*
2. $\lim_{\epsilon \to 0^+} \xi_p(\epsilon, \sigma_p) = 0$*, for all* $\sigma_p \in \Sigma_p \setminus \{\sigma_\emptyset\}$*;*
3. $\lim_{\epsilon \to 0^+} \dfrac{\xi_p(\epsilon, \sigma_p(I)a)}{\xi_p(\epsilon, \sigma_p(I))} = 0$*, for all* $I \in \mathcal{I}_p, a \in A(I)$.

Now, we introduce new solution concepts defined as limit points of sequences of SSEs and WSEs for perturbed game instances $\Gamma(\epsilon)$ as $\epsilon \to 0$, given particular perturbation schemes.

Definition 10.5 (Extensive-Form-Perfect SEs)**.** *Given a perturbed EFSG* $(\Gamma, \epsilon \mapsto R_\ell^{\text{EFP}}(\epsilon), \epsilon \mapsto R_f^{\text{EFP}}(\epsilon))$*,* $(r_\ell, r_f) \in R_\ell \times R_f$ *is an* extensive-form-perfect SSE (EFP-SSE) *(respectively,* extensive-form-perfect WSE (EFP-WSE)*) if it is a limit point of SSEs (respectively, WSEs) of* $\Gamma(\epsilon)$ *as* $\epsilon \to 0$.

Definition 10.6 (Quasi-Perfect SEs)**.** *Given a perturbed EFSG* $(\Gamma, \epsilon \mapsto R_\ell^{\text{QP}}(\epsilon), \epsilon \mapsto R_f^{\text{QP}}(\epsilon))$*,* $(r_\ell, r_f) \in R_\ell \times R_f$ *is an* quasi-perfect SSE (QP-SSE) *(respectively,* quasi-perfect WSE (QP-WSE)*) if it is a limit point of SSEs (respectively, WSEs) of* $\Gamma(\epsilon)$ *as* $\epsilon \to 0$. [2]

Since SSEs always exist in an EFSG, and since the strategy spaces are compact sets, EFP-SSEs and QP-SSEs always exist. The same is not true for EFP- and QP-WSEs, as a WSE need not exist in an EFSG.

The following proposition shows that the sets of EFP- or QP-SSEs can be disjoint from the set of SSEs, thereby showing that the EFP- and QP-SSE solution concepts are *not* refinements of the SSE solution concept!

Proposition 10.1. *There are perturbed EFSGs in which an EFP-SSE is not an SSE, a QP-SSE is not an SSE, and an EFP-WSE is not a WSE.*

Proof. Consider the game of Figure 10.2a. The SSE prescribes the leader and the follower to play a_ℓ^1 and a_f^1, respectively. On the other hand, in any perturbed instance the leader has to place positive probability on a_ℓ^2, and the follower's best response is a_f^2.

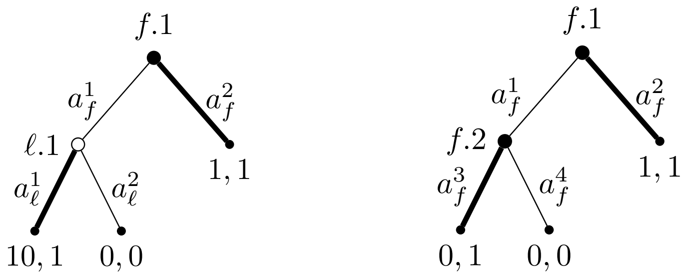
Consider the game of Figure 10.2b. The follower plays $a_f^1 a_f^3$, while in any perturbed instance resulting from an extensive-form-perfect ϵ-perturbation scheme, the follower has to put positive probability on a_f^4 and her best response at the root becomes a_f^2. □

We leave as an open problem the determination of whether a QP-WSE is also a WSE (assuming it exists) or not.

10.3 Stackelberg Trembling-Hand Refinements

As we showed in the previous section, SSEs and WSEs are not refinable by trembling. In this section we remedy this problem by showing that the

[2] Here, the definition of quasi-perfect SE is given directly in terms of perturbation schemes. In the following Chapter 11, we provide an axiomatic definition that does not rely on perturbation schemes (see Definition 11.4).



(a) *Game in which perturbed SSEs are not refinements of SSEs.*

(b) *Game in which perturbed WSEs are not refinements of WSEs.*

Figure 10.2: *Games that we use to prove that perturbed SSEs and WSEs are not refinements of SSEs and WSEs.*

universal set of all Stackelberg equilibria is natural for trembling-hand perfection: it does not suffer from the problem above. In other words, the set of SEs is closed under trembling-hand refinement. Formally, we prove that any limit point of SEs for the perturbed game $\Gamma(\epsilon)$ as $\epsilon \to 0$ is an SE of the original, unperturbed EFSG Γ.

Theorem 10.1. *Let $\{\epsilon_i\} \to 0$ and let $\{(r_{\ell i}, r_{f_i})\}$ be a sequence of SEs for the perturbed game instances $\{\Gamma(\epsilon_i)\}$. Then:*

- *$\{(r_{\ell i}, r_{f_i})\}$ has at least one limit point, and*
- *all limit points of $\{(r_{\ell i}, r_{f_i})\}$ are SEs.*

We now present three lemmas, and at the end of this section we present the proof of the theorem using these lemmas.

One can think of SEs as "minimally-rational" for the leader: any strategy for the leader is acceptable as long as there is no other strategy for the leader that is better *no matter how the follower breaks ties*. We now formalize this by giving the following alternative characterization of SEs.

Lemma 10.2. *A strategy profile $(r_\ell, r_f) \in R_\ell \times R_f$ is an SE if and only if $r_f \in \mathsf{BR}(r_\ell)$ and for all $r'_\ell \in R_\ell$ there exists $r'_f(r'_\ell) \in \mathsf{BR}(r'_\ell)$ such that $u_\ell(r'_\ell, r'_f(r'_\ell)) \leq u_\ell(r_\ell, r_f)$.*

Proof. ($\Leftarrow$) Construct the follower response function τ defined as $\tau(r_\ell) = r_f$ and $\tau(r'_\ell) = r'_f(r'_\ell)$. Then, for all $r'_\ell \in R_\ell$, $u_\ell(r'_\ell, \tau(r'_\ell)) \leq u_\ell(r_\ell, \tau(r_\ell))$, and thus (r_ℓ, r_f) is a τ-SE.

($\Rightarrow$) Assume that (r_ℓ, r_f) is a τ-SE. Then $r_f = \tau(r_\ell) \in \mathsf{BR}(r_\ell)$. Furthermore, by definition of τ-SE, for all $r'_\ell \in R_\ell$, $r'_f = \tau(r'_\ell)$ is such that $u_\ell(r'_\ell, r'_f) \leq u_\ell(r_\ell, \tau(r_\ell)) = u_\ell(r_\ell, r_f)$. □

Lemma 10.3. *Let $\{\epsilon_i\} \to 0$ and let $\{(r_{\ell i}, r_{f_i})\}$ be a sequence of strategy profiles for the EFSG instances $\{\Gamma(\epsilon_i)\}$. Then $\{(r_{\ell i}, r_{f_i})\}$ has at least one limit point.*

Proof. The conclusion follows directly from the Bolzano-Weierstrass theorem since $R_\ell(\epsilon) \times R_f(\epsilon) \subseteq R_\ell \times R_f$ for all $\epsilon \in (0, 1]$ and $R_\ell \times R_f$ is a compact set. $\square$

Lemma 10.4. *Let $\{\epsilon_i\} \to 0$ and let $\{(r_{\ell i}, r_{f_i})\}$ be a sequence of strategy profiles for the EFSG instances $\{\Gamma(\epsilon_i)\}$ where r_{f_i} is a best response to $r_{\ell i}$. Then, any limit point $(\bar{r}_\ell, \bar{r}_f)$ of $\{(r_{\ell i}, r_{f_i})\}$ is such that $\bar{r}_f$ is a best response to $\bar{r}_\ell$.*

Proof. Existence of at least one limit point for $\{(r_{\ell i}, r_{f_i})\}$ is guaranteed by Lemma 10.3. Without loss of generality, assume that $(r_{\ell i}, r_{f_i}) \to (\bar{r}_\ell, \bar{r}_f) \in R_\ell \times R_f$. Suppose, for contradiction, that $\bar{r}_f$ is *not* a best response to $\bar{r}_\ell$ which means that there exists $\hat{r}_f \in R_f$ such that $u_f(\bar{r}_\ell, \hat{r}_f) > u_f(\bar{r}_\ell, \bar{r}_f)$. By continuity of u_f, there exists $\delta > 0$ such that $u_f(r_\ell^>, r_f^>) > u_f(r_\ell^<, r_f^<)$ for all $(r_\ell^>, r_f^>) \in B_\delta(\bar{r}_\ell) \times B_\delta(\hat{r}_f)$ and $(r_\ell^<, r_f^<) \in B_\delta(\bar{r}_\ell) \times B_\delta(\bar{r}_f)$. From Lemma 10.1 we know that there exist $\bar{\epsilon} \in (0, 1]$ and $\tilde{r}_f$ such that $\tilde{r}_f \in R_f(\epsilon) \cap B_\delta(\hat{r}_f)$ for all $\epsilon \le \bar{\epsilon}$. Considering the three converging sequences $\epsilon_i \to 0$, $r_{\ell i} \to \bar{r}_\ell$ and $r_{f_i} \to \bar{r}_f$, we know that there exists an index $j \in \mathbb{N}$ such that $\epsilon_j \le \bar{\epsilon}, r_{\ell j} \in B_\delta(\bar{r}_\ell)$, and $r_{f_j} \in B_\delta(\bar{r}_f)$. Furthermore, from $\epsilon_j \le \bar{\epsilon}$ we deduce that $(r_{\ell j}, \tilde{r}_f) \in R_\ell(\epsilon_j) \times R_f(\bar{\epsilon}) \subseteq R_\ell(\epsilon_j) \times R_f(\epsilon_j)$. Thus $(r_{\ell j}, \tilde{r}_f)$ is a valid strategy profile for $\Gamma(\epsilon_j)$. Yet, $(r_{\ell j}, \tilde{r}_f) \in B_\delta(\bar{r}_\ell) \times B_\delta(\hat{r}_f)$ and $(r_{\ell j}, r_{f_j}) \in B_\delta(\bar{r}_\ell) \times B_\delta(\bar{r}_f)$, implying $u_f(r_{\ell j}, \tilde{r}_f) > u_f(r_{\ell j}, r_{f_j})$ and contradicting the fact that r_{f_j} is a best response to $r_{\ell j}$. $\square$

Proof of Theorem 10.1. The first bullet is by Lemma 10.3. We now prove the second one. Let $\mathsf{BR}_\Gamma(r_\ell)$ and $\mathsf{BR}_{\Gamma(\epsilon_i)}(r_\ell)$ be the sets of follower's best responses to r_ℓ in Γ and $\Gamma(\epsilon_i)$, respectively. Without loss of generality, assume that $\{r_{\ell i}, r_{f_i}\} \to (\bar{r}_\ell, \bar{r}_f) \in R_\ell \times R_f$. By Lemma 10.4, $\bar{r}_f$ is a best response to $\bar{r}_\ell$. Therefore, by Lemma 10.2, we only need to prove that for all r'_ℓ there exists $r'_f \in \mathsf{BR}_\Gamma(r'_\ell)$ with $u_\ell(r'_\ell, r'_f) \le u_\ell(\bar{r}_\ell, \bar{r}_f)$.

Suppose for contradiction that there exists r'_ℓ such that $u_\ell(r'_\ell, r'_f) > u_\ell(\bar{r}_\ell, \bar{r}_f)$ for all $r'_f \in \mathsf{BR}_\Gamma(r'_\ell)$. Let g_i be the family of functions with the property that, for all i, $g_i(r_\ell)$ is equal to one of the $r_f \in \mathsf{BR}_{\Gamma(\epsilon_i)}(r_\ell)$ such that $u_\ell(r_\ell, r_f) \le u_\ell(r_{\ell i}, r_{f_i})$; existence is guaranteed by Lemma 10.2 and the fact that $(r_{\ell i}, r_{f_i})$ is an SE by hypothesis. Construct any sequence $\{(r'_{\ell i}, r'_{f_i})\} \to (r'_\ell, \hat{r}'_f)$ such that $(r'_{\ell i}, r'_{f_i})$ are valid profiles for $\Gamma(\epsilon_i)$ with $r'_{f_i} = g_i(r'_{\ell i})$. From Lemma 10.4 we know that $\hat{r}'_f \in \mathsf{BR}_\Gamma(r'_\ell)$. However, $u_\ell(r'_{\ell i}, r'_{f_i}) \le u_\ell(r_{\ell i}, r_{f_i})$, and by continuity we have $u_\ell(r'_\ell, \hat{r}'_f) \le$

$u_\ell(\bar{r}_\ell, \bar{r}_f)$. But then $\hat{r}'_f \in \mathsf{BR}_\Gamma(r'_\ell)$ while having value no larger than $u_\ell(\bar{r}_\ell, \bar{r}_f)$, contradicting our assumption. □

Because SSEs and WSEs are SEs, we have from Theorem 10.1 that the limit of perturbed SSEs (and perturbed WSEs when they exist) is *guaranteed* to be an SE. This means that, even though the limits are not SSEs or WSEs in general, they preserve minimal rationality of the commitment—as per Lemma 10.2.

10.4 Computational Complexity of Refined SEs

In this section we study the computational complexity of deciding the existence of an SE (refined or not) that gives the leader expected value at least ν. This problem (in the unrefined case) is known to be polynomial in constant-sum settings, where all SEs give the same expected utility to the leader, equal to the value of the game. We show that this problem is NP-hard in general-sum settings, using a reduction from 3-SAT. In particular, given a 3-SAT formula, we construct a polynomially-large EFSG instance such that:

- If the 3-SAT formula is satisfiable, all SEs of the EFSG give an expected utility of 1 to the leader.
- If the 3SAT formula is *not* satisfiable, all SEs of the EFSG give an expected utility strictly less than 1 to the leader. [3]

Since the 3-SAT decision problem is NP-hard (Johnson and Garey, 1979), this implies the following theorems.

Theorem 10.2. *Deciding the existence of an SE (refined or not) that gives the leader expected value at least ν in an EFSG is* NP*-hard.*

Theorem 10.3. *Given a follower response function τ, deciding the existence of a τ-SE (refined or not) that gives the leader expected value at least ν in an EFSG is* NP*-hard.*

10.4.1 EFSG Instance Construction

Definition 10.7. *We are given a* 3-SAT *formula (C, V), where C is a set of three-literal clauses defined over a set V of variables. We construct a perfect-recall EFSG $\Gamma(C, V)$ as follows:*

[3] Our reduction is based on the construction of Letchford and Conitzer (2010). However, their construction only proves the NP-hardness for the special case of SSEs, since, whenever the 3-SAT formula is satisfiable, there are SEs of the EFSG that provide the leader with an expected utility strictly less than 1. We suitably modify players' payoffs so that the result holds for all SEs.

- *The root is* $h_f^0 \in H_f$ *such that* $\rho(h_f^0) = \{a_f^t\} \cup \{a_f^v : v \in V\} \cup \{a_f^\phi : \phi \in C\}$, $\chi(h_f^0, a_f^x) = h_\ell^{1,x} \in H_\ell$.
- *All nodes* $h_\ell^{1,x}$ *belong to* $I \in \mathcal{I}_\ell$. *The available actions at the information set are* $\rho(I) = \{a_\ell^v : v \in V\}$.
- *For all* $h_\ell^{1,v}$ *and* a_ℓ^w $(v, w \in V)$, *we let* $\chi(h_\ell^{1,v}, a_\ell^w) = z_{vw} \in Z$. *Furthermore,* $u_\ell(z_{vw}) = 0$, *and* $u_f(z_{vw}) = \left(\frac{|V|+2}{(|V|+1)^2} + 1\right) - \mathbf{1}\{v = w\}$.
- *For all* a_ℓ^v $(v \in V)$, $\chi(h_\ell^{1,t}, a_\ell^v) = h_f^{2,v} \in H_f$, $\rho(h_f^{2,v}) = \{a_f^{v,\mathrm{T}}, a_f^{v,\mathrm{F}}\}$, $\chi(h_f^{2,v}, a_f^{v,x}) = h_\ell^{3,vx} \in I_v \in \mathcal{I}_\ell$, $\rho(I_v) = \{a_\ell^{v,\mathrm{T}}, a_\ell^{v,\mathrm{F}}\}$, $\chi(h_\ell^{3,vx}, a_\ell^{v,y}) = z_{vxy} \in Z$, $u_\ell(z_{vxy}) = u_f(z_{vxy}) = \mathbf{1}\{x = y\}$.
- *For all* $h_\ell^{1,\phi}$ *and* a_ℓ^v $(\phi \in C, v \in V)$,

$$u_\ell(z_{\phi v}) = 0, u_f(z_{\phi v}) = 0, \chi(h_\ell^{2,\phi v}, a_\ell^{v,x}) = z_{\phi v x} \in Z,$$

$$\chi(h_\ell^{1,\phi}, a_\ell^v) = \begin{cases} h_\ell^{2,\phi v} \in I_v & \text{if } v \text{ is in } \phi \\ z_{\phi v} \in Z & \text{otherwise}, \end{cases}$$

and $u_\ell(z_{\phi v x}) = 0, u_f(z_{\phi v \mathrm{T}}) = \frac{|V|+1}{3}$ *(resp.,* $u_f(z_{\phi v \mathrm{F}}) = \frac{|V|+1}{3}$*) if* v *appears negated (resp., not negated) in* ϕ, $u_f(z_{\phi v x}) = 0$ *otherwise.*

Figure 10.3 shows an example of a game $\Gamma(C, V)$.

Intuitively, the leader looks for a strategy such that the follower's best response is to play action a_f^t, thus achieving an expected utility of 1. The leader's strategy at information sets I_v $(v \in V)$ defines a truth assignment to the variables such that, whenever a clause $\phi \in C$ is not satisfied, then the follower best-responds playing action a_f^ϕ. Thus, the leader's goal is to find a strategy that defines a truth assignment satisfying all clauses.

In the following, for the ease of presentation, given a player p's behavioral strategy $\pi_p \in \Pi_p$, we use the notation $\pi_p(a)$, rather than π_{pa}, to denote the probability of playing action $a \in A_p$.

First we show that when the 3-SAT formula is satisfiable there exists a leader's strategy that guarantees a payoff of 1.

Lemma 10.5. *If* (C, V) *is satisfiable, then there exists a leader's strategy* π_ℓ *such that for all follower's best responses* $\pi_f \in \mathsf{BR}(\pi_\ell)$ *it holds* $u_\ell(\pi_\ell, \pi_f) = 1$.

Proof. Let T be a truth assignment satisfying all clauses. Take π_ℓ such that $\pi_\ell(a_\ell^v) = \frac{1}{|V|}$ for all $v \in V$ and $\pi_\ell(a_\ell^{v,\mathrm{T}}) = 1$ if $T(v) = 1$, while $\pi_\ell(a_\ell^{v,\mathrm{F}}) = 1$

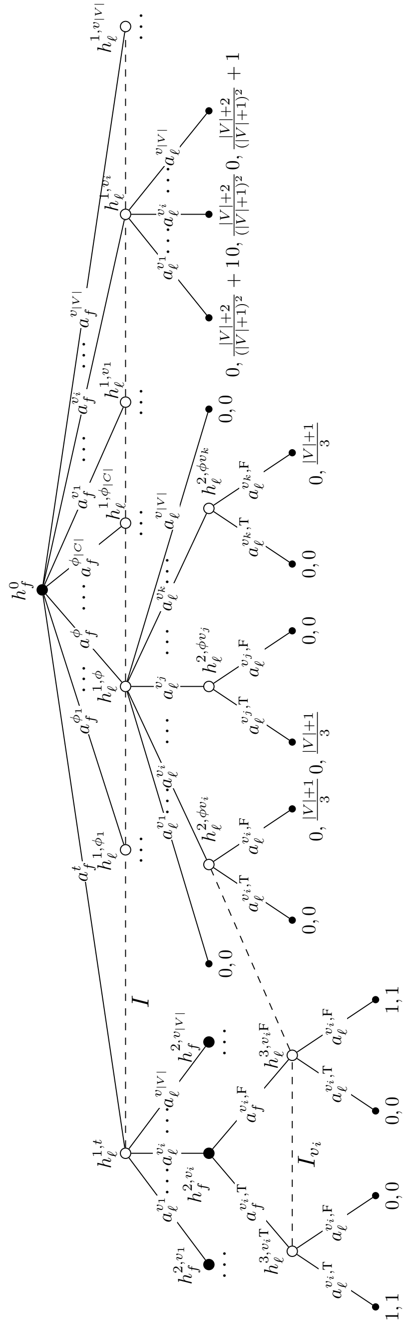


Figure 10.3: *Game $\Gamma(C, V)$, where $V = \{v_1, \ldots, v_{|V|}\}$, $C = \{\phi_1, \ldots, \phi_{|C|}\}$, and clause $\phi \in C$ is such that $\phi = v_i \bar{v}_j v_k$. Hollow nodes belong to the leader, while solid ones belong to the follower.*

whenever $T(v) = 0$. Clearly, at each node $h_f^{2,v}$, the unique follower's best response is to play the action corresponding to that played by the leader in I_v. As a result, the follower gets a utility of 1 by playing a_f^t at the root, *i.e.*, $u_f(\pi_\ell, a_f^t) = 1$. Now, let us prove that playing action a_f^t at the root is the unique follower's best-response to π_ℓ. Let us consider actions a_f^v, for all $v \in V$, we have:

$$u_f(\pi_\ell, a_f^v) = \left(\frac{|V|+2}{(|V|+1)^2} + 1\right)\frac{|V|-1}{|V|} + \left(\frac{|V|+2}{(|V|+1)^2}\right)\frac{1}{|V|}$$
$$= \left(\frac{|V|+2}{(|V|+1)^2} + 1\right) - \frac{1}{|V|} < 1.$$

Thus, playing a_f^v is not a best-response, for all $v \in V$. Now, we analyze actions a_f^ϕ, for all $\phi \in C$. Since T satisfies all clauses, each clause $\phi \in C$ has a literal l_k that is true under T and, thus, $\pi_\ell(a_\ell^{v(l_k),\mathrm{T}}) = 1$ if l_k requires the corresponding variable to be true, or $\pi_\ell(a_\ell^{v(l_k),\mathrm{F}}) = 1$ if it requires false. Assume, without loss of generality, that l_k requires the variable to be true for all $l_k \in \phi$. By playing a_f^ϕ, the follower gets utility $u_f(\pi_\ell, a_f^\phi) = \pi_\ell(a_\ell^{v(l_1)})\pi_\ell(a_\ell^{v(l_1),\mathrm{F}})\frac{|V|+1}{3} + \pi_\ell(a_\ell^{v(l_2)})\pi_\ell(a_\ell^{v(l_2),\mathrm{F}})\frac{|V|+1}{3} + \pi_\ell(a_\ell^{v(l_3)})\pi_\ell(a_\ell^{v(l_3),\mathrm{F}})\frac{|V|+1}{3}$. Three cases are possible.

1. There exists unique $l_k \in \phi$ such that $\pi_\ell(a_\ell^{v(l_k),\mathrm{T}}) = 1$, for instance literal l_1. Thus, since $\pi_\ell(a_\ell^{v(l_2),\mathrm{F}}), \pi_\ell(a_\ell^{v(l_3),F}) \leq 1$, it holds $u_f(\pi_\ell, a_f^\phi) \leq \frac{|V|+1}{3}(\pi_\ell(a_\ell^{v(l_2)}) + \pi_\ell(a_\ell^{v(l_3)}))$. Also, $\pi_\ell(a_\ell^v) = \frac{1}{|V|}$ for all $v \in V$ implies $u_f(\pi_\ell, a_f^\phi) < \frac{|V|+1}{3}\left(1 - \frac{1}{|V|}(|V|-2)\right) = \frac{2}{3}\frac{|V|+1}{|V|} < 1$, for n sufficiently large ($|V| > 2$).

2. Exactly two literals l_k in ϕ are such that $\pi_\ell(a_\ell^{v(l_k),\mathrm{T}}) = 1$. With a similar reasoning, we have $u_f(\pi_\ell, a_f^\phi) < \frac{|V|+1}{3}\left(1 - \frac{1}{|V|}(|V|-1)\right) < 1$, for every $|V|$.

3. $\pi_\ell(a_\ell^{v(l_k),\mathrm{T}}) = 1$ for all literals $l_k \in \phi$, and $u_f(\pi_\ell, a_f^\phi) < 1$. Therefore, it must be $u_f(\pi_\ell, a_f^\phi) < 1$ and a_f^ϕ is not a follower's best-response to π_ℓ.

In conclusion, the unique follower's best response is to play a_f^t at node h_f^0, and $u_\ell(\pi_\ell, a_f^t) = 1$. $\square$

Given that 1 is the maximum leader's payoff in $\Gamma(C, V)$, we can conclude the following:

Corollary 10.3.1. *If (C, V) is satisfiable, then all SEs of $\Gamma(C, V)$ give the leader an expected utility of 1.*

We now show that a utility of 1 for the leader implies the existence of a truth assignment satisfying the 3-SAT formula.

Lemma 10.6. *If there exists a leader's strategy π_ℓ and a follower's best response $\pi_f \in \mathsf{BR}(\pi_\ell)$ such that $u_\ell(\pi_\ell, \pi_f) = 1$, then (C, V) is satisfiable.*

Proof. Since $u_\ell(\pi_\ell, \pi_f) = 1$, it must be the case that in π_f the follower plays a_f^t at the root node h_f^0, or else the leader would not get a utility of 1. Moreover, the leader's strategy π_ℓ must be such that either $\pi_\ell(a_\ell^{v,\mathrm{T}}) = 1$ or $\pi_\ell(a_\ell^{v,\mathrm{F}}) = 1$, for every $v \in V$, and, at each node $h_f^{2,v}$, the follower must play the action corresponding to that played by the leader in I_v. Because a_f^t is a best response, it must be that $u_f(\pi_\ell, a_f^v) \leq 1$ for every $v \in V$, otherwise a_f^t would not be a follower best response. From $u_f(\pi_\ell, a_f^v) \leq 1$, it follows that $u_f(\pi_\ell, a_f^v) = \frac{|V|+2}{(|V|+1)^2} + 1 - \pi_\ell(a_\ell^v) \leq 1$, so that $\pi_\ell(a_\ell^v) \geq \frac{|V|+2}{(|V|+1)^2} > \frac{1}{|V|+1}$ for every $|V|$. For every $\phi \in C$ we have $u_f(\pi_\ell, a_f^\phi) \leq 1$, otherwise playing a_f^t is not a best response for the follower. As a consequence, for every $\phi \in C$, there must exist at least one literal $l_k \in \phi$ such that $\pi_\ell(a_\ell^{v(l_k),\mathrm{T}}) = 1$ if l_k requires true, or $\pi_\ell(a_\ell^{v(l_k),\mathrm{F}}) = 1$ if l_k requires false. By contradiction, suppose such a literal l_k does not exist, and assume, without loss of generality, that l_k requires true for all $l_k \in \phi$. Thus, $u_f(\pi_\ell, a_f^\phi) = \pi_\ell(a_\ell^{v(l_1)})\pi_\ell(a_\ell^{v(l_1),\mathrm{F}})\frac{|V|+1}{3} + \pi_\ell(a_\ell^{v(l_2)})\pi_\ell(a_\ell^{v(l_2),\mathrm{F}})\frac{|V|+1}{3} + \pi_\ell(a_\ell^{v(l_3)})\pi_\ell(a_\ell^{v(l_3),\mathrm{F}})\frac{|V|+1}{3} = \frac{|V|+1}{3}(\pi_\ell(a_\ell^{v(l_1)}) + \pi_\ell(a_\ell^{v(l_2)}) + \pi_\ell(a_\ell^{v(l_3)})) > 1$, as $\pi_\ell(a_\ell^v) > \frac{1}{|V|+1}$ for all $v \in V$. This contradicts the fact that $u_f(\pi_\ell, a_f^\phi) \leq 1$. It follows that ϕ must be satisfied. Since ϕ was arbitrary, this shows that all clauses are satisfied. In conclusion, a variable assignment T such that $T(v) = 1$ if $\pi_\ell(a_\ell^{v,\mathrm{T}}) = 1$, while $T(v) = 0$ whenever $\pi_\ell(a_\ell^{v,\mathrm{F}}) = 1$, satisfies all clauses. □

It directly follows from Lemma 10.6 that:

Corollary 10.3.2. *If (C, V) is not satisfiable, then all SEs of $\Gamma(C, V)$ give an expected utility smaller than 1 to the leader.*

CHAPTER 11

Quasi-Perfect Stackelberg Equilibrium

In this chapter, we focus on a particular refinement of the SE, which is based on the idea of quasi perfection introduced by Van Damme (1984).

Initially, in Section 11.1, we provide an axiomatic definition of quasi-perfect SE, following the line of the original definition of Van Damme (1984). We also anticipate the main result of this chapter, *i.e.*, that the limit points (as $\epsilon \to 0$) of sequences of SEs in perturbed EFSGs obtained for some quasi-perfect ϵ-perturbation scheme are quasi-perfect SEs according to our axiomatic definition. Then, in Section 11.2, we first show some properties of EFSGs perturbed with quasi-perfect ϵ-perturbation schemes and, then, we prove our main result. Finally, Section 11.3 provides an algorithm to compute (approximate) quasi-perfect SSEs and experimentally evaluates it on some classical extensive-form game test instances.

11.1 Definition of Quasi-Perfect Stackelberg Equilibrim

We start providing an axiomatic definition of quasi-perfect SEs, *i.e.*, one that does not rely on perturbation schemes defined over the sequence form (as Definition 10.6), but, instead, it is directly concerned with the extensive form. Before that, we introduce alternative definitions for SEs and SSEs, which will help the reader to understand how our definition of quasi-perfect SEs works. In the following, for the ease of presentation, we denote with $\mathsf{BR}_\Gamma(\pi_\ell) \subseteq \Pi_f$ the set of follower's best responses to the leader's strategy $\pi_\ell \in \Pi_\ell$ in the game Γ.

Definition 11.1 (Stackelberg Equilibrium)**.** *Given an EFSG* Γ*,* (π_ℓ, π_f) *is an* SE *of* Γ *if* $\pi_f \in \mathsf{BR}_\Gamma(\pi_\ell)$ *and, for all* $\hat{\pi}_\ell \in \Pi_\ell$*, there exists* $\hat{\pi}_f \in \mathsf{BR}_\Gamma(\hat{\pi}_\ell)$ *such that* $u_\ell(\pi_\ell, \pi_f) \geq u_\ell(\hat{\pi}_\ell, \hat{\pi}_f)$.

Definition 11.2 (Strong Stackelberg Equilibrium)**.** *Given an EFSG* Γ*,* (π_ℓ, π_f) *is an* SSE *of* Γ *if* $\pi_f \in \mathsf{BR}_\Gamma(\pi_\ell)$ *and, for all* $\hat{\pi}_\ell \in \Pi_\ell$ *and* $\hat{\pi}_f \in \mathsf{BR}_\Gamma(\hat{\pi}_\ell)$*, it holds* $u_\ell(\pi_\ell, \pi_f) \geq u_\ell(\hat{\pi}_\ell, \hat{\pi}_f)$.

Notice that, using the equivalence between behavioral strategies and realization plans, SEs and SSEs can be defined analogously for EFSGs in sequence form.

Before introducing our axiomatic definition of quasi-perfect SE, we provide some useful, additional notation. We say that $\pi_p \in \Pi_p$ is *completely mixed* if $\pi_{pa} > 0$ for all $a \in \mathcal{A}_p$. Given two information sets $I, \hat{I} \in \mathcal{I}_p$, we write $I \succeq \hat{I}$ whenever $\hat{I}$ follows I, *i.e.*, there exists a path from $h \in I$ to $\hat{h} \in \hat{I}$. We assume $I_\varnothing \succeq \hat{I}$ for all $\hat{I} \in \mathcal{I}_p$ such that there is no $I \neq \hat{I} \in \mathcal{I}_p : I \succeq \hat{I}$. In perfect-recall games, $\succeq$ is a partial order over $\mathcal{I}_p \cup \{I_\varnothing\}$. Given $\pi_p, \hat{\pi}_p \in \Pi_p$ and $I \in \mathcal{I}_p \cup \{I_\varnothing\}$, $\pi_p /_I \hat{\pi}_p$ is equal to $\hat{\pi}_p$ at all $\hat{I} \in \mathcal{I}_p : I \succeq \hat{I}$, while it is equal to π_p everywhere else. Moreover, for $I \in \mathcal{I}_p$, we write $\pi_p =_I \hat{\pi}_p$ if $\pi_{pa} = \hat{\pi}_{pa}$ for all $a \in A(I)$. Finally, given completely mixed strategies $\pi_\ell \in \Pi_\ell$, $\pi_f \in \Pi_f$ and $I \in \mathcal{I}_p$, $u_{p,I}(\pi_\ell, \pi_f)$ denotes player p's expected utility given that I has been reached and strategies π_ℓ and π_f are played.

Next, we introduce a fundamental building block: the idea of follower's best response at an information set $I \in \mathcal{I}_f$. Intuitively, π_f is an I-best response to π_ℓ whenever playing as prescribed by π_f at the information set I is part of some follower's best response to π_ℓ in the game following I, given that I has been reached during play. Formally:

Definition 11.3. *Given an EFSG* Γ*, a completely mixed* $\pi_\ell \in \Pi_\ell$*, and* $I \in \mathcal{I}_f$*, we say that* $\pi_f \in \Pi_f$ *is an* I-best response *to* π_ℓ*, written* $\pi_f \in \mathsf{BR}_I(\pi_\ell)$*, if the following holds:*

$$\max_{\substack{\hat{\pi}_f \in \Pi_f: \\ \pi_f =_I \hat{\pi}_f}} u_{f,I}\left(\pi_\ell, \pi_f /_I \hat{\pi}_f\right) = \max_{\hat{\pi}_f \in \Pi_f} u_{f,I}\left(\pi_\ell, \pi_f /_I \hat{\pi}_f\right).$$

For $p \in N$ and $\pi_p \in \Pi_p$, let $\{\pi_{p,k}\}_{k \in \mathbb{N}}$ be a sequence of completely mixed player p's strategies with π_p as a limit point. We are now ready to define the refinement concept. In words, in a quasi-perfect SE, the leader selects an optimal strategy to commit to in *all* information sets, given that the follower best responds to it at *every* information set, following *some* tie-breaking rule. Specifically, the second point in Definition 11.4 ensures that the leader's commitment is optimal also in those information sets that are unreachable in absence of players' errors. Notice that the leader only accounts for follower's future errors, while the follower assumes that only the leader can make mistakes in future. This is in line with the idea underlying quasi-perfect equilibria in non-Stackelberg games Van Damme (1984). [1]

Definition 11.4. *Given an EFSG* Γ*,* (π_ℓ, π_f) *is a* quasi-perfect Stackelberg equilibrium (QP-SE) *of* Γ *if there exist sequences* $\{\pi_{p,k}\}_{k\in\mathbb{N}}$*, defined for every* $p \in N$ *and* $\pi_p \in \Pi_p$*, such that:*

1. $\pi_f \in \mathsf{BR}_I(\pi_{\ell,k})$ *for all* $I \in \mathcal{I}_f$*;*
2. *for all* $I \in \mathcal{I}_\ell \cup \{I_\varnothing\}$ *and* $\hat{\pi}_\ell \in \Pi_\ell$*, there exists* $\hat{\pi}_f \in \Pi_f : \hat{\pi}_f \in \mathsf{BR}_{\hat{I}}(\pi_{\ell,k}/_I\hat{\pi}_{\ell,k})$ *for all* $\hat{I} \in \mathcal{I}_f$*, with:*

$$u_\ell\left(\pi_{\ell,k}/_I\pi_\ell, \pi_{f,k}\right) \geq u_\ell\left(\pi_{\ell,k}/_I\hat{\pi}_\ell, \hat{\pi}_{f,k}\right). \tag{11.1}$$

As with SEs, we introduce the *strong* version of QP-SEs. [2]

Definition 11.5. *Given an EFSG* Γ*,* (π_ℓ, π_f) *is a* quasi-perfect strong Stackelberg equilibrium (QP-SSE) *of* Γ *if there exist* $\{\pi_{p,k}\}_{k\in\mathbb{N}}$*, defined for every* $p \in N$ *and* $\pi_p \in \Pi_p$*, such that:*

1. $\pi_f \in \mathsf{BR}_I(\pi_{\ell,k})$ *for all* $I \in \mathcal{I}_f$*;*
2. *for all* $I \in \mathcal{I}_\ell \cup \{I_\varnothing\}$*,* $\hat{\pi}_\ell \in \Pi_\ell$*, and* $\hat{\pi}_f \in \Pi_f : \hat{\pi}_f \in \mathsf{BR}_{\hat{I}}(\pi_{\ell,k}/_I\hat{\pi}_{\ell,k})$ *for all* $\hat{I} \in \mathcal{I}_f$*, Equation* (11.1) *holds.*

As we will show in Subsection 11.1.1, QP-SEs are refinements of SEs, *i.e.*, any QP-SE is also an SE.

11.1.1 QP-SEs and Perturbation Schemes

Let us recall that, in Definition 10.4, we introduced a family of ϵ-perturbation schemes for EFSGs in sequence form, claiming that quasi-perfect SEs can

[1] Van Damme (1984) defines a quasi-perfect equilibrium of an n-player extensive-form game as a strategy profile $(\pi_p)_{p\in N}$ obtained as a limit point of a sequence of completely mixed strategy profiles $\{(\pi_{p,k})_{p\in N}\}_{k\in\mathbb{N}}$ such that $\pi_p \in \mathsf{BR}_I((\pi_{q,k})_{q\neq p\in N})$ for all $p \in N$ and $I \in \mathcal{I}_p$.

[2] Since Equation (11.1) must hold for every $\hat{\pi}_\ell \in \Pi_\ell$ and $\hat{\pi}_f \in \Pi_f : \hat{\pi}_f \in \mathsf{BR}_{\hat{I}}(\pi_{\ell,k}/_I\hat{\pi}_{\ell,k})$ for all $\hat{I} \in \mathcal{I}_f$, Definition 11.5 assumes that the follower breaks ties in favor of the leader.

be defined in terms of such perturbations. The main result of this chapter is to show that this is indeed the case, *i.e.*, perturbed EFSGs obtained for such perturbation schemes satisfy the following fundamental property: *limits of SEs in perturbed sequence-form EFSGs are QP-SEs of the original unperturbed EFSGs as the magnitude of the perturbation ϵ goes to zero*. In addition to being theoretically relevant, this result enables us to design an algorithm for computing QP-SEs in EFSGs (Section 11.3).

In the following, for the ease of presentation, we denote by $(\Gamma, \xi_\ell, \xi_f)$ a *ξ-perturbed* EFSG obtained for some quasi-perfect ϵ-perturbation scheme based on the functions ξ_ℓ and ξ_f (see Definition 10.4). Moreover, we let $\Gamma(\epsilon)$ be a particular *ξ-perturbed game instance* in sequence form, obtained from Γ by restricting each set of realization plans R_p to be $R_p(\epsilon)$. We also denote by $r_p(\epsilon)$ any realization plan in $R_p(\epsilon)$, and we let $\xi_p(\epsilon) \in \mathbb{R}^{|\Sigma_p|}$ be a vector whose components are the lower-bounds $\xi_p(\epsilon, \sigma_p)$. We denote by $\tilde{r}_p(\epsilon) = r_p(\epsilon) - \xi_p(\epsilon)$ the *residual* of $r_p(\epsilon)$, which represents the part of player p's strategy that is not fixed by the perturbation. [3]

Next, we state our main result about sequences of SEs in ξ-perturbed games. We postpone the proof to Section 11.2.

Theorem 11.1. *Given a ξ-perturbed EFSG $(\Gamma, \xi_\ell, \xi_f)$, let $\{\epsilon_k\}_{k\in\mathbb{N}} \to 0$ and let $\{(r_\ell(\epsilon_k), r_f(\epsilon_k))\}_{k\in\mathbb{N}}$ be a sequence of SEs in $\Gamma(\epsilon_k)$. Then, any limit point (π_ℓ, π_f) of the sequence $\{(\pi_{\ell,k}, \pi_{f,k})\}_{k\in\mathbb{N}}$ is a QP-SE of Γ, where $(\pi_{\ell,k}, \pi_{f,k})$ are equivalent to $(r_\ell(\epsilon_k), r_f(\epsilon_k))$ for all $k \in \mathbb{N}$.*

Theorem 11.1 also allows us to conclude the following, as a consequence of Theorem 10.1.

Corollary 11.1.1. *Any QP-SE of an EFSG Γ is an SE of Γ.*

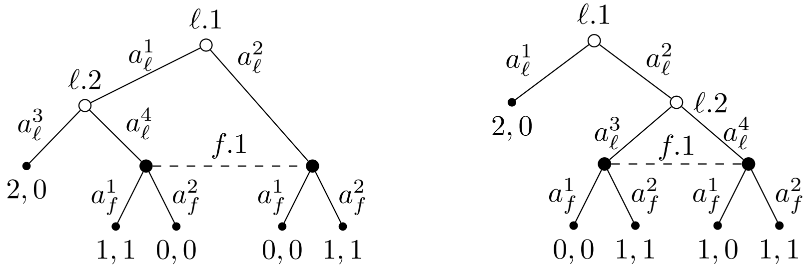
The second and third points in Definition 10.4 cannot be removed:

Proposition 11.1. *There are ξ-perturbed EFSGs $(\Gamma, \xi_\ell, \xi_f)$ obtained for ξ_p-perturbation schemes that violate the second or third point in Definition 10.4 for which Theorem 11.1 does not hold.*

Proof. Consider the EFSG in Figure 11.1b with $\xi_\ell(\epsilon, a_\ell^1) = \xi_\ell(\epsilon, a_\ell^2) = \epsilon$ and $\xi_\ell(\epsilon, a_\ell^2 a_\ell^3) = \xi_\ell(\epsilon, a_\ell^2 a_\ell^4) = \frac{\epsilon}{3}$, which violates the third requirement in Definition 10.4. Clearly, any SE of $\Gamma(\epsilon)$ requires $r_\ell(\epsilon, a_\ell^1) = 1 - \epsilon$, $r_\ell(\epsilon, a_\ell^2) = \epsilon$, $r_\ell(\epsilon, a_\ell^2 a_\ell^3) = \frac{\epsilon}{3}$, and $r_\ell(\epsilon, a_\ell^2 a_\ell^4) = \frac{2\epsilon}{3}$. Thus, any limit point of a sequence of SEs has $\pi_{\ell a_\ell^3} = \frac{1}{3}$ and $\pi_{\ell a_\ell^4} = \frac{2}{3}$, which cannot be the case in a QP-SE of Γ, as the leader's optimal strategy at $\ell.2$ is to play a_ℓ^4.

[3] We assume without loss of generality that $\Gamma(\epsilon)$ is well-defined, that is, each set $R_p(\epsilon)$ is non-empty for every $\epsilon \in (0, 1]$.

As for the second requirement, we can build a similar example by setting $\xi_\ell(\epsilon, a_\ell^2) = \frac{1}{3}$. $\square$



(a) *Example of EFSG used to show that our perturbation schemes generalize those introduced by Miltersen and Sørensen (2010).*

(b) *Example of EFSG used to show that the second and third points in Definition 10.4 are necessary for Theorem 11.1 to hold.*

Figure 11.1: *Examples of EFSGs.*

Miltersen and Sørensen (2010) introduced the idea of perturbing games using the sequence form in order to find a quasi-perfect equilibrium. Our perturbation scheme generalizes theirs, where $\xi_p(\epsilon, \sigma_p) = \epsilon^{|\sigma_p|}$ for all $\sigma_p \in \Sigma_p \setminus \{\sigma_\varnothing\}$, with $|\sigma_p|$ being the number of actions in σ_p. There are games where our perturbation captures QP-SEs that are *not* obtainable with theirs. For instance, in the EFSG in Figure 11.1a, (π_ℓ, π_f), with $\pi_{\ell a_\ell^1} = \pi_{\ell a_\ell^3} = 1$, $\pi_{\ell a_\ell^2} = \pi_{\ell a_\ell^4} = 0$, and $\pi_{f a_f^1} = \pi_{f a_f^2} = \frac{1}{2}$, is a QP-SE that cannot be obtained with their perturbation scheme while it is reachable by setting $\xi_\ell(\epsilon, a_\ell^2) = \epsilon^2$. We observe that (π_ℓ, π_f) is also a quasi-perfect equilibrium when we look at the game as its non-Stackelberg counterpart; this shows that our perturbation scheme generalizes theirs also for quasi-perfect equilibria.

11.2 Limits of SEs in ξ-Perturbed Games are QP-SEs

Before proving our main result, we now study the properties of the follower's best responses to the leader's strategy in ξ-perturbed games. These properties will be useful for proving our results later in the section.

In the following, letting $\Sigma_p(a) = \{\sigma_p \in \Sigma_p \mid a \in \sigma_p\}$ for all $a \in A_p$, $\Sigma_p(I) = \bigcup_{a \in A(I)} \Sigma_p(a)$ denotes player p's sequences that pass through information set $I \in \mathcal{I}_p$. For the ease of presentation, given $I \in \mathcal{I}_p$, $g_{p,I}(r_\ell, r_f) = \sum_{\sigma \in \Sigma : \sigma_p \in \Sigma_p(I)} u_p(\sigma) r_\ell(\sigma_\ell) r_f(\sigma_f)$ denotes player p's expected utility contribution from terminal nodes reachable from I. Finally, for

$I \in \mathcal{I}_p$, let $R_p(I) \subseteq R_p$ be the set of $r_p \in R_p : r_p(\sigma_p(I)) = 1$, while, for $a \in A(I)$, $R_p(a) \subseteq R_p(I)$ is the set of $r_p \in R_p : r_p(\sigma_p(I)a) = 1$.

Let $\mathsf{BR}_{\Gamma(\epsilon)}(r_\ell(\epsilon)) = \arg\max_{r_f(\epsilon) \in R_f(\epsilon)} u_f(r_\ell(\epsilon), r_f(\epsilon))$ be the set of follower's best responses to $r_\ell(\epsilon) \in R_\ell(\epsilon)$ in $\Gamma(\epsilon)$. The next lemma gives a mathematical programming formulation of the follower's best-response problem in $\Gamma(\epsilon)$.

Lemma 11.1. *For every $r_\ell(\epsilon) \in R_\ell(\epsilon)$, $r_f(\epsilon) \in \mathsf{BR}_{\Gamma(\epsilon)}(r_\ell(\epsilon))$ if and only if $\tilde{r}_f(\epsilon)$ is optimal for Problem $\mathcal{P}(\epsilon)$ below.*

$$\mathcal{P}(\epsilon) \ : \ \begin{cases} \max\limits_{\tilde{r}_f} & r_\ell(\epsilon)^T U_f \tilde{r}_f \\ \text{s.t.} & F_f \tilde{r}_f = f_f - F_f \xi_f(\epsilon), \quad \tilde{r}_f \geq 0. \end{cases}$$

Proof. Since, $r_f(\epsilon) \in \mathsf{BR}_{\Gamma(\epsilon)}(r_\ell(\epsilon))$ if and only if

$$r_f(\epsilon) \in \operatorname*{arg\,max}_{r_f : F_f r_f = f_f, r_f \geq \xi_f(\epsilon)} r_\ell(\epsilon)^T U_f r_f,$$

introducing variables $\tilde{r}_f = r_f - \xi_f(\epsilon)$ and dropping the constant term $r_\ell(\epsilon)^T U_i \xi_f(\epsilon)$ from the objective, we obtain that $r_f(\epsilon)$ must be an optimal solution to Problem $\mathcal{P}(\epsilon)$. $\square$

The dual of Problem $\mathcal{P}(\epsilon)$ above is as follows.

Proposition 11.2. *For $r_\ell(\epsilon) \in R_\ell(\epsilon)$, Problem $\mathcal{D}(\epsilon)$ below is the dual of Problem $\mathcal{P}(\epsilon)$, where $v_f \in \mathbb{R}^{|\mathcal{I}_f|+1}$ is the vector of dual variables.*

$$\mathcal{D}(\epsilon) \ : \ \begin{cases} \min\limits_{v_f} & (f_f - F_f \xi_f(\epsilon))^T v_f & \text{(11.2a)} \\ \text{s.t.} & F_f^T v_f \geq U_f^T r_\ell(\epsilon). & \text{(11.2b)} \end{cases}$$

Remark 11.1. *Constraints* (11.2b) *in Problem $\mathcal{D}(\epsilon)$ defined above ensure that, for every $I \in \mathcal{I}_f$ and $a \in A(I)$, we have*

$$v_{f,I} \geq \sum_{\sigma \in \Sigma : \sigma_f = \sigma_f(I)a} u_f(\sigma) r_\ell(\epsilon, \sigma_\ell) + \sum_{\hat{I} \in \mathcal{I}_f : \sigma_f(\hat{I}) = \sigma_f(I)a} v_{f,\hat{I}}. \qquad (11.3)$$

The optimal solutions to Problem $\mathcal{D}(\epsilon)$ enjoy important properties that are stated in the following lemmas. The first one says that, in an optimal solution, each variable $v_{f,I}$ must equal the maximum possible expected utility the follower can achieve following information set $I \in \mathcal{I}_f$. The second lemma says that if an optimal solution to Problem $\mathcal{D}(\epsilon)$ satisfies Constraint (11.3) with equality for an information set $I \in \mathcal{I}_f$ and an action $a \in A(I)$, then playing a at I is optimal in the game following I.

Lemma 11.2. *For every $r_\ell(\epsilon) \in R_\ell(\epsilon)$, if $v_f^* \in \mathbb{R}^{|\mathcal{I}_f|+1}$ is optimal for Problem $\mathcal{D}(\epsilon)$, then for every $I \in \mathcal{I}_f$:*

$$v_{f,I}^* = \max_{\hat{r}_f \in R_f(I)} g_{f,I}(r_\ell(\epsilon), \hat{r}_f). \tag{11.4}$$

Proof. Let us consider Problem $\mathcal{D}(\epsilon)$. First, observe that, for every information set $I \in \mathcal{I}_f$, the objective function coefficient for the variable $v_{f,I}$ is equal to $\xi_f(\epsilon, \sigma_f(I)) - \sum_{a \in A(I)} \xi_f(\epsilon, \sigma_f(I)a)$. Assuming $\Gamma(\epsilon)$ is well-defined, such coefficients are positive for every $v_{f,I}$. Then, in an optimal solution $v_f^* \in \mathbb{R}^{|\mathcal{I}_f|+1}$ to Problem $\mathcal{D}(\epsilon)$, each variable $v_{f,I}$ is set to its minimum given Constraints (11.3). We prove Equation (11.4) using a simple inductive argument. The base case of the induction is when there is no information set $\hat{I} \neq I \in \mathcal{I}_f$ with $I \succeq \hat{I}$. For every action $a \in A(I)$, $v_{f,I} \geq \sum_{\sigma \in \Sigma : \sigma_f = \sigma_f(I)a} u_f(\sigma) r_\ell(\epsilon, \sigma_\ell)$, which, using the fact that $v_{f,I}^*$ must be set to its minimum possible value given the constraints, implies the following:

$$\begin{aligned} v_{f,I}^* &= \max_{a \in A(I)} \sum_{\sigma \in \Sigma : \sigma_f = \sigma_f(I)a} u_f(\sigma) r_\ell(\epsilon, \sigma_\ell) = \\ &= \max_{\hat{r}_f \in R_f(I)} g_{f,I}(r_\ell(\epsilon), \hat{r}_f), \end{aligned}$$

where the last equality holds since $\sum_{a \in A(I)} \hat{r}_f(\sigma_f(I)a) = \hat{r}_f(\sigma_f(I)) = 1$, for the definition of realization plan. As for the inductive step, let us consider an information set $I \in \mathcal{I}_f$ and assume, by induction, that Equation (11.4) holds for every information set $\hat{I} \neq I \in \mathcal{I}_f$ with $I \succeq \hat{I}$. We can write:

$$\begin{aligned} v_{f,I}^* &= \max_{a \in A(I)} \sum_{\sigma \in \Sigma : \sigma_f = \sigma_f(I)a} u_f(\sigma) r_\ell(\epsilon, \sigma_\ell) + \sum_{\hat{I} \in \mathcal{I}_f : \sigma_f(\hat{I}) = \sigma_f(I)a} v_{f,\hat{I}}^* = \\ &= \max_{a \in A(I)} \sum_{\sigma \in \Sigma : \sigma_f = \sigma_f(I)a} u_f(\sigma) r_\ell(\epsilon, \sigma_\ell) \quad + \\ &\qquad + \sum_{\hat{I} \in \mathcal{I}_f : \sigma_f(\hat{I}) = \sigma_f(I)a} \max_{\hat{r}_f \in R_f(\hat{I})} g_{f,\hat{I}}(r_\ell(\epsilon), \hat{r}_f) = \\ &= \max_{\hat{r}_f \in R_f(I)} g_{f,I}(r_\ell(\epsilon), \hat{r}_f), \end{aligned}$$

where the first equality directly follows from the optimality of v_f^*, the second one from the inductive hypothesis, while the last equality holds since we have $\sum_{a \in A(I)} \hat{r}_f(\sigma_f(I)a) = \hat{r}_f(\sigma_f(I)) = 1$. $\square$

Lemma 11.3. *For every $r_\ell(\epsilon) \in R_\ell(\epsilon)$, $I \in \mathcal{I}_f$, and $a \in A(I)$, if Constraint* (11.3) *holds with equality in an optimal solution to Problem $\mathcal{D}(\epsilon)$, then*

$$\max_{\hat{r}_f \in R_f(a)} g_{f,I}(r_\ell(\epsilon), \hat{r}_f) = \max_{\hat{r}_f \in R_f(I)} g_{f,I}(r_\ell(\epsilon), \hat{r}_f). \tag{11.5}$$

Proof. Let $v^* \in \mathbb{R}^{|\mathcal{I}_i|+1}$ be an optimal solution to Problem $\mathcal{D}(\epsilon)$ that satisfies Constraint (11.3), for $I \in \mathcal{I}_f$ and $a \in A(I)$, with equality. We can write:

$$\begin{aligned} v^*_{f,I} &= \sum_{\sigma \in \Sigma : \sigma_f = \sigma_f(I)a} u_f(\sigma) r_\ell(\epsilon, \sigma_\ell) + \sum_{\hat{I}_f \in \mathcal{I}_f : \sigma_f(\hat{I}) = \sigma_f(I)a} v^*_{f,\hat{I}} = \\ &= \max_{\hat{r}_f \in R_f(a)} g_{f,I}(r_\ell(\epsilon), \hat{r}_f) = \max_{\hat{r}_f \in R_f(I)} g_{f,I}(r_\ell(\epsilon), \hat{r}_f), \end{aligned}$$

where the second equality holds for the optimality of v^*_f and the last one for Lemma 11.2. □

Now, we are ready to prove a fundamental property of the follower's best responses in ξ-perturbed game instances $\Gamma(\epsilon)$. Intuitively, in a perturbed game instance, the follower best responds playing sequence $\sigma(I_f)a$ with probability strictly greater than its lower-bound $\xi_f(\epsilon, \sigma_f(I)a)$ only if playing a is optimal in the game following I. Theorem 11.2 formally expresses the idea that, in a perturbed game instance $\Gamma(\epsilon)$, when the follower decides how to best respond to a leader's commitment in a given information set, she does not take into account her future trembles, but only opponents' ones.

Theorem 11.2. *Given $r_\ell(\epsilon) \in R_\ell(\epsilon)$, $r_f(\epsilon) \in \mathsf{BR}_{\Gamma(\epsilon)}(r_\ell(\epsilon))$, $I \in \mathcal{I}_f$, and $a \in A(I)$, if $r_f(\epsilon, \sigma_f(I)a) > \xi_f(\epsilon, \sigma_f(I)a)$, then*

$$\max_{\hat{r}_f \in R_f(a)} g_{f,I}(r_\ell(\epsilon), \hat{r}_f) = \max_{\hat{r}_f \in R_f(I)} g_{f,I}(r_\ell(\epsilon), \hat{r}_f).$$

Proof. By Lemma 11.1, $r_f(\epsilon) \in \mathsf{BR}_{\Gamma(\epsilon)}(r_\ell(\epsilon))$ if and only if $\tilde{r}_f(\epsilon) = r_f(\epsilon) - \xi_f(\epsilon)$ is optimal for Problem $\mathcal{P}(\epsilon)$. By applying the complementarity slackness theorem to Problems $\mathcal{P}(\epsilon)$ and $\mathcal{D}(\epsilon)$ we have that, if $\tilde{r}_f(\epsilon)$ and $v^*_f \in \mathbb{R}^{|\mathcal{I}_f|+1}$ are optimal, then, whenever $\tilde{r}_f(\epsilon, \sigma_f(I)a) > 0$, *i.e.*, $r_f(\epsilon, \sigma_f(I)a) > \xi_f(\epsilon, \sigma_f(I)a)$, Constraint (11.3) for information set I and action a must hold with equality, which, by Lemma 11.3, yields Equation (11.5). □

Now, we are ready to prove Theorem 11.1.

First, we introduce two lemmas. The first provides a characterization of I-best responses in terms of sequence form. Intuitively, a follower's strategy π_f is an I-best response to π_ℓ if and only if it places positive probability only on actions $a \in A(I)$ that are part of some best response of the follower below information set I.

Lemma 11.4. *Given an SEFG Γ, a completely mixed $\pi_\ell \in \Pi_\ell$ and $I \in \mathcal{I}_f$, $\pi_f \in \mathsf{BR}_I(\pi_\ell)$ if for every $a \in A(I)$:*

$$\pi_{fa} > 0 \implies \max_{\hat{r}_f \in R_f(a)} g_{f,I}(r_\ell, \hat{r}_f) = \max_{\hat{r}_f \in R_f(I)} g_{f,I}(r_\ell, \hat{r}_f),$$

where $r_\ell \in R_\ell$ is equivalent to π_ℓ.

Proof. First, let us notice that, for every $I \in \mathcal{I}_f$ and $a \in A(I)$, the following relation holds:

$$\begin{aligned} &\max_{\hat{r}_f \in R_f(a)} g_{f,I}(r_\ell, \hat{r}_f) = \max_{\hat{r}_f \in R_f(I)} g_{f,I}(r_\ell, \hat{r}_f) \implies \\ &\max_{\hat{\pi}_f \in \Pi_f : \hat{\pi}_{fa} = 1} u_{f,I}\left(\pi_\ell, \pi_f \big/_I \hat{\pi}_f\right) = \max_{\hat{\pi}_f \in \Pi_f} u_{f,I}\left(\pi_\ell, \pi_f \big/_I \hat{\pi}_f\right) \end{aligned} \tag{11.6}$$

In order to see this, for $I \in \mathcal{I}_f$, let $Z(I) \subseteq Z$ be the set of terminal nodes that are potentially reachable from I, and, for $h \in Z(I)$ and $\hat{\pi}_f \in \Pi_f$, let

$$\mathcal{U}_{f,h}(\pi_\ell, \hat{\pi}_f) = u_f(h) \prod_{a \in \sigma_\ell(h)} \pi_{\ell a} \prod_{a \in \sigma_f(h) \setminus \sigma_f(I)} \hat{\pi}_{fa}.$$

Given the realization equivalence of r_ℓ and π_ℓ, and the fact that $\hat{r}_f(\sigma_f(I)) = 1$, the left-hand side in the first line of Equation (11.6) is equivalent to $\max_{\hat{\pi}_f \in \Pi_f : \hat{\pi}_{fa} = 1} \sum_{h \in Z(I)} \mathcal{U}_{f,h}(\pi_\ell, \hat{\pi}_f)$, while the right-hand side is the same as $\max_{\hat{\pi}_f \in \Pi_f} \sum_{h \in Z(I)} \mathcal{U}_{f,h}(\pi_\ell, \hat{\pi}_f)$. Then, by dividing both sides of the equality in the first line of Equation (11.6) by $\sum_{h \in Z(I)} \prod_{a \in \sigma_f(h)} \pi_{fa}$, by definition of $u_{f,I}(\pi_\ell, \pi_f \big/_I \hat{\pi}_f)$ we get the second line. Now, say that the condition of the lemma holds for every $a \in A(I)$. Clearly, we have

$$\max_{\hat{\pi}_f \in \Pi_f : \pi_f =_I \hat{\pi}_f} u_{f,I}(\pi_\ell, \pi_f \big/_I \hat{\pi}_f) = \sum_{a \in A(I)} \pi_{fa} \max_{\hat{\pi}_f \in \Pi_f : \hat{\pi}_{fa} = 1} u_{f,I}(\pi_\ell, \pi_f \big/_I \hat{\pi}_f),$$

and, since $\pi_{fa} > 0$ only if it holds that

$$\max_{\hat{r}_f \in R_f(a)} g_{f,I}(r_\ell, \hat{r}_f) = \max_{\hat{r}_f \in R_f(I)} g_{f,I}(r_\ell, \hat{r}_f),$$

Equation (11.6) proves the result. $\square$

The next lemma shows that any limit point of a sequence of follower's best responses in ξ-perturbed games is a follower's best response at every information set in Γ.

Lemma 11.5. *Given a ξ-perturbed SEFG $(\Gamma, \xi_\ell, \xi_f)$, let $\{\epsilon_k\}_{k\in\mathbb{N}} \to 0$ and let $\{(r_\ell(\epsilon_k), r_f(\epsilon_k))\}_{k\in\mathbb{N}}$ be a sequence of realization plans in $\Gamma(\epsilon_k)$ with $r_f(\epsilon_k) \in \mathsf{BR}_{\Gamma(\epsilon_k)}(r_\ell(\epsilon_k))$. Then, any limit point (π_ℓ, π_f) of $\{(\pi_{\ell,k}, \pi_{f,k})\}_{k\in\mathbb{N}}$ is such that, eventually, $\pi_f \in \mathsf{BR}_{I_f}(\pi_{\ell,k})$ for all $I \in \mathcal{I}_f$, where $(\pi_{\ell,k}, \pi_{f,k})$ are equivalent to $(r_\ell(\epsilon_k), r_f(\epsilon_k))$ for all $k \in \mathbb{N}$.*

Proof. First, notice that there must exist $\bar{k} \in \mathbb{N}$ such that, for all $k \in \mathbb{N} : k \geq \bar{k}$, and for every follower's information set $I \in \mathcal{I}_f$ and action $a \in A(I)$, if $\pi_{fa} > 0$, then $r_f(\epsilon_k, \sigma_f(I)a) > \xi_f(\epsilon_k, \sigma_f(I)a)$. Otherwise, by the conditions in Definition 10.4, it would be $\pi_{fa} = 0$. Let us fix $I \in \mathcal{I}_f$ and $a \in A(I)$. Suppose that $\pi_{fa} > 0$. For all $k \in \mathbb{N} : k \geq \bar{k}$, we have that $r_f(\epsilon_k, \sigma_f(I)a) > \xi_f(\epsilon_k, \sigma_f(I)a)$, which, by Theorem 11.2, implies the following:

$$\max_{\hat{r}_f \in R_f(a)} g_{f,I}(r_\ell(\epsilon_k), \hat{r}_f) = \max_{\hat{r}_f \in R_f(I)} g_{f,I}(r_\ell(\epsilon_k), \hat{r}_f).$$

Thus, Lemma 11.4 allows us to conclude that $\pi_f \in \mathsf{BR}_I(\pi_{\ell,k})$ for all $k \in \mathbb{N} : k \geq \bar{k}$, which proves the result. $\square$

Finally, we can prove Theorem 11.1.

Proof of Theorem 11.1. First, since $r_f(\epsilon_k) \in \mathsf{BR}_{\Gamma(\epsilon_k)}(r_\ell(\epsilon_k))$ for all $k \in \mathbb{N}$, Lemma 11.5 allows us to conclude that the first point in Definition 11.4 holds. Therefore, in order to prove Theorem 11.1, we need to show that the second point holds as well. For contradiction, suppose that it does not hold, *i.e.*, no matter how we choose sequences $\{\pi_{p,k}\}_{k\in\mathbb{N}}$, for $p \in N$ and $\pi_p \in \Pi_p$, there is an information set $I \in \mathcal{I}_\ell \cup \{I_\varnothing\}$ and a leader's strategy $\hat{\pi}_\ell \in \Pi_\ell$ such that, for every $\hat{\pi}_f \in \Pi_f : \hat{\pi}_f \in \mathsf{BR}_{\hat{I}}(\pi_{\ell,k}\big/_I\hat{\pi}_{\ell,k})$ for all $\hat{I} \in \mathcal{I}_f$, we have:

$$u_\ell(\pi_{\ell,k}\big/_I\pi_\ell, \pi_{f,k}) < u_\ell(\pi_{\ell,k}\big/_I\hat{\pi}_\ell, \hat{\pi}_{f,k}).$$

By continuity, there must exist an index $\bar{k} \in \mathbb{N}$ such that, for all $k \in \mathbb{N} : k \geq \bar{k}$, the following holds:

$$u_\ell(\pi_{\ell,k}\big/_I\pi_{\ell,k}, \pi_{f,k}) < u_\ell(\pi_{\ell,k}\big/_I\hat{\pi}_{\ell,k}, \hat{\pi}_{f,k}).$$

Moreover, $u_\ell(\pi_{\ell,k}\big/_I\pi_{\ell,k}, \pi_{f,k}) = u_\ell(\pi_{\ell,k}, \pi_{f,k})$. Let sequence $\{\hat{\pi}_{\ell,k}\}_{k\in\mathbb{N}}$ be such that $\hat{r}_\ell(\epsilon_k) \in R_\ell(\epsilon_k)$ for all $k \in \mathbb{N}$, where each realization plan $\hat{r}_\ell(\epsilon_k)$ is equivalent to the strategy $\pi_{\ell,k}\big/_I\hat{\pi}_{\ell,k}$. This is always possible since the third point in Definition 10.4 is satisfied. Now, let us consider a sequence $\{(\hat{r}_\ell(\epsilon_k), \hat{r}_f(\epsilon_k)\}_{k\in\mathbb{N}}$ with $\hat{r}_f(\epsilon_k) \in \mathsf{BR}_{\Gamma(\epsilon_k)}(\hat{r}_\ell(\epsilon_k))$, and let us define $\{(\pi_{\ell,k}\big/_I\hat{\pi}_{\ell,k}, \hat{\pi}_{f,k})\}_{k\in\mathbb{N}}$ as a sequence such that each strategy $\hat{\pi}_{f,k}$ is equivalent to $\hat{r}_f(\epsilon_k)$. By Lemma 11.5, any limit point $(\pi_\ell\big/_I\hat{\pi}_\ell, \hat{\pi}_f)$ of $\{(\pi_{\ell,k}\big/_I\hat{\pi}_{\ell,k}, \hat{\pi}_{f,k})\}_{k\in\mathbb{N}}$ satisfies $\hat{\pi}_f \in \mathsf{BR}_{\hat{I}}(\pi_{\ell,k}\big/_I\hat{\pi}_{\ell,k})$ for all $\hat{I} \in \mathcal{I}_f$. Thus, using the equivalence between strategies and realization plans, for all $k \in \mathbb{N} : k \geq \bar{k}$ we have the following:

$$u_\ell(r_\ell(\epsilon_k), r_f(\epsilon_k)) < u_\ell(\hat{r}_\ell(\epsilon_k), \hat{r}_f(\epsilon_k)).$$

Notice that this holds no matter how we choose $\hat{r}_f(\epsilon_k) \in \mathsf{BR}_{\Gamma(\epsilon_k)}(\hat{r}_\ell(\epsilon_k))$, which contradicts the fact that $(r_\ell(\epsilon_k), r_f(\epsilon_k))$ is an SE of $\Gamma(\epsilon_k)$. □

11.3 Computing a Quasi-Perfect Stackelberg Equilibrium

One can use our perturbation scheme to compute an (approximate) QP-SE. We do this by developing an LP for computing a *Stackelberg extensive-form correlated equilibrium* (SEFCE) in a given ξ-perturbed game instance, where we maximize the leader's value. We then conduct a *branch-and-bound* search on this SEFCE LP. It branches on which actions to *force* be recommended to the follower (by the correlation device of the SEFCE). The idea is that, as long as we only recommend a single action to the follower at any given information set, we get an SE of the perturbed game (specifically an SSE), and, thus, according to Theorem 11.1, a QP-SE (specifically QP-SSE) if we take the limit point of the perturbations. As in prior papers on *extensive-form correlated equilibrium* (EFCE) computation in general-sum games, we focus on games without chance nodes (Von Stengel and Forges, 2008; Cermak et al., 2016).

For computing an SEFCE we need to specify joint probabilities over sequence pairs $(\sigma_\ell, \sigma_f) \in \Sigma := \Sigma_\ell \times \Sigma_f$. However, not all pairs need to specify probabilities, only pairs such that choosing σ_f is affected by the probability put on σ_ℓ (we do not need to care about the converse of this, as only the follower needs to be induced to follow the recommended strategy). Intuitively, the set of the leader's sequences relevant to a given $\sigma_f \in \Sigma_f$ is made of those sequences that affect the expected value of σ_f or any alternative sequence $\hat{\sigma}_f \in \Sigma_f$ whose last action is available at $I_f(\sigma_f)$.

Definition 11.6 (Relevant Sequences)**.** *A pair $(\sigma_\ell, \sigma_f) \in \Sigma$ is* relevant *if either $\sigma_\ell = \sigma_\varnothing$ or there exists $h, \hat{h} \in H$ s.t. $\hat{h}$ precedes h, $h \in I_f(\sigma_f)$, and $\hat{h} \in I_\ell(\sigma_\ell)$, or if the condition holds with the roles of σ_ℓ and σ_f reversed.*

For every information set $I \in \mathcal{I}_p$, we let $rel(I)$ be the set of sequences relevant to each child sequence $\sigma_p(I)a$ for $a \in A(I)$. We let $p(\sigma_\ell, \sigma_f)$ be the probability that we recommend that the leader plays sequence σ_ℓ, and that the follower sends her *residual* (*i.e.*, the probability that is not fixed by the perturbation) to σ_f. Moreover, we let $\eta(\sigma_f)$ be the maximum probability that the follower can put on a sequence $\sigma_f \in \Sigma_f$ given the ξ_f-perturbation scheme.

First, we introduce a new value function representing the value to the leader of the sequence pair $(\sigma_\ell, \sigma_f) \in \Sigma$ given that σ_f represents an assignment of residual probability:

$$u_\ell^\epsilon(\sigma_\ell, \sigma_f) = \sum_{h \in Z : \sigma_\ell(h) = \sigma_\ell \wedge \sigma_f(h) = \sigma_f} \eta(\sigma_f) u_\ell(h) + \sum_{\hat{\sigma}_f \in \Sigma_f} \xi_f(\epsilon, \hat{\sigma}_f) u_\ell(\sigma_\ell, \hat{\sigma}_f).$$

The following LP finds an SEFCE in a ξ-perturbed SEFG.

$$\max_{p,v} \quad \sum_{(\sigma_\ell, \sigma_f) \in \Sigma} p(\sigma_\ell, \sigma_f) u_\ell^\epsilon(\sigma_\ell, \sigma_f) \tag{11.7a}$$

$$\text{s.t.} \quad p(\sigma_\emptyset, \sigma_\emptyset) = 1 \tag{11.7b}$$

$$p(\sigma_\ell, \sigma_f) \geq 0 \qquad \forall (\sigma_\ell, \sigma_f) \in \Sigma \tag{11.7c}$$

$$\sum_{\sigma_f \in rel(\sigma_\ell)} p(\sigma_\ell, \sigma_f) \geq \xi_\ell(\epsilon, \sigma_\ell) \qquad \forall \sigma_\ell \in \Sigma_\ell \tag{11.7d}$$

$$p(\sigma_\ell(I), \sigma_f) = \sum_{a \in A(I)} p(\sigma_\ell(I)a, \sigma_f) \qquad \forall I \in \mathcal{I}_\ell, \sigma_f \in rel(I) \tag{11.7e}$$

$$p(\sigma_\ell, \sigma_f(I)) = \sum_{a \in A(I)} p(\sigma_\ell, \sigma_f(I)a) \qquad \forall I \in \mathcal{I}_f, \sigma_\ell \in rel(I) \tag{11.7f}$$

$$\begin{aligned} v(\sigma_f) = \eta(\sigma_f) & \sum_{\sigma_\ell \in rel(\sigma_f)} p(\sigma_\ell, \sigma_f) u_f(\sigma_\ell, \sigma_f) + \\ & + \sum_{I \in \mathcal{I}_f : \sigma_f(I) = \sigma_f} \sum_{a \in A(I)} v(\sigma_f a) \qquad \forall \sigma_f \in \Sigma_f \end{aligned} \tag{11.7g}$$

$$\begin{aligned} v(I, \sigma_f) \geq \eta(\sigma_f(I)a) & \sum_{\sigma_\ell \in rel(\sigma_f)} p(\sigma_\ell, \sigma_f) u_f(\sigma_\ell, \sigma_f(I)a) + \\ & + \sum_{\hat{I} \in \mathcal{I}_f ; \sigma_f(\hat{I}) = \sigma_f(I)a} v(\hat{I}, \sigma_f) \\ & \qquad \forall I \in \mathcal{I}_f, a \in A(I), \sigma_f \in prec(I) \end{aligned} \tag{11.7h}$$

$$v(\sigma_f(I)a) = v(I, \sigma_f(I)a) \qquad \forall I \in \mathcal{I}_f, a \in A(I). \tag{11.7i}$$

In (11.7h) of this LP, $prec(I)$, where $I \in \mathcal{I}_f$, is the set of follower's sequences σ_f that precede I in the sense that there is $\hat{I} \in \mathcal{I}_f$ with $\sigma_f(\hat{I}) \sqsubseteq \sigma_f(I)$ and $\sigma_f = \sigma_f(\hat{I})a$ for some $a \in A(\hat{I})$. This LP is a modification of the SEFCE LP given by Cermak et al. (2016). The new LP has two modifications to allow perturbation. First, it has constraints (11.7d) to ensure that the sum of recommendation probabilities on any leader's sequence is at least $\xi_\ell(\epsilon, \sigma_\ell)$. Second, because we are now recommending where to send residual probability for the follower, we must modify the objective in order to give the correct expected value for the leader. [4]

We can branch-and-bound on recommendations to the follower in a way that ensures that the final outcome is an SSE. That is guaranteed by the following theorem, which shows that we can add and remove constraints on which follower actions to recommend in a way that guarantees an SSE of the perturbed game as long as the follower is recommended a "pure" strategy with respect to the residual probabilities.

Theorem 11.3. *If a solution to LP* (11.7) *is such that for all $I \in \mathcal{I}_f$ there exists $a \in A(I)$ such that $p(\sigma_\ell, \sigma_f(I)\hat{a}) = 0$ for all $\hat{a} \in A(I), \sigma_\ell \in rel(\sigma_f(I)a)$ with $\hat{a} \neq a$, then a strategy profile can be extracted in polynomial time such that it is an SSE of the perturbed game instance.*

Proof. First, we check that the leader strategy is valid. The argument is identical to that of Cermak et al. (2016). For the leader strategy at a given information set I we pick an arbitrary $\sigma_f \in rel(\sigma_\ell(I))$ that is played with positive probability and use the value $p(\sigma_\ell(I)a, \sigma_f)$ for all $a \in I$. All $\sigma_f \in rel(\sigma_\ell(I))$ recommend identical probability on $\sigma_\ell(I)a$ due to (11.7e) and the fact that we allow only a single follower action to be recommended at every follower information set. The incentive constraints (11.7g) - (11.7i) are identical to the original constraints given by Von Stengel and Forges (2008), so we only need to argue that we correctly represent the value of sending the residual along each sequence. But the value of sending the residual on σ_f is simply the original value $\sum_{\sigma_\ell \in rel(\sigma_f)} p(\sigma_\ell, \sigma_f) u_f(\sigma_\ell, \sigma_f)$, except that we can send at most $\eta(\sigma_f)$ probability on σ_f, plus the value of whichever choice we make for sending residual along descendants of σ_f. This is exactly the value that we encode in our constraints. It is easy to see that any SSE is a feasible solution to the LP: since the follower plays a pure strategy we can assign them their pure strategy, and assign the leader SSE strategy the same way across all follower recommendations. □

[4] We use the definition of relevant sequences and the LP from Von Stengel and Forges (2008) rather than those of Cermak et al. (2016). The latter are not well defined for (11.7e) and (11.7f).

Now it is obvious that the LP (11.7) upper bounds the value of any SSE since the SSE is a feasible solution to the LP.

Theorem 11.3 shows that one way to find an SSE is to find a solution to LP (11.7) where the follower is recommended a pure strategy with respect to the residual probabilities. Since any SSE represents such a solution, we can branch on which actions we make pure at each information set, and use branch-and-bound to prune the space of possible solutions. This approach was proposed by Cermak et al. (2016) for computing SSEs in unperturbed games, where they showed that it performs better than a single MIP. Because our LP for perturbed games uses residual probabilities for the follower, we can apply the branching methodology of Cermak et al. (2016). At each node in the search we choose some information set I where more than one action is recommended. We then branch on which action in $A(I)$ to recommend. Forcing a given action is accomplished by requiring all other action probabilities be zero. Our branch-and-bound chooses information sets according to depth, always branching on the shallowest one with at least two recommended action. We explore actions in descending order of mass, where the mass on $a \in A(I)$ (with sequence σ_f) is $\sum_{\sigma_\ell \in rel(\sigma_f)} p(\sigma_\ell, \sigma_f)$.

The algorithm finds an SSE of the perturbed game. In the limit as the perturbation approaches zero, this yields a QP-SE. No algorithm is currently known for computing such an exact limit. In practice, we pick a small perturbation and solve the branch-and-bound using that value. This immediately leads to an approximate notion of QP-SE (akin to approximate refinement notions in non-Stackelberg extensive-form games (Farina et al., 2017; Kroer et al., 2017)). Another approach is to use our algorithm as an anytime algorithm where one runs it repeatedly with smaller and smaller perturbation values.

11.3.1 Experimental Evaluation

We conducted experiments with our algorithm on two common benchmark extensive-form games. The first is a search game played on the graph shown in Figure 11.2. It is a simultaneous-move game (which can be modeled as a turn-taking EFG with appropriately chosen information sets). The leader controls two patrols that can each move within their respective shaded areas (labeled P1 and P2), and at each time step the controller chooses a move for both patrols. The follower is always at a single node on the graph, initially the leftmost node labeled S and can move freely to any adjacent node (except at patrolled nodes, the follower cannot move from a patrolled node to another patrolled node). The follower can also choose to wait in place for a time step in order to clean up their traces. If a patrol visits a node that was previously visited by the follower, and the follower did not wait to clean up their traces, they can see that the follower was there. If the follower reaches any of the rightmost nodes they received the respective payoff at the node (5 and 10, respectively). If the follower and any patrol are on the same node at any time step, the follower is captured, which leads to a payoff of 0 for the follower and a payoff of 1 for the leader. Finally, the game times out after k simultaneous moves, in which case the leader receives payoff 0 and the follower receives $-\infty$ (because we are interested in games where the follower attempts to reach an end node). This is the game considered by Kroer et al. (2018) except with the bottom layer removed, and is similar to games considered by Bošanskỳ et al. (2014) and Bošanský and Cermak (2015).

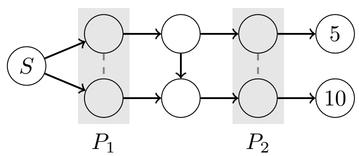


Figure 11.2: *The graph on which the search game is played.*

The second game is a variant on Goofspiel (Ross, 1971), a bidding game where each player has a hand of cards numbered 1 to 3. There are 3 prizes worth $1, \ldots, 3$. In each turn, the prize is the smallest among the remaining prizes. Within the turn, the each of two players simultaneously chooses some private card to play. The player with the larger card wins the prize. In case of a tie, the prize is discarded, so this is not a constant-sum game. The cards that were played get discarded. Once all cards have been played, a player's score is the sum of the prizes that she has won.

The LP solver we used is GLPK 4.63 (GLPK, 2017). We had to make the following changes to GLPK. First, we had to expose some internal routines so that we could input to the solver rational numbers rather than double-precision numbers. Second, we fixed a glitch in GLPK's rational LP solver in its pivoting step (it was not correct when the rational numbers were too small). Our code and GLPK use the GNU GMP library to provide arbitrary-precision arithmetic. The code, written in the C++14 language, was compiled with the g++ 7.2.0 compiler. It was run on a single thread on a 2.3 GHz Intel Xeon processor. The results are shown in Figure 11.3.

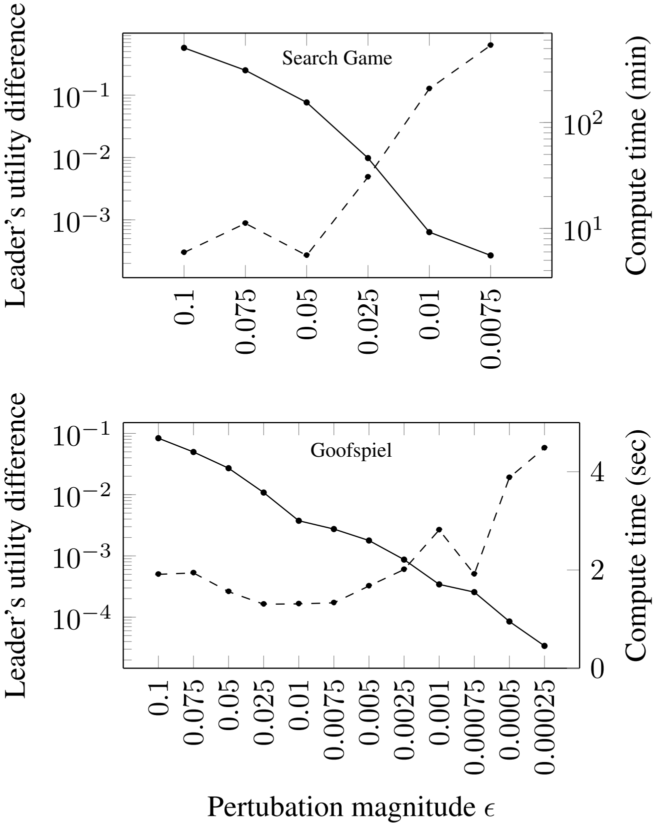


Figure 11.3: *Experiments. Dashed lines show compute time. Solid lines show the loss in the leader's utility compared to the SSE value in the unperturbed game.*

CHAPTER 12

Conclusions and Discussion

In this thesis, we significantly advanced the state of the art on equilibrium computation in Stackelberg games. In the first part of the work, we addressed settings involving a *single leader* and *multiple followers*, which, with the exception of very specific cases, were largely unexplored before. Then, in the second part of the work, we studied for the first time the problem of computing the strategies to commit to in Stackelberg games with *multiple leaders* and followers. Finally, in the last part of the work, we investigated how to refine the Stackelberg equilibrium in extensive-form Stackelberg games, providing the first application of *trembling-hand perfection* in Stackelberg settings.

In the rest of this chapter, we conclude the work with a final discussion on our results and future research directions.

12.1 Single-Leader Multi-Follower Stackelberg Games

We provided the first systematic study of the problem of computing Stackelberg equilibria in games with a single leader and multiple followers, focusing on the case in which the latter play a Nash equilibrium after observing the leader's commitment. Specifically, we addressed the case where the followers are restricted to pure strategies, for three reasons. First, the general case of mixed strategies is already known to be computationally intractable even in the basic setting of two-follower Stackelberg polymatrix games (Basilico et al., 2017a). Second, as we showed, the restriction to pure strategies leads to non-trivial computational results. Finally, this restriction is without loss of generality in games that always admit a pure-strategy Nash equilibrium, such as congestion games (Rosenthal, 1973).

In the setting of n-player normal-form Stackelberg games, after briefly showing that a *strong* equilibrium (*i.e.*, with the followers breaking ties in favor of the leader) can be computed in time polynomial in the size of the input, we extensively studied the problem of computing a *weak* equilibrium (*i.e.*, where the tie-breaking is against the leader), which is much more involved. Among the other results, we provided the first, to the best of our knowledge, exact algorithms for finding a weak Stackelberg equilibrium in settings beyond single-leader single-follower games (for an algorithm working in this case, see the work by Von Stengel and Zamir (2010)).

The algorithms we have proposed can constitute a useful framework for developing solution methods for games in which the normal-form representation cannot be assumed as input (such as, *e.g.*, succinct games of polynomial type like polymatrix and congestion games). Retaining the main structure of our algorithms, such games could be tackled by adapting the subproblems that are solved for each (relaxed) outcome configuration to the case where the followers' actions cannot be all taken into account explicitly. For outcomes in S^+, a cutting plane method could be employed to generate a best response for each of the followers iteratively, without having to generate all of them *a priori*. For outcomes in S^-, one could adopt a column generation approach to iteratively add sets $D_p(a_{-n}, a'_p)$ for different followers $p \in F$ and action profiles $a_{-n} \in S^-$, thus iteratively enlarging the set of strategies the leader could play to improve her utility while guaranteeing that the outcomes in S^- are not Nash equilibria.

Future developments along the research line of normal-form games, include establishing the approximability status of the problem with two followers (left open by Theorem 4.3), and the generalization to the case with both leader and followers playing mixed strategies, partially addressed in (Basilico et al., 2017a,b, 2019) (even though we conjecture that this problem could be much harder, probably Σ_2^p-hard).

As for Stackelberg polymatrix games, our main contribution was to identify classes of games in which an equilibrium can be computed in polynomial time once the number of players is fixed (while for the strong case the algorithm is a straightforward variation of the enumerative algorithm designed by Conitzer and Sandholm (2006), for the weak case we proposed a novel exact algorithm, see Algorithm 5.1). Besides being interesting for deriving computational complexity results, our classes of games are also useful in practice, as they model reasonable real-world security problems and they are equivalent to specific Stackelberg Bayesian games in which the follower may be of different types. Future developments may investigate more efficient implementations of our exact algorithm (Algorithm 5.1), for instance, enhancing it with a branch-and-bound scheme like that used by Jain et al. (2011) in the setting of Stackelberg Bayesian games.

In conclusion, as for Stackelberg congestion games, our main contribution was a comprehensive characterization of hard and easy game instances for the specific setting of games with singleton actions. While we also analyzed the impact of non-singleton actions (showing that the problems become highly intractable), possible future works may address what happens when the actions have specific structures, as it is the case, *e.g.*, in congestion games played on graphs, where the players' actions represent either paths (Fabrikant et al., 2004) or spanning trees (Werneck et al., 2000). On the algorithmic side, in this work we provided MILP formulations for the strong version of the problem. As discussed above, for the weak case one could adapt the exact algorithm proposed in the normal-form games setting, which leads to new research challenges.

12.2 Multi-Leader Multi-Follower Stackelberg Games

We introduced a new way to apply the Stackelberg paradigm to any (underlying) finite game. Differently from previous works, our approach deals with scenarios involving multiple leaders by introducing a preliminary agreement stage in which each leader can decide whether to be a leader or become a follower. We defined and studied three natural solution concepts that differ depending on the properties that they require on the agreement stage (other solution concepts, *e.g.*, requiring stability and perfect efficiency, will be explored in future).

Our equilibria generalize the optimal correlated strategies to commit to introduced by Conitzer and Korzhyk (2011) for single-leader multi-follower Stackelberg games. At the same time, they also provided a significant advancement over the multi-leader solution concepts introduced in the security context (see, *e.g.*, (Gan et al., 2018)). First, correlated-strategy commitments are more natural than leaders' strategies satisfying some Nash-like constraints. Secondly, our equilibria are funded on strong game-theoretic groundings, as they are guaranteed to exist independently of the game structure. Last but not least our solutions apply to general games.

Finally, our computational findings related to multi-leader Stackelberg games exploit a general framework relying on a game-independent stability oracle. Thus, our positive results can be extended to other game classes by simply designing polynomial-time oracles.

In future, we will investigate new ways to model the agreement stage involving the leaders. One possibility is to adopt *cooperative* solution concepts at the leaders' level. Moreover, we will also study how our model can be applied in practical applications.

12.3 Refinements of the Stackelberg Equilibrium

We initiated the study of equilibrium refinement based on trembling-hand perfection in extensive-form Stackelberg games. To the best of our knowledge, this is the first solution concept that guarantees off-equilibrium-path optimality in extensive-form Stackelberg games. We studied the equilibrium space of all the Stackelberg equilibria (containing both strong and weak Stackelberg equilibria), and showed that it is complete with respect to the limit points induced by perturbation schemes. We showed that this is not the case for strong and weak Stackelberg equilibria. We also showed that deciding the existence of any Stackelberg equilibrium—refined or not—giving the leader expected value of at least ν is NP-hard.

Then, we focused on quasi-perfection in Stackelberg settings. We provided a game-theoretic, axiomatic definition of quasi-perfect Stackelberg equilibrium. We developed a family of game perturbation schemes that lead to a quasi-perfect Stackelberg equilibrium in the limit. Our family generalizes prior perturbation schemes introduced for finding (non-Stackelberg) quasi-perfect equilibria. Using our perturbation schemes, we developed a branch-and-bound algorithm for quasi-perfect Stackelberg equilibrium. It leverages a perturbed variant of the LP for computing a Stackelberg extensive-form correlated equilibrium. Experiments show that our algorithm can be used to find an approximate quasi-perfect Stackelberg equilibrium in games with thousands of nodes.

We showed that some perturbation schemes outside our family do not lead to quasi-perfect Stackelberg equilibria in some games. It remains an open question whether our perturbation family fully characterizes the whole set of such equilibria. As to the first requirement in Definition 10.4, can all the quasi-perfect Stackelberg equilibria be captured by perturbation schemes that only use polynomial lower bounds on trembles?

It was recently shown that in non-Stackelberg extensive-form games, there exists a perturbation size that is small enough (while still strictly positive) that an exact refined (*e.g.*, quasi-perfect) equilibrium can be found by solving a mathematical program with that perturbation size (Miltersen and Sørensen, 2010; Farina and Gatti, 2017a; Farina et al., 2018a), and Farina et al. (2018a) provide an algorithm for checking whether a given guess of perturbation size is small enough. That obviates the need to try to explicitly compute a limit of a sequence. It would be interesting to see whether such theory can also be developed for Stackelberg extensive-form games—and for our perturbation family in particular.

Bibliography

H. Ackermann, H. Röglin, and B. Vöcking. On the impact of combinatorial structure on congestion games. *Journal of the ACM (JACM)*, 55(6):25, 2008.

F. A. Al-Khayyal and J. E. Falk. Jointly constrained biconvex programming. *Mathematics of Operations Research*, 8(2):273–286, 1983.

E. Amaldi, A. Capone, S. Coniglio, and L. G. Gianoli. Network optimization problems subject to max-min fair flow allocation. *IEEE COMMUN LETT*, 17(7):1463–1466, 2013.

B. An, J. Pita, E. Shieh, M. Tambe, C. Kiekintveld, and J. Marecki. Guards and Protect: Next generation applications of security games. *ACM SIGecom Exchanges*, 10(1):31–34, 2011.

S. P. Anderson and G. Glomm. Incumbency effects in political campaigns. *Public Choice*, 74(2): 207–219, 1992.

E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4): 1602–1623, 2008.

S. Arora and B. Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

R. J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of mathematical Economics*, 1(1):67–96, 1974.

R. Avenhaus, A. Okada, and S. Zamir. Inspector leadership with incomplete information. In *Game equilibrium models IV*, pages 319–361. Springer, 1991.

N. Basilico, S. Coniglio, and N. Gatti. Methods for finding leader-follower equilibria with multiple followers: (extended abstract). In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (AAMAS 2016)*, pages 1363–1364, 2016.

N. Basilico, S. Coniglio, and N. Gatti. Methods for finding leader–follower equilibria with multiple followers. *arXiv preprint arXiv:1707.02174*, 2017a.

N. Basilico, S. Coniglio, N. Gatti, and A. Marchesi. Bilevel programming approaches to the computation of optimistic and pessimistic single-leader-multi-follower equilibria. In *SEA*, volume 75, pages 1–14. Schloss Dagstuhl-Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, 2017b.

N. Basilico, G. De Nittis, and N. Gatti. Adversarial patrolling with spatially uncertain alarm signals. *Artificial Intelligence*, 246:220–257, 2017c.

N. Basilico, S. Coniglio, N. Gatti, and A. Marchesi. Bilevel programming methods for computing single-leader-multi-follower equilibria in normal-form and polymatrix games. *EURO Journal on Computational Optimization*, pages 1–29, 2019.

D. Bertsimas and J. N. Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

M. Blonski. Characterization of pure strategy equilibria in finite anonymous games. *Journal of Mathematical Economics*, 34(2):225–233, 2000.

B. Bošanský and J. Cermak. Sequence-form algorithm for computing Stackelberg equilibria in extensive-form games. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2015)*, pages 805–811, 2015.

B. Bošanskỳ, C. Kiekintveld, V. Lisý, and M. Pěchouček. An exact double-oracle algorithm for zero-sum extensive-form games with imperfect information. *Journal of Artificial Intelligence Research*, pages 829–866, 2014.

B. Bošanskỳ, S. Brânzei, K. A. Hansen, T. B. Lund, and P. B. Miltersen. Computation of Stackelberg equilibria of finite sequential games. *ACM Transactions on Economics and Computation (TEAC)*, 5(4):23, 2017.

M. Braverman, Y. K. Ko, and O. Weinstein. Approximating the best Nash equilibrium in no (log n)-time breaks the exponential time hypothesis. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 970–982. SIAM, 2014.

M. Breton, A. Alj, and A. Haurie. Sequential Stackelberg equilibria in two-person games. *Journal of Optimization Theory and Applications*, 59(1):71–97, 1988.

N. Brown and T. Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

N. Brown and T. Sandholm. Superhuman AI for multiplayer poker. *Science*, 365(6456):885–890, 2019.

M. Campbell, A. J. Hoane Jr, and F.-h. Hsu. Deep blue. *Artificial intelligence*, 134(1-2):57–83, 2002.

A. Caprara, M. Carvalho, A. Lodi, and G. J. Woeginger. Bilevel knapsack with interdiction constraints. *INFORMS Journal on Computing*, 28(2):319–333, 2016.

M. Castiglioni, A. Marchesi, and N. Gatti. Be a leader or become a follower: The strategy to commit to with multiple leaders. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 123–129, 2019a.

M. Castiglioni, A. Marchesi, and N. Gatti. Be a leader or become a follower: The strategy to commit to with multiple leaders (extended version). *CoRR*, abs/1905.13106, 2019b. URL `http://arxiv.org/abs/1905.13106`.

M. Castiglioni, A. Marchesi, N. Gatti, and S. Coniglio. Leadership in singleton congestion games: What is hard and what is easy. *Artificial Intelligence*, 2019c.

A. Celli, A. Marchesi, T. Bianchi, and N. Gatti. Learning to correlate in multi-player general-sum sequential games. In *Advances in Neural Information Processing Systems*, 2019.

J. Cermak, B. Bošanský, K. Durkota, V. Lisý, and C. Kiekintveld. Using correlated strategies for computing Stackelberg equilibria in extensive-form games. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI 2016)*, pages 439–445, 2016.

N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

P. Cheeseman, B. Kanefsky, and W. Taylor. Where the really hard problems are. In *IJCAI*, pages 331–337, 1991.

B.-G. Chun, K. Chaudhuri, H. Wee, M. Barreno, C. H. Papadimitriou, and J. Kubiatowicz. Selfish caching in distributed systems: a game-theoretic analysis. In *Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing (PODC 2004)*, pages 21–30, 2004.

S. Coniglio, N. Gatti, and A. Marchesi. Pessimistic leader-follower equilibria with multiple followers. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 171–177, 2017.

S. Coniglio, N. Gatti, and A. Marchesi. Computing a pessimistic Stackelberg equilibrium with multiple followers: the mixed-pure case. *Algorithmica*, 2019.

V. Conitzer and D. Korzhyk. Commitment to correlated strategies. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2011)*, pages 632–637, 2011.

V. Conitzer and T. Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 82–90, 2006.

V. Conitzer and T. Sandholm. New complexity results about Nash equilibria. *Games and Economic Behavior*, 63(2):621–641, 2008.

J. Correa, J. De Jong, B. De Keijzer, and M. Uetz. The curse of sequentiality in routing games. In *International Conference on Web and Internet Economics*, pages 258–271. Springer, 2015.

C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

J. de Jong and M. Uetz. The sequential price of anarchy for atomic congestion games. In *International Conference on Web and Internet Economics*, pages 429–434. Springer, 2014.

G. De Nittis, A. Marchesi, and N. Gatti. Computing the strategy to commit to in polymatrix games. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 989–996, 2018a.

G. De Nittis, A. Marchesi, and N. Gatti. Computing the strategy to commit to in polymatrix games (extended version). *CoRR*, abs/1807.11914, 2018b. URL http://arxiv.org/abs/1807.11914.

A. Deligkas, J. Fearnley, and R. Savani. Inapproximability results for approximate Nash equilibria. In *International Conference on Web and Internet Economics*, pages 29–43. Springer, 2016.

V. DeMiguel and H. Xu. A stochastic multiple-leader Stackelberg model: analysis, computation, and application. *Operations Research*, 57(5):1220–1235, 2009.

E. Diamantoudi. Stable cartels revisited. *Economic Theory*, 26(4):907–921, 2005.

B. C. Eaves. Polymatrix games with joint constraints. *SIAM Journal on Applied Mathematics*, 24 (3):418–423, 1973.

A. Fabrikant, C. Papadimitriou, and K. Talwar. The complexity of pure Nash equilibria. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 604–612. ACM, 2004.

G. Farina and N. Gatti. Extensive-form perfect equilibrium computation in two-player games. In *AAAI*, pages 502–508, 2017a.

G. Farina and N. Gatti. Adopting the cascade model in ad auctions: Efficiency bounds and truthful algorithmic mechanisms. *Journal of Artificial Intelligence Research*, 59:265–310, 2017b.

G. Farina, C. Kroer, and T. Sandholm. Regret minimization in behaviorally-constrained zero-sum games. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1107–1116. JMLR. org, 2017.

G. Farina, N. Gatti, and T. Sandholm. Practical exact algorithm for trembling-hand equilibrium refinements in games. In *Advances in Neural Information Processing Systems*, pages 5039–5049, 2018a.

G. Farina, A. Marchesi, C. Kroer, N. Gatti, and T. Sandholm. Trembling-hand perfection in extensive-form games with commitment. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 233–239, 2018b.

D. Fotakis. Stackelberg strategies for atomic congestion games. *Theory of Computing Systems*, 47 (1):218–249, 2010.

D. Fotakis, S. Kontogiannis, E. Koutsoupias, M. Mavronicolas, and P. Spirakis. The structure and complexity of Nash equilibria for a selfish routing game. *Theoretical Computer Science*, 410(36): 3305–3326, 2009.

J. Gan, E. Elkind, and M. Wooldridge. Stackelberg security games with multiple uncoordinated defenders. In *Proceedings of the 17th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 703–711. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

N. Gatti, A. Lazaric, M. Rocco, and F. Trovò. Truthful learning mechanisms for multi-slot sponsored search auctions with externalities. *Artificial Intelligence*, 227:93–139, 2015.

GLPK. Gnu linear programming kit, version 4.63, 2017. URL `http://www.gnu.org/software/glpk/glpk.html`.

S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

J. Hartline, V. Syrgkanis, and E. Tardos. No-regret learning in bayesian games. In *Advances in Neural Information Processing Systems*, pages 3061–3069, 2015.

J. T. Howson Jr. Equilibria of polymatrix games. *Management Science*, 18(5-part-1):312–318, 1972.

J. T. Howson Jr and R. W. Rosenthal. Bayesian equilibria of finite two-person games with incomplete information. *Management Science*, 21(3):313–315, 1974.

S. Ieong, R. McGrew, E. Nudelman, Y. Shoham, and Q. Sun. Fast and compact: A simple class of congestion games. In *AAAI*, volume 5, pages 489–494, 2005.

M. Jain, C. Kiekintveld, and M. Tambe. Quality-bounded solutions for finite bayesian stackelberg games: Scaling up. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 997–1004. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

E. Janovskaja. Equilibrium situations in multi-matrix games. *Litovskiı Matematicheskiı Sbornik*, 8: 381–384, 1968.

A. X. Jiang and K. Leyton-Brown. A general framework for computing optimal correlated equilibria in compact games. In *International Workshop on Internet and Network Economics*, pages 218–229. Springer, 2011.

A. X. Jiang and K. Leyton-Brown. Polynomial-time computation of exact correlated equilibrium in compact games. *Games and Economic Behavior*, 91:347–359, 2015.

D. S. Johnson and M. R. Garey. *Computers and intractability: A guide to the theory of NP-completeness*. WH Freeman, 1979.

R. M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

M. Kearns, M. L. Littman, and S. Singh. Graphical models for game theory. *arXiv preprint arXiv:1301.2281*, 2013.

M. J. Kearns, M. L. Littman, and S. P. Singh. Graphical models for game theory. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence (UAI 2001)*, pages 253–260, 2001.

C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez, and M. Tambe. Computing optimal randomized resource allocations for massive security games. In *AAMAS*, pages 689–696, 2009.

D. Konur and J. Geunes. Competitive multi-facility location games with non-identical firms and convex traffic congestion costs. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):373–385, 2012.

C. Kroer, G. Farina, and T. Sandholm. Smoothing method for approximate extensive-form perfect equilibrium. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 295–301, 2017.

C. Kroer, G. Farina, and T. Sandholm. Robust Stackelberg equilibria in extensive-form games and extension to limited lookahead. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 1130–1137, 2018.

H. W. Kuhn. Extensive games and the problem of information, contributions to the theory of games II. *Annals of Mathematics Studies*, 28:193–216, 1953.

A. A. Kulkarni and U. V. Shanbhag. A shared-constraint approach to multi-leader multi-follower games. *Set-valued and variational analysis*, 22(4):691–720, 2014.

M. Labbé and A. Violin. Bilevel programming and price setting problems. *Annals of Operations Research*, 240(1):141–169, 2016.

M. Labbé, P. Marcotte, and G. Savard. A bilevel model of taxation and its application to optimal highway pricing. *Management science*, 44(12-part-1):1608–1622, 1998.

A. Laszka, J. Lou, and Y. Vorobeychik. Multi-defender strategic filtering against spear-phishing attacks. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

R. P. Leme, V. Syrgkanis, and É. Tardos. The curse of simultaneity. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 60–67. ACM, 2012.

J. Letchford and V. Conitzer. Computing optimal strategies to commit to in extensive-form games. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 83–92. ACM, 2010.

J. Letchford, V. Conitzer, and K. Munagala. Learning and approximating the optimal strategy to commit to. In *International Symposium on Algorithmic Game Theory*, pages 250–262. Springer, 2009.

S. Leyffer and T. Munson. Solving multi-leader–common-follower games. *Optimisation Methods & Software*, 25(4):601–623, 2010.

K. Leyton-Brown and M. Tennenholtz. Local-effect games. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 772–780, 2003.

J. Lou and Y. Vorobeychik. Equilibrium analysis of multi-defender security games. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

J. Lou, A. M. Smith, and Y. Vorobeychik. Multidefender security games. *IEEE Intelligent Systems*, 32(1):50–60, 2017.

Z.-Q. Luo, J.-S. Pang, and D. Ralph. *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996.

A. Marchesi, S. Coniglio, and N. Gatti. Leadership in singleton congestion games. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 447–453, 2018a.

A. Marchesi, G. Farina, C. Kroer, N. Gatti, and T. Sandholm. Quasi-perfect Stackelberg equilibrium. *CoRR*, abs/1811.03871, 2018b. URL `http://arxiv.org/abs/1811.03871`.

A. Marchesi, M. Castiglioni, and N. Gatti. Leadership in congestion games: Multiple user classes and non-singleton actions. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 485–491, 2019a.

A. Marchesi, M. Castiglioni, and N. Gatti. Leadership in congestion games: Multiple user classes and non-singleton actions (extended version). *CoRR*, abs/1905.13108, 2019b. URL `http://arxiv.org/abs/1905.13108`.

A. Marchesi, G. Farina, C. Kroer, N. Gatti, and T. Sandholm. Quasi-perfect Stackelberg equilibrium. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019.*, pages 2117–2124, 2019c.

J. Matuschke, S. T. McCormick, G. Oriolo, B. Peis, and M. Skutella. Protection of flows under targeted attacks. *OPER RES LETT*, 45(1):53–59, 2017.

G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part i-convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

J.-F. Mertens. Two examples of strategic equilibrium. *Games and Economic Behavior*, 8(2):378–388, 1995.

P. B. Miltersen and T. B. Sørensen. Computing a quasi-perfect equilibrium of a two-player game. *Economic Theory*, 42(1):175–192, 2010.

D. Monderer and L. S. Shapley. Potential games. *Games and economic behavior*, 14(1):124–143, 1996.

H. Moulin and J.-P. Vial. Strategically zero-sum games: the class of games whose completely mixed equilibria cannot be improved upon. *International Journal of Game Theory*, 7(3-4):201–221, 1978.

J. Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic game theory*. Cambridge university press, 2007.

E. Nudelman, J. Wortman, Y. Shoham, and K. Leyton-Brown. Run the gamut: A comprehensive approach to evaluating game-theoretic algorithms. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 880–887. IEEE Computer Society, 2004.

J.-S. Pang and M. Fukushima. Quasi-variational inequalities, generalized Nash equilibria, and multi-leader-follower games. *Computational Management Science*, 2(1):21–56, 2005.

C. H. Papadimitriou and T. Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM (JACM)*, 55(3):14, 2008.

P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. 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*, pages 895–902, 2008.

I. Romanovskii. Reduction of a game with complete memory to a matrix game. *Soviet Mathematics*, 3, 1962.

R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.

S. M. Ross. Goofspiel-the game of pure strategy. *Journal of Applied Probability*, 8(3):621–625, 1971.

T. Roughgarden. Stackelberg scheduling strategies. *SIAM journal on computing*, 33(2):332–350, 2004.

N. V. Sahinidis. *BARON 14.3.1: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2014.

T. Sandholm, A. Gilpin, and V. Conitzer. Mixed-integer programming methods for finding Nash equilibria. In *AAAI*, pages 495–501, 2005.

W. H. Sandholm. Evolutionary implementation and congestion pricing. *The Review of Economic Studies*, 69(3):667–689, 2002.

R. Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *International journal of game theory*, 4(1):25–55, 1975.

Y. Sharma and D. P. Williamson. Stackelberg thresholds in network routing games or the value of altruism. *Games and Economic Behavior*, 67(1):174–190, 2009.

H. D. Sherali. A multiple leader Stackelberg model and analysis. *Operations Research*, 32(2): 390–404, 1984.

Y. Shoham and K. Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

A. Sinha, P. Malo, A. Frantsev, and K. Deb. Finding optimal strategies in a multi-period multi-leader–follower Stackelberg game using an evolutionary algorithm. *Computers & Operations Research*, 41:374–385, 2014.

S. Suri, C. D. Tóth, and Y. Zhou. Selfish load balancing and atomic congestion games. *Algorithmica*, 47(1):79–96, 2007.

C. Swamy. The effectiveness of Stackelberg strategies and tolls for network congestion games. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1133–1142. Society for industrial and applied mathematics, 2007.

M. Tambe. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge university press, 2011.

E. Van Damme. A relation between perfect equilibria in extensive form games and proper equilibria in normal form games. *International Journal of Game Theory*, 13(1):1–13, 1984.

E. Van Damme. *Stability and Perfection of Nash Equilibria*. Springer-Verlag, Berlin, Heidelberg, 1987. ISBN 0-387-17101-0.

H. Von Stackelberg. *Marktform und gleichgewicht*. J. springer, 1934.

B. Von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14 (2):220–246, 1996.

B. Von Stengel and F. Forges. Extensive-form correlated equilibrium: Definition and computational complexity. *Mathematics of Operations Research*, 33(4):1002–1022, 2008.

B. Von Stengel and S. Zamir. Leadership games with convex strategy sets. *Games and Economic Behavior*, 69(2):446–457, 2010.

R. Werneck, J. Setubal, and A. da Conceicao. Finding minimum congestion spanning trees. *Journal of Experimental Algorithmics (JEA)*, 5:11, 2000.

A. B. Zemkoho. Solving ill-posed bilevel programs. *Set-Valued and Variational Analysis*, 24(3): 423–448, 2016.

D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 681–690. ACM, 2006.