Theoretical foundations for planning in partially observable stochastic games

Aurélien Delage

N° d'ordre NNT : 2024ISAL0062

Thèse de doctorat de l'Université de Lyon

opérée au sein de

Institut National des Sciences Appliquées de Lyon

École Doctorale 512

Infomaths

Spécialité / Discipline de doctorat :

informatique

Soutenue publiquement le 28 juin 2024, par :

Aurélien Delage

Theoretical foundations of planning in Partially Observable Stochastic Games

---

Devant le jury composé de :

Christine Solnon	Professeure	INSA Lyon	Présidente
Aurélie Beynier	Maîtresse de Conférences (HDR)	Sorbonne Université	Rapportrice
Frédéric Koriche	Professeur	Université d'Artois	Examinateur
Régis Sabbadin	Directeur de recherche	INRAE, Toulouse	Examinateur
Bruno Zanuttini	Professeur	Université de Caen Normandie	Rapporteur
Jilles Dibangoye	Maître de Conférences (HDR)	Bernoulli Institute, Groningen	Directeur
Olivier Buffet	Chargé de recherche (HDR)	INRIA, Nancy	co-Directeur

Département FEDORA – INSA Lyon - Ecoles Doctorales

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
ED 206 CHIMIE	**CHIMIE DE LYON** https://www.edchimie-lyon.fr Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage secretariat@edchimie-lyon.fr	**M. Stéphane DANIELE** C2P2-CPE LYON-UMR 5265 Bâtiment F308, BP 2077 43 Boulevard du 11 novembre 1918 69616 Villeurbanne directeur@edchimie-lyon.fr
ED 341 E2M2	**ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION** http://e2m2.universite-lyon.fr Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.e2m2@univ-lyon1.fr	**Mme Sandrine CHARLES** Université Claude Bernard Lyon 1 UFR Biosciences Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69622 Villeurbanne CEDEX e2m2.codir@listes.univ-lyon1.fr
ED 205 EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTÉ** http://ediss.universite-lyon.fr Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.ediss@univ-lyon1.fr	**Mme Sylvie RICARD-BLUM** Laboratoire ICBMS - UMR 5246 CNRS - Université Lyon 1 Bâtiment Raulin - 2ème étage Nord 43 Boulevard du 11 novembre 1918 69622 Villeurbanne Cedex Tél : +33(0)4 72 44 82 32 sylvie.ricard-blum@univ-lyon1.fr
ED 34 EDML	**MATÉRIAUX DE LYON** http://ed34.universite-lyon.fr Sec. : Yann DE ORDENANA Tél : 04.72.18.62.44 yann.de-ordenana@ec-lyon.fr	**M. Stéphane BENAYOUN** Ecole Centrale de Lyon Laboratoire LTDS 36 avenue Guy de Collongue 69134 Ecully CEDEX Tél : 04.72.18.64.37 stephane.benayoun@ec-lyon.fr
ED 160 EEA	**ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE** https://edeea.universite-lyon.fr Sec. : Philomène TRECOURT Bâtiment Direction INSA Lyon Tél : 04.72.43.71.70 secretariat.edeea@insa-lyon.fr	**M. Philippe DELACHARTRE** INSA LYON Laboratoire CREATIS Bâtiment Blaise Pascal, 7 avenue Jean Capelle 69621 Villeurbanne CEDEX Tél : 04.72.43.88.63 philippe.delachartre@insa-lyon.fr
ED 512 INFOMATHS	**INFORMATIQUE ET MATHÉMATIQUES** http://edinfomaths.universite-lyon.fr Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 infomaths@univ-lyon1.fr	**M. Hamamache KHEDDOUCI** Université Claude Bernard Lyon 1 Bât. Nautibus 43, Boulevard du 11 novembre 1918 69 622 Villeurbanne Cedex France Tél : 04.72.44.83.69 direction.infomaths@listes.univ-lyon1.fr
ED 162 MEGA	**MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE** http://edmega.universite-lyon.fr Sec. : Philomène TRECOURT Tél : 04.72.43.71.70 Bâtiment Direction INSA Lyon mega@insa-lyon.fr	**M. Etienne PARIZET** INSA Lyon Laboratoire LVA Bâtiment St. Exupéry 25 bis av. Jean Capelle 69621 Villeurbanne CEDEX etienne.parizet@insa-lyon.fr
ED 483 ScSo	**ScSo**[1] https://edsciencessociales.universite-lyon.fr Sec. : Mélina FAVETON Tél : 04.78.69.77.79 melina.faveton@univ-lyon2.fr	**M. Bruno MILLY** (INSA : J.Y. TOUSSAINT) Univ. Lyon 2 Campus Berges du Rhône 18, quai Claude Bernard 69365 LYON CEDEX 07 Bureau BEL 319 bruno.milly@univ-lyon2.fr

[1] ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Remerciements

Cher lecteur, le document que vous vous apprêtez à lire est le fruit, certes, de plusieurs années de travail, mais surtout d'un Aurélien qui a été soutenu et choyé pendant trois longues années par des tas de gens. Ce document n'existerait pas si ces dernier.es n'avaient pas été là. Je vous dois ce titre de docteur, et la fierté d'être allé jusqu'au bout. Ainsi, vous trouverez des remerciements spécifiques dans les prochaines lignes.

Je commence par remercier Pr. Bruno Zanuttini et Dr. Aurélie Beynier pour leur relecture attentive de mon manuscrit. Je remercie aussi tous les membres constituant mon jury de thèse pour leurs questions pertinentes posées durant la soutenance. Les diverses discussions furent particulièrement enrichissantes.

Il est coutume, dans ses remerciements, d'avoir un mot pour ses directeurs de thèse. Cependant, mes remerciements pour Jilles et Olivier sont profondément sincères. Un immense merci pour votre soutien, votre compréhension et votre patience pendant les périodes difficiles. En ce qui concerne la science, j'ai évidemment appris énormément en travaillant avec vous. Votre rigueur, votre intuition et votre culture m'auront fait grandir en tant qu'apprenti sorcier (chercheur, pardon). J'ai commencé à apprendre à communiquer, à réfléchir plus rigoureusement, à prendre du recul et à imaginer. Je suis conscient de la chance que j'ai eue de pouvoir préparer ma thèse sous votre encadrement. En bref, merci pour tout.

Merci à Guillaume, Christine, Lionel et Nicolas pour tous nos débats de société, pour nos désaccords et nos accords, mais aussi et surtout pour le soutien. Nos discussions ont fait mûrir mes opinions, et m'en ont fait découvrir beaucoup d'autres, dont la pertinence ne cessera d'alimenter mes réflexions, mais bien pour important pour moi, vous avez pris le temps d'écouter un doctorant parfois en détresse, et je vous en remercie sincèrement.

Au-delà de l'encadrement que j'ai reçu, une multitude de proches m'ont accompagné et ont finalement permis que cette thèse soit soutenue. Merci à mes collègues de bureau, Guillaume, Alix et Maxime de m'avoir supporté et d'avoir gardé mon bureau au chaud. Merci à Johan, que je n'ai eu la chance de connaître que cette dernière année, mais qui aura réussi à m'impressionner scientifiquement. Merci à David pour nos premières sorties post-covid, pour tout ce que tu m'as appris humainement, et pour tout ton soutien. Globalement, merci à toute la troupe de collègue doctorants chez qui j'ai trouvé du réconfort, de l'amusement et beaucoup de conseils vélo. Merci à Audrey, Clémentine et Thomas, pour avoir fait de mes retours à la campagne des bulles d'oxygène. Merci à Elsa, pour ta bonne humeur, pour les discussions Tour de France et pour ton aide. Je t'en suis très sincèrement reconnaissant et ai hâte de pouvoir te montrer, dans les forêts vosgiennes, que j'ai appris à prendre la roue. Hiba[1], je ne sais pas bien où te placer dans cette longue liste de personnes à remercier. Tu n'as pas seulement été d'un soutien précieux ces deux dernières années, tu as fait mon bonheur. J'ai hâte que ta dernière année de thèse pointe le bout de son nez, pour pouvoir tenter de te montrer à quel point tu as été incroyable.

Enfin, mes derniers et plus profonds remerciements vont à ma famille. Maman, Papa, Marc, Pauline et Eliott, il m'est bien difficile d'exprimer l'ampleur de ma gratitude pour votre soutien ces dernières années. Assurément, votre présence, votre soutien inconditionnel et vos encouragements ont été indispensables et j'y ai puisé l'énergie nécessaire pour construire, brique après brique, chaque jour, ce document. Un infini merci que j'espère pouvoir faire durer encore bien longtemps. Au fond, ce document est pour vous.

[1] A wise owl once said "there is a prominent theory stating that owls are the most magnificent creatures on earth ! indeed this theory is true, my sources ? well common sense !"

Pour Carole et Michel.
Ce document est pour vous.

Abstract

A recent theory suggests recasting common-payoff POSGs into non-observable problems through the introduction of an appropriate statistic. Doing so offers additional levers to search for optimal behaviors. Showing that Bellman's optimality principle applies in the non-observable game allows applying powerful algorithms designed for fully observable games (*e.g.,* heuristic search value iteration). The algorithms, taking advantage of the discovered levers (*e.g.,* breaking problems into subproblems; allowing knowledge generalization between subproblems), offer theoretical convergence guarantees and empirically competitive results. Still, while this approach has succeeded in subclasses of 2-player zero-sum partially observable stochastic games (zs-POSG), how to apply it in the general case still remains an open question. Besides, recasting the original problem in a non-observable one introduces decision problems at every time step whose time and memory complexities become prohibitive for large games.

In the first contribution of this manuscript, we address the first concern and propose for the first time an HSVI-like solver that provably converges to an ϵ-optimal solution of any zs-POSG in finite time. This opens the door to a novel family of promising approaches complementing those relying on linear programming or iterative methods.

In a second contribution of this manuscript, we take a look at games involving n players but assuming (i) that they all share the same payoff function and (ii) a hierarchical knowledge structure (*i.e.,* each agent knows what their subordinate knows, and so forth). We show that a specialization of point-based value iteration efficiently takes advantage of particular levers offered by this subclass. This work paves the way to multiple extensions of the proposed hierarchical structure while maintaining the scalability of point-based algorithms.

In the final contribution of this manuscript, we present a related, although adjacent, contribution to min-max optimization problems with mild continuity properties.

Keywords: Computational game theory, zero-sum games, partial observability, dynamic programming, heuristic search, min-max optimization, games with common payoff, hierarchical information sharing.

Résumé

Une théorie récente suggère de reformuler les POSG à gain commun en des problèmes non observables via l'introduction d'une statistique suffisante appropriée, ce qui offre des leviers supplémentaires pour rechercher des plans optimaux. Montrer que le principe d'optimalité de Bellman s'applique sur le jeu non-observable permet l'application d'algorithmes efficaces conçus pour les jeux complètement observables (tels que *heuristic search value iteration*). Les algorithmes exploitant les leviers découverts (par exemple la division des problèmes en sous-problèmes; la généralisation des connaissances entre les sous-problèmes) offrent une garantie de convergence théorique et des résultats compétitifs sur le plan empirique. Cependant, bien que cette approche ait réussi dans des sous-classes de jeux stochastiques partiellement observables à somme nulle et à deux joueurs (zs-POSG), comment l'appliquer dans le cas général reste une question ouverte. De plus, reformuler le problème original en un problème non-observable introduit des problèmes de décision à chaque étape, dont les complexités temporelle et mémorielle deviennent prohibitives pour les jeux de grande envergure.

Dans la première contribution de ce manuscrit, nous abordons la première préoccupation et proposons pour la première fois un solveur de type *heuristic search value iteration* dont nous démontrons qu'il converge vers une solution ϵ-optimale en temps fini pour n'importe quel zs-POSG. Cela ouvre la voie à une nouvelle famille d'approches prometteuses et complémentaires à celles reposant sur la programmation linéaire ou les méthodes itératives.

Dans une deuxième contribution de ce manuscrit, nous examinons des jeux impliquant n joueurs et en supposant (i) qu'ils partagent tous la même fonction de récompense et (ii) que les joueurs sont organisés selon une structure de connaissance hiérarchique (c.-à-d. chaque agent sait ce que son subordonné sait, et ainsi de suite). Nous montrons qu'une spécialisation du schéma algorithmique *point-based value iteration* tire efficacement parti des leviers offerts par cette sous-classe. Ce travail ouvre la voie à de multiples extensions de la structure hiérarchique proposée tout en conservant le passage à l'échelle du schéma algorithmique proposé.

Dans la dernier contribution de ce manuscrit, nous présentons une contribution connexe, bien qu'annexe, aux problèmes d'optimisation min-max avec des propriétés de continuité faibles.

Mots-clés: Théorie des jeux computationnelle, jeux à somme nulle, observabilité partielle, programmation dynamique, recherche heuristique, optimisation min-max, jeux à récompense commune, partage d'information hiérarchique.

Contents

List of Figures

List of Tables

List of Algorithms

List of Theorems

1

General Introduction

Contents

The introduction chapter of this manuscript firstly lays down general concepts and key challenges in game theory. Secondly, we circumscribe the topics of interest of this manuscript, mainly addressing the problem of optimally planning in imperfect information games through dynamic programming. We then discuss the similarities and differences between two common frameworks for such games (POSGs and EFGs). Finally, we focus on the first one and detail the challenges that have prevented the application of dynamic programming to optimally planning in zero-sum POSGs, along with the levers this algorithmic scheme might offer.

1.1 Game Theory

Game theory is a branch of mathematics that studies the strategic interactions between any number of players. It is commonly acknowledged (Schwalbe et al. 2001) that the first formal definitions published on the topic of game theory can be attributed to Ernst Zermelo, showing in 1913 that, in chess, one side can either force a win or a draw. A few years later, Émile Borel introduced crucial questions about optimally playing in competitive games, pointing out the importance of finding strategies that maximize the expected reward against *any* possible opponent (Borel 1921), which will later be linked to von Neumann's minimax theorem (von Neumann 1928). Thanks to the communications of Emile Borel to the French Academy of Science, the importance that game theory might have in economics, military and psychology was already known in the early 1920s.

Game theory has been largely studied ever since, offering crucial frameworks (Kuhn et al. 1953; Bernstein et al. 2002), interesting results and practical solutions for a wide variety of fields. From entertainment (Campbell et al. 2002; Silver et al. 2018; Vinyals et al. 2019) to economics

(Gibbons 1992), through security (Horák 2019; Manshaei et al. 2013), physics (Hauert et al. 2005; Brunner et al. 2013), engineering (Tang et al. 2016) and communications (Han et al. 2012), game theory holds substantial practical or potential influence on our daily life. Interestingly, game theory has also proven useful in enhancing the understanding of the climate system through the design of specific games (Meadows et al. 2016) and/or illustrating the interdependence of biophysical flows in a context of climate crisis (Boissier et al. 2023).

Generally speaking, players are modeled by a possibly infinite set of actions. Those actions impact the game by both influencing the outcome players receive and making the state of the game evolve. The expected outcome of the game depending on all players' actions make it non-trivial to define optimal behaviors. In particular, players must take into account their immediate outcome, the game possibly evolving into a state with negative expected payoff and the other players' behaviors. The main problem is thereby to (i) design a pertinent solution concept and (ii) be able to compute it through solving optimization problems.

The solution concept of Nash equilibria was introduced in 1950 by John Nash and tackles point (i) mentioned above. It consists in a strategy profile (*i.e.,* a tuple containing a strategy for each player) from which no player has incentive in unilaterally deviating. Nash equilibria are stable points in players' search to maximize their expected outcome, provided that they act rationally. In other words, given a Nash equilibrium, no player has incentive in unilaterally continuing her search.

The ability to efficiently find Nash equilibria is, for its part, a major stake. Several classes of games admit algorithms solving nontrivial real-life games (Cohen-Solal 2020; Buffet et al. 2020; Horák et al. 2019a; Zang et al. 2023; Chadès et al. 2012; Van der Pol et al. 2016). Some others, however, remain challenging and are subject to extensive recent work (Sokota et al. 2023; Perolat et al. 2022; Lanctot et al. 2017; Brown et al. 2020; Ling et al. 2021; López et al. 2022)[2]. The difficulty of the search highly depends on, at least, (i) the number of players (Porter et al. 2008, Figure 11) and/or their cooperation degree, (ii) the game's payoff function's continuity properties (Fiez et al. 2021, Table 1) and (iii) the state of the game being perfectly known to players or not (high-level topic discussed by Burch (2018)). Negative complexity results show for example that computing Nash equilibria for general-sum normal-form games is already in PPAD-complete (Goldberg et al. 2006; Daskalakis et al. 2009), even for two-players (Chen et al. 2009), while normal-form games are among the simplest ones considered in game theory.

When considering games that are no longer convex-concave w.r.t. players' strategies, deciding wether a function $f : [0,1]^d \mapsto [-1,1]$ has an ϵ-approximate "minimax strategy" is already NP-hard (Daskalakis et al. 2021). Finally, games modeling uncertainty are especially hard to tackle while being at the center of daily life practical problems. Uncertainty implies that players are unaware of the precise state of the game, and do not observe each other's actions, making the search for a Nash equilibrium even harder.

Partially observable stochastic games (POSGs) (Hansen et al. 2004) is a generic n-player framework for dynamic games with imperfect information. Players partially observe the current state of the game through noisy (and possibly partial) observations. Even though sequences of actions and observations of all players are sufficient to infer a probability distribution over the possible states of the game, this knowledge remains unknown to players that, in general, do not know their opponents' actions and observations. Each time players take actions, they are rewarded depending on the current state of the game. An influence diagram representing such games is given in Figure 1.1. When there are two players whose reward functions are opposite[4], the game is called a zero-sum POSG. Specific time complexity results were established for POSGs and depend on how much players are cooperating or competing. While common-payoff POSGs (cp-POSGs) are NEXP-hard (Bernstein et al. 2002), competition adds complexity as deciding whether a strategy with positive expected reward in a POSG with $2k$ ($k \geqslant 2$) players exists has been shown to be NEXP^{NP}-complete (Goldsmith et al. 2007). Two-player zero-sum POSGs can be solved by linear

[2]Interestingly, a significant part of recent work[3]focuses on explicitly or implicitly transforming the games with imperfect information into games with perfect information and retrieving from it "good" strategies for the original game.

[3]or am I biased?

[4]We discard the zero-sum case with more than two players.

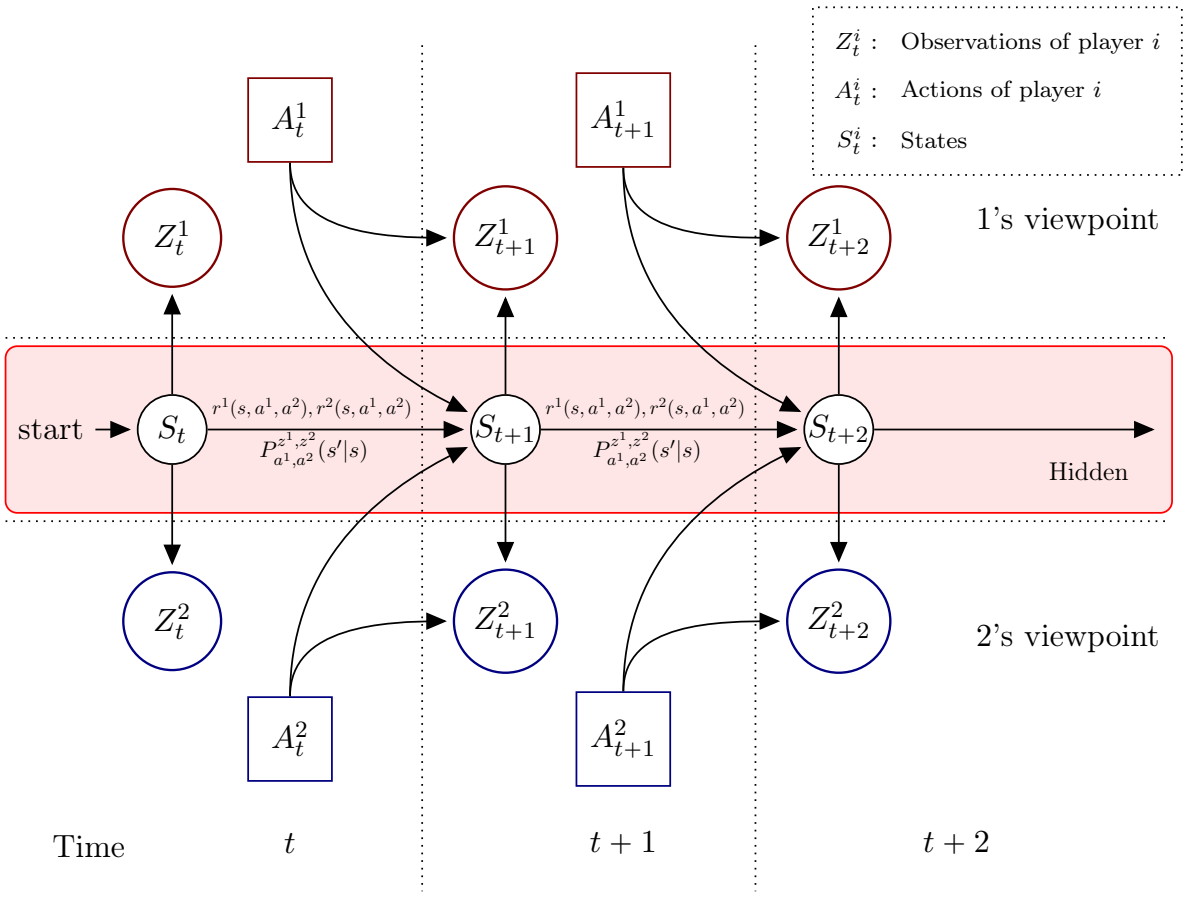


Figure 1.1: Dynamic influence diagram representing the evolution of a `zs-POSG`

programs (obtained by flattening time) that run in linear time and space with respect to the size of the game tree (Koller et al. 1994). Unfortunately, the size of the game tree is exponential with respect to the players' numbers of actions and observations.

While planning in `POSG`s might be a challenging computational task in general, many real-life applications possess structure which can be exploited by algorithms to find solutions empirically way faster than generic algorithms. Exhibiting tractable subclasses of `POSG`s led to various applications of game theory to real life (Becker et al. 2004; Dibangoye et al. 2012; Nair et al. 2005; Dibangoye et al. 2014b; Nayyar et al. 2010; Horák et al. 2017; Horák et al. 2019b; Hadfield-Menell et al. 2016).

1.2 Computational Aspects in Game Theory

The study of game theory raises at least two orthogonal problems. Firstly, it involves the challenge of mathematically modeling real-life situations with various types of interactions between players and the game. Consequently, it must appropriately be defined what is considered as a "solution" (*e.g.,* Nash equilibria, trembling-hand Nash equilibria, Stackelberg equilibria, correlated equilibria). Even when a pertinent solution concept is identified, there might exist multiple behaviors verifying the conditions of the solution concept, but with different values for each player. Selecting one particular behavior is a non-trivial problem (Harsanyi et al. 1988).

Secondly, researchers have shown increasing interest in the problem of efficiently computing solutions for well-established models and solution concepts. This manuscript focuses on the latter problem, aiming at finding a Nash equilibrium of any given `zs-POSG` through a dynamic programming approach.

1.2.1 Methodologies to Tackle zs-POSGs

Multiple methodologies to tackle zs-POSGs have been introduced throughout the years. The ones mentioned in this paragraph will be presented in more detail thereafter. Of particular interest for this manuscript are linear programming, regret minimization and dynamic programming. Linear programming directly searches for Nash equilibria as distribution probabilities over *pure strategies* (decision trees mapping each reachable sequence of actions and observations to an action), while regret minimization and dynamic programming perform local computations for different parts of their game tree[5], and then extract a Nash equilibrium based on the resulting computations after a given time budget.

Regret minimization computes regrets at each decision point in the game tree (and thus traverses the whole game tree at each iteration) while dynamic programming implements a divide and conquer strategy by breaking problems into subproblems and backwardly updating the knowledge for each problem for all time steps. Linear programming, as it searches for a global solution by optimizing all its parameters at the same time, did not initially offer as many levers as regret minimization or dynamic programming to improve the original scheme. The two latter approaches performing local updates of knowledge highly benefited from smartly deciding which local computations to make and/or approximating the local solutions (Moravčík et al. 2017; Vojtěch Kovařík et al. 2022b; Smith et al. 2005; Horák et al. 2023; Johanson et al. 2012). Recently, double-oracle approaches to solve linear programs (besides efficiently finding Nash equilibria whose supports contain few pure strategies) nuanced this observation by offering very interesting levers (*e.g.,* approximating best responses (McAleer et al. 2021; Lanctot et al. 2017), novel constructions of restricted games (McAleer et al. 2021)).

Linear Programming Linear programming first appeared in game theory as a solving algorithm for two-player zero-sum normal-form games. Such games are among the simplest ones in game theory and allow for an efficient polynomial-time computation of a Nash equilibrium. It was later generalized to many other general settings, including (i) imperfect information about the game (Koller et al. 1996; Harsanyi 1968), (ii) any number of players sharing the same payoff function (Aras et al. 2010), (iii) two-player general-sum games (Lemke et al. 1964). Regarding imperfect information games, a technical discovery introduced by Koller et al. (1996) offered a more concise representation of strategies as realization weights, and the resulting linear program has exponentially lower complexity than the original one. Overall, linear programming clearly has benefits regarding its relative simplicity to implement, the computation of an *exact* solution and the possibility to iteratively construct the whole program (Bošanský et al. 2014; McAleer et al. 2021; Lanctot et al. 2017). On the downsides, very large games still remain a challenge as their respective complete[6] linear programs do not even fit in memory (Bowling et al. 2015).

Regret Minimization The concept of *regret minimization* finds roots in the single-agent setting and belongs to the set of schemes learning by performing self plays and asymptotically converging towards a solution.

The regret matching algorithm (Hart et al. 2000) tackles normal-form games and implements this scheme by specifying *regret* computation rules and strategy update procedures. Given an action a, the regrets are defined for all other actions k as the cumulative difference between rewards that would have been obtained by playing k instead of a in the past. In the case of zero-sum games, it holds that, if both players use a regret matching minimizer, the average strategy converges towards a Nash equilibrium (Hart et al. 2000). Overall, regret matching has been a building block for planning in zero-sum games. In particular, the state-of-the-art algorithm called *counterfactual regret minimization* (Zinkevich et al. 2007) builds upon this regret rule to tackle two-player zero-sum games with imperfect information, and has been at the core of many recent works (Moravčík et al. 2017; Brown et al. 2018), yielding impressive results. The algorithm's

[5] Both algorithms operate on different notions of game tree.

[6] Double oracle-based methods iteratively construct the linear program, but their convergence guarantee relies on the worst-case scenario of constructing the whole linear program. Besides, double oracle-based methods typically depend on best-response computations, which is known to be intractable for large-size games.

iterations having linear time complexity in the size of the game tree provides an algorithm running even for relatively large games. One of the drawbacks that recent work (Brown et al. 2020) attempted to tackle is not taking advantage of underlying structure that most real-life games possess. Indeed, despite decision points being linked to each other through continuity properties, CFR considers them independently.

Dynamic Programming Dynamic programming is another orthogonal approach that recursively divides difficult problems into easier subproblems. It is required that problems can be divided (for example, those containing multiple time steps at which a player must take actions) and have the property ensuring that a solution to a problem can be found using the solutions of subproblems. The latter property is known as *optimal substructure*[7] and not all problems satisfy it[8]. The "definition" of dynamic programming is very general and actually encompasses a wide variety of approaches. Of particular interest for this thesis are (i) Hansen et al.'s iterative pruning of dominated strategies for general POSGs, and (ii) the re-solving scheme (Burch et al. 2014), which we will discuss in further details in Section 3.3. A more detailed presentation of dynamic programming and its potential advantages compared to the two previous approaches is given later in this introduction, in Section 1.4.

1.2.2 Efficient Algorithms for Structured Games

As mentioned earlier, planning in POSGs in general is often a computationally particularly hard task. Still, real-life scenarios often exhibit structure. Multiple structures have been identified as of particular interest—*e.g.,* dynamics independence (Becker et al. 2004; Dibangoye et al. 2012), weak-separability (Nair et al. 2005; Dibangoye et al. 2014b), delayed information-sharing (Nayyar et al. 2010), one-sidedness (Horák et al. 2017; Horák et al. 2019b; Hadfield-Menell et al. 2016; Malik et al. 2018; Xie et al. 2020). Adapting all three algorithms described above to take advantage of these structures allows solving the problems empirically way faster. Exhibiting tractable subclasses of POSGs relevant for real-life applications is a very fruitful line of research in computational game theory.

1.3 Formalisms for Imperfect Information Games

Partially observable stochastic games (POSGs) and extensive-form games (EFGs) are two different formalisms modeling stochastic games with imperfect information. A wide variety of games fit this description so that a formalism convenient to describe a game might misfit some others. Modeling games as POSGs often naturally follows from describing the evolution of a game through generic rules (*e.g.,* "the probability that a player observes the true position of her opponent in a $n \times k$ grid is p and a false position with probability $1 - p$"). On the other hand, one can define a game in extensive form through "unfolding" a POSG or by describing a game through enumerating all the possible evolutions of the game. For some games (*e.g.,* Kuhn's poker (Vojtěch Kovařík et al. 2022a, Example 2.2)) which contain multiple rules specific to some evolutions of the game, EFGs appear more adapted. On the contrary, games with intrinsic structure (*e.g.,* meeting in a grid) are often more efficiently described by POSGs. It is noteworthy that linear programming and regret minimization were historically introduced for EFGs while dynamic programming approaches for games with imperfect information were developed for POSGs (Hansen et al. 2004; Horák et al. 2017).

The next two sections provide further details for EFGs and POSGs that will permit discussing further their respective advantages and drawbacks.

[7] https://en.wikipedia.org/wiki/Optimal_substructure

[8] For example, computing a^{15} for $a \in \mathbb{R}$ with as few multiplications as possible does not satisfy this property. Indeed, the subproblem a^6 admits the minimal expression $(a^2)^3$ requiring 3 multiplications while it would be more efficient to compute it as $(a^3)^2$ to store the value of a^3 and compute a^9 as $(a^3)^3$.

1.3.1 Extensive Form Games

We assume that players have perfect recall of what they previously saw and did in the game. EFGs are a very generic and widely used representation of sequential games. In their most general definition, they are able to model a large part of real-life scenarios. For example, the literature tackling the challenging task of optimally planning in poker games efficiently describes them through EFGs.

An EFG can naturally be derived from the description of a game through a finite tree of its possible evolutions, in which nature is considered as a player and its strategy encodes the possible stochasticity of the game. Nature acts by randomly taking actions according to a fixed probability distribution for each of "its" nodes of the tree. A zero-sum extensive form game is defined by:

- a rooted tree;
- a payoff value for each leaf of the tree;
- a partition of non-terminal nodes, one set per player including nature, to indicate for each node which player acts;
- probability distributions of nature's behavior;
- for each player, a partition of her acting nodes which groups them in *information sets*.

At each node of the game tree, exactly one player acts, which dictates the evolution of the game to a child node. A node thus corresponds to sequence of actions for all players. The formalism captures imperfect information by specifying what players know when playing. Whenever a player has to take an action, she is in one specific information set that groups together nodes that the player can not distinguish at execution time, and hence, must play the same actions at all those nodes.

1.3.2 Partially Observable Stochastic Games

The POSG framework extends both (i) single-player Markov decision processes with imperfect information by considering any number of agents rather than one, and (ii) stochastic games by introducing imperfect information. A POSG describes the interaction between any number of players and an environment, whose current state is hidden. The game is divided in multiple time steps, at which all players act, making the system evolve stochastically. The new state of the game is still hidden to players, but they receive partial and noisy observations, with probabilities that depend on players' actions and the state reached. Upon taking actions, players also receive reward, that classically depend on the current state of the game and on players' actions. A sequence of actions and observations of a player is called a private history. The imperfect information aspect of the game implies that players are unable to tell (i) what the current state of the game is and (ii) what their opponents' current history of actions and observations is.

We below try to highlight the conceptual differences between EFGs and POSGs, along with their respective advantages.

1.3.3 Why Do We Care About POSGs?

The last two sections introduced two different frameworks, namely POSGs and EFGs, which appear rather close in terms of expressivity, while they adopt two different and complementary viewpoints. Mainly, a POSG description of a game can be viewed as a representation of its rules while the same game can be described by a EFG by enumerating the possible evolutions of the game, *i.e.,* "unrolling" the POSG (Vojtěch Kovařík et al. 2022a). Both describe games with stochastic dynamics and imperfectly informed players that take actions and obtain returns. Besides, information sets appear similar to local histories. The discussion below discusses to what extent their apparent technical differences (*e.g.,* rewards being obtained at the end of the game *v.s.* each time players act; players acting simultaneously or not) are not fundamental.

Firstly, both notions of actions are the same. Also, nodes in EFGs are characterized by an environment state and the players' knowledge about the game. Informations sets gathering indistinguishable nodes from a player's point of view consequently correspond to histories of actions and observations in POSGs.

Secondly, players act simultaneously in a POSG, but one after another in a EFG. Turning a serialized game into a simultaneous one is the simpler case, since it only involves adding fictitious *noop* actions. Conversely, one can buffer, while maintaining them hidden, the actions made by players until everyone has played, and only then make the nature move.

Finally, rewards are typically obtained at each time step in a POSG while only at the end in an EFG. Again, one can simply aggregate the rewards obtained alongside a trajectory in a POSG game tree to form the unique reward given at the end in an EFG description of the POSG. Conversely, null fictitious rewards can be given to players at each time step.

As a matter of fact, Vojtěch Kovařík et al. (2022) proved that timeable[9] EFGs with perfect recall are equivalent to factored observation stochastic games (FOSGs), and conversely. FOSGs slighly modify POSGs by allowing the game to provide players with public observations (*i.e.,* all players know (i) their opponent's observations, (ii) that their opponent also knows it and (iii) that they know that the other player knows it etc. . .). The observation space is thus factorized as $\mathcal{Z} = \mathcal{Z}^{pub} \times \mathcal{Z}^{priv}$. Doing so renders some structure in the game explicit.

POSGs and EFGs thus do not differ in terms of expressivity, so that we will now investigate their differences for practical uses. More specifically, the following section discusses to what extent POSGs might be easier to work with for games with underlying structure that allow very compact representations.

1.3.3.1 Different Game Descriptions Offer Complementary Benefits

The underlying structure in the game (*e.g.,* publicly available information, additional observability assumptions) is kept implicit in its EFG description, while being key to a significant part of recent search algorithms (Brown et al. 2020; Schmid et al. 2021; Moravčík et al. 2017; Dibangoye et al. 2014b). Retrieving the underlying structure of a game described by a perfect-recall and timeable EFG might be possible, so that points mentioned below are to be understood as a characterization of the convenience of a framework, not as a description of its theoretical properties.

Overall, POSGs tend to factorize information by describing games through rules, whereas EFGs often duplicate it as pieces of information are often used to define multiple different parts of their game tree.

While, in EFGs, the nature's behavior can be completely different for two different nodes, the dynamics of any POSG is represented by probability matrices. In contrast with EFGs, nature in POSGs will behave according to the same transition matrices for any joint history h or $\tilde{h}$. In some real-world applications (Dibangoye et al. 2016), two different histories h and $\tilde{h}$ often induce the same normalized distribution over hidden states. This means that the probability the game has to be in any hidden state s is the same, given h or $\tilde{h}$. Then, acting optimally starting from h on is exactly the same as acting optimally starting from $\tilde{h}$ on. Consequently, both histories can be considered as equivalent and thus can be merged into a single one. The ability to thereby take advantage of real-world structure is key to a state-of-the-art algorithm tackling common-payoff POSGs introduced by Dibangoye et al. (2016). The procedure used to analyze possible compressions is completely game-independent.

Interestingly, POSGs also allow lossy compression of "nearly" equivalent histories (Dibangoye et al. 2014a) while bounding the loss. On the contrary, lossy abstractions typically used to reduce the dimensionality of EFGs are game-dependent and are made by hand (Brown et al. 2018) (though automated procedures for games with enough structure exist (Gilpin et al. 2007)).

[9] A EFG is timeable if there is a function from the set of nodes to the set of possible time (let us say, $\mathbb{N}$) which ensures that any node will have a greater image than any one of any of its parents.

1.4 Tackling zs-POSGs through Dynamic Programming

Section 1.2 introduced three state-of-the-art approaches to tackle zs-POSGs or subclasses. Applying dynamic programming to the general case of zs-POSGs still remains an open question, and we below investigate the benefits this approach could bring to this difficult task along with the main issues that have been preventing from doing so.

While infinite-horizon partially observable Markov decision processes (POMDPs) (Åström 1965) are undecidable (Madani 2010) and finite-horizon ones are PSPACE-complete (Papadimitriou et al. 1987), algorithms finding approximate solutions in reasonable time were discovered throughout the years (Williams 1992; Smith et al. 2005; Pineau et al. 2003; Kurniawati et al. 2008; Spaan et al. 2005). POMDPs have thus been widely used to represent many real-life problems.

One of the main techniques to tackle those problems relies on recasting the POMDP into an equivalent fully observable MDP and applying Bellman's optimality principle to iteratively construct approximations (*e.g.,* lower-bounding envelopes of hyperplanes and upper-bounding convex hulls) of the optimal value function (Pineau et al. 2003; Smith et al. 2005). This was later generalized to decentralized partially observable Markov decision processes through the introduction of a central planner reasoning upon *occupancy states* that summarize players' past strategies (Oliehoek et al. 2013; Dibangoye et al. 2013a). The ability to generalize values from visited subproblems to unvisited ones, efficient initializations of approximations through close relaxations, pruning techniques, knowledge compression, and point-based backups instead of full exhaustive ones were among the main ingredients to solve classical benchmarks from the literature (Dibangoye et al. 2013a). Recently, those approaches were generalized to tackle zero-sum stochastic games with one-sided observability (Horák et al. 2017) or public observability (Horák et al. 2019b) and solve real-life security problems efficiently.

1.4.1 The Challenges with Dynamic Programming for Imperfect Information Games

Generalizing Shapley's dynamic programming[10] approach for zero-sum stochastic games (Shapley 1953) to zero-sum partially observable stochastic games still remains an open question. In particular, one needs to properly characterize what a subproblem is in such games. From one player's point of view, subproblems are defined by her previous private strategy. Unfortunately, private strategies are not enough to infer the game's probability to be in each of its possible states, while this information is necessary to act optimally.

One possibility is for each player to make assumptions on how other players reason and to maintain a belief over 1. the current state of the game, and 2. the other players' internal states (*e.g.,* AOHs or beliefs). This may induce recursively defined beliefs, as in *Interactive POMDPs* (Gmytrasiewicz et al. 2005) or similar approaches (MacDermed 2013; Vojtěch Kovařík et al. 2022a). Such a convoluted reasoning requires making some assumptions (*e.g.,* on how other players make decisions and on the depth of the recursion). Also, it is not clear how to derive a principled method to compute a Nash equilibrium in this setting.

Instead, one could draw from the existing literature (Dibangoye et al. 2016) and define subproblems by the knowledge of both players' past strategies. Unfortunately, it raises an important question for zs-POSGs: how are such subproblems related to the original zs-POSG as they rely on information not available to players during execution?

Furthermore, the dynamic programming scheme mentioned above (which uses a sufficient statistic and approximates the optimal value function) offers a key lever: the capability to generalize knowledge between subgames, relying on continuity properties of the optimal value function. Unfortunately, while Wiggers et al. (2016) showed convex-concave properties of the optimal value function for some dimensions of a natural statistics used before for Dec-POMDPs (*i.e.,* occupancy states), they failed to derive approximations generalizing throughout the whole occupancy-state space, which is required by algorithmic schemes such as HSVI. A statistic offering both (i) conciseness and (ii) strong continuity properties has not been discovered yet for zs-POSGs.

[10] Shapley applied to zero-sum stochastic games similar principles to the ones used by Bellman to tackle Markov decision processes.

The main goal of this manuscript is to provide a positive answer to the issues mentioned above that arise when trying to generalize Shapley's dynamic programming approach for zs-SGs to zs-POSGs. Besides, our work on this subject inspired us for two other connected but independent contributions, which we present in other chapters.

1.5 Research Outline and Contributions

The contributions of this thesis are organized through essentially three parts. The first one tackles the problem of optimally planning in zs-POSGs, implementing a heuristic search value iteration scheme. The second one addresses the scalability issues in general common-payoff POSGs by exhibiting a relevant subclass for real-life applications. Finally, the third one studies one-shot games for which we relax continuity properties of payoff functions.

1.5.1 Planning in zs-POSGs

Chapter 3 shows that previous work on solving subclasses of zs-POSGs through the introduction of a non-observable game can be generalized to the whole class of zs-POSGs as well. This essentially requires (i) showing that the problem of computing a Nash equilibrium in the non-observable game possesses optimal substructure (*i.e.,* Bellman's optimality principle applies), (ii) discovering continuity properties in the non-observable game, and (iii) very carefully studying the translation of a Nash equilibrium of the non-observable game into a Nash equilibrium of the original game. This contribution led to a workshop article in MSDM 2023 as part of the AAMAS 2023 conference (Aurélien Delage et al. 2023b), and to a publication to the journal Dynamic Games and Applications (Aurélien Delage et al. 2023).

1.5.2 Planning in Common-Payoff POSGs under Hierarchical Information Sharing

Chapter 4 studies n-player common-payoff POSGs assuming a linear hierarchy in players' knowledge about the game. Every player knows what her subordinate knows and so forth. It results in a complexity drop for Bellman's backup operators, which allows adapting the PBVI algorithm to show empirically improved results compared to state-of-the-art approaches. This work has been published as a preprint (Peralez et al. 2024).

1.5.3 General Reward Model

Chapter 5 considers reward models with weak continuity properties with respect to players' strategies. We show that one can extend a state-of-the-art algorithm for single-variable optimization of Lipschitz functions to tackle two-player games whose payoff functions possess weak continuity properties and for which the players' set of actions can depend on the other's behavior. Our findings led to a workshop article in GAIW 2023 as part of the AAMAS 2023 conference, and an article at the ICTAI 2023 conference (Aurélien Delage et al. 2023a) (Best paper).

2

Background

Contents

*Some elements of this background (*e.g.*, the approach used to introduce readers to (partially observable) Markov decision processes, zero-sum (stochastic) games, their solution concepts and discussions about performance criteria) are inspired by Garcia et al.'s (2008) textbook. The following background section introducing game-theoretical concepts and (zero-sum) stochastic games of the book is itself inspired by Haurie et al.'s (2012) textbook.*

Game theory allows modeling very general interactive situations, so that there exists a wide range of formalisms. They all come with different assumptions, which translate into various difficulties in the search for optimal ways to act. Restricting ourselves to 2-player zero-sum games, we below provide elementary models, definitions, and properties relevant to understand both our contributions to optimally planning in `zs-POSG`s and the corresponding related work. Whenever pertinent, we provide some results for general-sum games. Besides, several definitions are given for any number n of players[11]. This chapter is organized in two sections. The first one provides definitions for various types of games along with elementary results. These results will be helpful for understanding the next section, which addresses the resolution of the games.

2.1 Overview on Various Subclasses of POSGs

After having detailed the vocabulary characterizing players' ways to act in games that we will use throughout this section, we present different formalisms for games. A classification highlighting the inclusion relations between them is given in Figure 2.1, depending on (i) how many decision points are involved, (ii) players having perfect or imperfect information, and (iii) the number of players (here one or two). For example, Bayesian games (`BG`s) generalize normal-form games (`NFG`s) by introducing about the state of the game, and stochastic games (`SG`s) generalize `NFG`s by allowing the game to be divided into multiple time steps while `NFG`s only contain one. Gathering both generalization axes gives partially observable stochastic games (`POSG`s). Similarly, partially observable Markov decision processes (`POMDP`s) generalize Markov decision processes (`MDP`s) by introducing uncertainty regarding the current state of the system.

[11] We are especially interested in $n = 2$, but made the editorial choice to stick to the usual definitions.

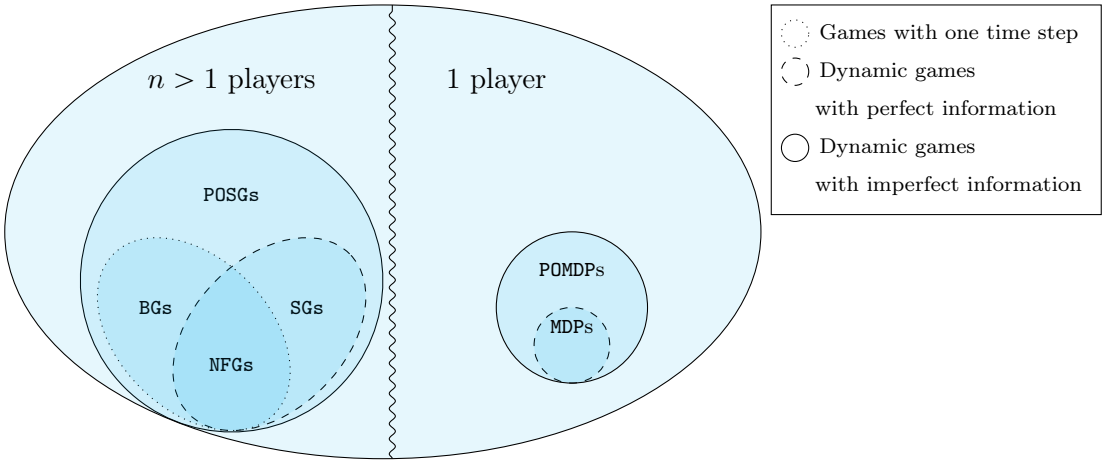


Figure 2.1: A Venn diagram representing the inclusion relations between several formalisms introduced in the background section.

Table 2.1: Vocabulary and notations used to describe players' behaviors.

One-shot games			
	notation	definition	note
action of i	a^i	$a^i \in \mathcal{A}^i$	
pure strategy of i	π^i	$\pi^i \in \mathcal{A}^i$	
mixed strategy of i	μ^i	$\mu^i \in \Delta(\mathcal{A}^i)$	
pure decision rule of i	d^i	$d^i : \Theta^i \to \mathcal{A}^i$	Θ^i: i's set of decision points
(behavioral) decision rule of i	β^i	$\beta^i : \Theta^i \to \Delta(\mathcal{A}^i)$	Θ^i: i's set of decision points
Dynamic games			
action of i	a^i	$a^i \in \mathcal{A}^i$	
decision rule of i	β^i_t	$\beta^i_t : \Theta_t \to \Delta(\mathcal{A}^i)$	
pure strategy of i	$\pi^i_{0:H-1}$	$\pi^i_{0:H-1} : \cup_t \Theta^i_t \to \mathcal{A}^i$	$\cup_t \Theta^i_t$: set of decision points
mixed strategy of i	μ^i	$\mu^i \in M^i_{0:H-1}$	$M^i_{0:H-1}$: set of mixed strategies
behavioral strategy of i	$\beta^i_{t:t'}$	$\beta_{t:t'} = (\beta^i_t, \dots, \beta^i_{t'})$	$t \leqslant t'$; usually $t = 0$, $t' = H-1$

2.1.1 Zoo of Behavior Descriptions

The next subsections introduce various game settings, each of them coming with its specific vocabulary[12] commonly used to describe player strategies. Brief presentations of concepts below are complemented with formal definitions appearing later in the chapter and are summarized in Table 2.1.

Regarding games involving only one time step, we shall distinguish between *actions* a and *strategies*. Actions (also named *pure strategies*) are used to refer to elementary interactions a player has with its environment. Strategies, on the contrary, encompass procedures that can be used by players to select actions. For example, commonly used strategies are probability distributions over actions; and playing according to a strategy means for example rolling a dice and playing an action according to the result. The latter type of strategy will be referred to as *a mixed strategy*.

Games involving several time steps lead to more complex and more varied descriptions for possible player behaviors. These descriptions typically rely on *decision rules* β_τ which are, for each time step τ, mappings from all possible reachable decision points to probability distributions over the finite set of actions. Collections of decision rules for multiple time steps τ to τ' ($\tau \leqslant \tau'$) are referred to as *behavioral strategies* $\beta_{\tau:\tau'}$. *Pure strategies* $\pi_{0:\tau'}$ are behavioral strategies always starting at time step 0, and for which each conditional probability distribution is deterministic (*i.e.,* is an action). Pure strategies can be viewed as a decision tree. Probability distributions μ over the set of all possible pure strategies are called *mixed strategies*. Discussing the equivalence between behavioral and mixed strategies, as well as their respective benefices, is deferred to

[12]Note that, sometimes, we modify the classical vocabulary to remain consistent.

Section 2.2.2.2 (page 29), once sufficient background has been provided.

The term *profile* will be used to refer to tuples of strategies for all players in the game. Also, if i denotes a player, $\neg i$ corresponds to all her opponents. For example, if there are n players, $a^{\neg i}$ is the tuple $(a^1, \dots, a^{i-1}, a^{i+1}, \dots, a^n)$. Finally, tuples are often summarized as a single bold variable, *e.g.,* $\boldsymbol{a}$ refers to the action profile $(a^1, \dots, a^n)$.

2.1.2 One-shot Games

Consider a scenario in which n players compete within an environment that rewards them based on their behavior. Such situations can be formalized as *one-shot games*.

Definition 2.1.1 (One-shot Game (OSG))**.** *A* one-shot game *is defined by a tuple* $\langle n, \times_{i=1}^{n} \mathcal{A}^i, f_1, \dots, f_n \rangle$*, where:*

- n *is the number of players,*
- *for any player* i*,* $\mathcal{A}^i$ *is her set of* actions *(which can be continuous),*
- *for any player* i*,* $f_i : \times_{i=1}^{n} \mathcal{A}^i \to \mathbb{R}$ *is her* payoff function.

A tuple $(a^1, \dots, a^n) \in \times_i \mathcal{A}^i$ *is called an* action profile.

One-shot games can be placed in the Venn diagram at the same place as NFGs, as they only generalize the latter ones by allowing continuous sets of actions, which is not an axis represented in the diagram.

Example 2.1.2 (One-Shot Game)**.** *To illustrate this formalism, let us consider the game (Madsen 2013) in which two players pick a real number in* $[0, 1]$*, which is their payoff, unless both players chose* 1*. In the latter case, both players' payoff is* 0*. This game can be described by an OSG* $G \stackrel{\text{def}}{=} \langle 2, [0,1], [0,1], f_1, f_2 \rangle$*, where* $\forall (x, y), f_1(x, y) = x \cdot \mathbb{1}_{\{xy<1\}}$ *and* $f_2(x, y) = y \cdot \mathbb{1}_{\{xy<1\}}$*.*

Since the formalized game stops after its unique time step, player i searches for an action that maximizes f_i, with no extra considerations for the game evolving in some possibly unpleasant states. Still, f_i depends on the other players' actions, so that i must take into account other players' behaviors. This raises the issue of defining a proper solution concept.

To free herself from the dependence on her opponents' strategies, a player could, at least when the game satisfies mild properties, (de Wolf 1999) search for the strategy that has the higher value, whatever her opponents do.

Definition 2.1.3 (Security Levels)**.** *Let* $G = \langle n, \times_i \mathcal{A}^i, f_1, \dots, f_n \rangle$ *be an OSG, such that sets* $\mathcal{A}^i$ *are compact and payoff functions* f^i *are continuous. Player* i*'s* security level *for* G *is:*

$$\max_{a^i \in \mathcal{A}^i} \min_{a^{\neg i}} f_i(a^1, \dots, a^i, \dots, a^n). \tag{2.1}$$

Any action $a^i \in \arg\max_{a^i \in \mathcal{A}^i} \min_{a^{\neg i}} f_i(a^1, \dots, a^i, \dots, a^n)$ *is a* security strategy*.*

Still, in general, all players playing according to a security strategy yields suboptimal behaviors. Instead, Nash introduced in 1950 one possible solution concept at the center of many studies in game theory, namely *Nash equilibria*.

2.1.2.1 Nash Equilibria: Definition and Existence Theorem

In essence, a Nash equilibrium strategy profile (NES) is an action[13] profile, known to all of the participants, in which unilateral deviations are not beneficial to any of the players. Such actions are fixed point in the reasoning *"but what if my opponent plays A? Then I would change my action; then my opponent would switch to B . . . "*. These equilibria are key to understanding the dynamics of decision making in strategic interactions.

[13] We use "action" here instead of "strategy", but depending on the context, Nash equilibria are also often described by strategies (see Theorem 2.1.5 and definition 2.1.6).

		Player Y: $Head$	Player Y: $Tail$
Player X	$Head$	$(1,-1)$	$(-1,1)$
	$Tail$	$(-1,1)$	$(1,-1)$

Figure 2.2: Matching pennies game

Formally, for any one-shot game G, a Nash equilibrium strategy profile of G in pure strategies is an action profile $\boldsymbol{a}^* = (a^{1,*}, \dots, a^{n,*})$ such that:

$$\forall i,\ \forall a^i,\ f_i(a^{1,*}, \dots, a^i, \dots, a^{n,*}) \leqslant f_i(\boldsymbol{a}^*). \tag{2.2}$$

For any player i, the problem of computing $\max_{\tilde{a}^i} f_i(a^1, \dots, \tilde{a}^i, \dots, a^{n,*})$ given her opponent's actions $a^{\neg i}$ is called computing a *best response to* $a^{\neg i}$. A Nash equilibrium strategy profile thus corresponds to an action profile in which each action of player i is a best response to her opponent's strategy. Still, such stable points do not necessarily exist in general, as illustrated in the *matching pennies* game (Example 2.1.4). One can also verify that the one-shot game introduced in Example 2.1.2 does not admit any Nash equilibrium.

Example 2.1.4 (Matching pennies)**.** *Matching pennies is a well-known zero-sum one-shot game ($n = 2, \mathcal{A}^1 = \mathcal{A}^2 = \{Head, Tail\}, f_2 = -f_1$) in which each player has a penny and secretly chooses one side (head or tail). Then, both penny's sides are revealed, and player 1 wins (payoff $+1$) if both chosen sides match, and loses (payoff -1) if not. The game matrix for this game, which defines f_1, is given in Figure 2.2.*

One can check using Figure 2.2 that, in any cell, one player has incentive to change her action. For example, if players play $(Head, Tail)$ (payoff -1 for player 1), then 1 would change to play Tail and receive $+1$ payoff. Similar observations hold for all other action profiles.

However, Nash's theorem (1950) shows that such an equilibrium always exists (but may not be unique) if sets $\mathcal{A}^i$ are finite, at the cost of considering "randomized actions", *i.e.,* mixed strategies.

Theorem 2.1.5 (Extension to Mixed Strategies (Nash 1950))**.** *If $\forall i \in \{1, \dots, n\}$, $\mathcal{A}^i$ is finite, then the one-shot game $G' = \langle n, \times_{i=1}^n \Delta(\mathcal{A}^i), (\tilde{f}_1, \dots, \tilde{f}_n)\rangle$, where*

- *$\Delta(\mathcal{A}^i)$ denotes the set of distributions over $\mathcal{A}^i$, and*
- *$\forall \boldsymbol{\mu} \in \times_{i=1}^n \Delta(\mathcal{A}^i),\ \tilde{f}_i(\mu) = \mathbb{E}_{\boldsymbol{a}\sim\boldsymbol{\mu}}[f_i(\boldsymbol{a})]$,*

always admits at least one Nash equilibrium.

Definition 2.1.6 (Vocabulary for the description of players' decision making)**.** *Given a one-shot game G, elements μ^i of $\Delta(\mathcal{A}^i)$ are called* mixed strategies *and strategies providing a probability 1 (*i.e., *a vertex of the simplex $\Delta(\mathcal{A}^i)$) to an action are called* pure strategies*, or equivalently* actions*, depending on the context (referring to a vertex of $\Delta(\mathcal{A}^i)$ or an element of $\mathcal{A}^i$).*

Remark 2.1.7. *In the game G', element $\mu^i \in \Delta(\mathcal{A}^i)$ are called actions. Still, we refer to them as mixed strategies, taking the viewpoint of the game G in which μ^i corresponds to procedures "roll a dice to pick the action". Unless explicitly forbidding the use of mixed strategies, games G' extended to mixed strategies are often assimilated to the original game G.*

Also, games G' introduced in Theorem 2.1.5 are normal-form games*, which we define later in Section 2.1.2.2.*

For example, the only[14] Nash equilibrium strategy profile for the matching pennies game is to play Head or Tail with equal probability 0.5, for both players.

If sets $\mathcal{A}^i$ of a one-shot game are infinite, the game is called an *infinite game* (Maitra et al. 1970) and without additional assumptions (Sion 1958) on (i) the convexity of the action space

[14]This game only admits a unique Nash equilibrium strategy profile.

and (ii) the convex-concavity of functions f_i, Nash Equilibria do not necessarily exist (Daskalakis 2022).

It is noteworthy that Theorem 2.1.5 does not come with a uniqueness result. Besides, Nash equilibria are only defined with respect to player's deviations from it but nothing is known about their respective value.

As a matter of fact, in general (*i.e.,* for games involving any number of players and not common-payoff ($f_1 = \cdots = f_n$) or zero-sum ($n = 2$ and $f_1 + f_2 = 0$)), there typically exist multiple Nash equilibria with different values and selecting one is a non-trivial problem, subject to research (*e.g.,* Harsanyi et al. (1988) propose a criterion based on payoff dominance and risk dominance to select a Nash equilibrium). A game is said to be *solvable* if all Nash equilibria are *interchangeable*, meaning that players can adopt strategies from *different* Nash equilibria while ensuring that the resulting strategy profile remains a Nash equilibrium strategy profile. There exist, however, games for which players can search for an *individual Nash equilibrium* on their own, in which case players do not need to agree beforehand on one specific equilibrium.

Remark 2.1.8. *An idea to search for a Nash equilibrium strategy profile would be to make a player i best-respond to a given (pure) strategy of players $\neg i$, update i's strategy, then make another player j best respond to the best response, and so on. The resulting algorithmic scheme is called* best-response dynamics[15] *(Amiet et al. 2021) and stops whenever the algorithm runs out of time budget or whenever no player changes her pure strategy by responding to the others' one. In that case, the strategy profile is necessarily a Nash equilibrium. In the common-payoff setting, and assuming a sufficient time budget, the algorithm is guaranteed to converge to a Nash equilibrium, but not necessarily to the one with highest value.*

In the following, we will focus on games that involve only two players and have the property that the gain of one player equals the loss of the other player, expressed mathematically as $f_1 = -f_2$. Such games are known as *zero-sum games*. Both players having exactly opposite interests will highly simplify the search for Nash equilibrium strategy profiles as, under certain continuity properties for the payoff function, the game will be solvable.

2.1.2.2 Zero-Sum Normal-Form Games

Zero-sum normal-form games are zero-sum one-shot games with only two players and finite sets of actions. They are among the main building blocks of game theory, often used to provide baselines or complexity results by turning complex, and possibly dynamic, games into their equivalent *normal form*.

Definition 2.1.9 (Zero-Sum NFG (von Neumann 1928))**.** *A zero-sum normal-form game*[16] *is defined by a tuple $\langle \mathcal{A}^1, \mathcal{A}^2, M \rangle$ where*

- *$\mathcal{A}^1$ and $\mathcal{A}^2$ are finite sets of respective cardinal p and m,*
- *$M \in \mathcal{M}_{p,m}(\mathbb{R})$.*

The payoff associated to an action profile (a^1_i, a^2_j), where $(i, j) \in \{1, \ldots, p\} \times \{1, \ldots, n\}$, is $M_{i,j}$.

Note that a normal-form game $\langle \mathcal{A}^1, \mathcal{A}^2, M \rangle$ is a two-player one-shot game in which f^1 is induced by M, $f_2 = -f_1$ and $\forall i, \mathcal{A}^i$ is finite, so that Theorem 2.1.5 applies. In coherence with Definition 2.1.6, in a normal-form game, we call mixed strategy for player 1 (resp. player 2) an element μ^1 of $\Delta(\mathcal{A}^1)$ (resp. μ^2 of $\Delta(\mathcal{A}^2)$) and the value of a mixed strategy profile (μ^1, μ^2) is $\mu^{1,\top} \cdot M \cdot \mu^2$.

Example 2.1.10 (Matching pennies as an NFG)**.** *The matching pennies game can be described by an NFG $\langle \{Head, Tail\}, \{Head, Tail\}, M \rangle$, where M is the matrix given in Figure 2.2.*

[15] For cp-POSGs (to be defined later), the implementation of this algorithmic scheme is called JESP (Nair et al. 2003).

[16] We voluntarily restrict the number of players and make the zero-sum hypothesis in this definition for simplicity, even though general normal-form games are defined for any number of players and for general-sum.

2.1.2.3 Minimax Theorem

In zs-NFGs, the well-known *minimax theorem* (von Neumann 1928) provides a very useful characterization of Nash equilibria.

Theorem 2.1.11 (Minimax theorem (von Neumann 1928))**.** *Let* $\langle \mathcal{A}^1, \mathcal{A}^2, M \rangle$ *be an* NFG*. Then,*

$$\max_{\mu^1 \in \Delta(\mathcal{A}^1)} \min_{\mu^2 \in \Delta(\mathcal{A}^2)} \mu^{1,\top} \cdot M \cdot \mu^2 = \min_{\mu^2 \in \Delta(\mathcal{A}^2)} \max_{\mu^1 \in \Delta(\mathcal{A}^1)} \mu^{1,\top} \cdot M \cdot \mu^2. \tag{2.3}$$

Besides,

- *for any* $\mu^1 \in \arg\max_{\mu^1 \in \Delta(\mathcal{A}^1)} \min_{\mu^2 \in \Delta(\mathcal{A}^2)} \mu^{1,\top} \cdot M \cdot \mu^2$*, and*
- *for any* $\mu^2 \in \arg\min_{\mu^2 \in \Delta(\mathcal{A}^2)} \max_{\mu^1 \in \Delta(\mathcal{A}^1)} \mu^{1,\top} \cdot M \cdot \mu^2$*,*

the pair (μ^1, μ^2) *is a Nash equilibrium strategy profile.*

Note that Theorem 2.1.11 implies that any zs-NFG is solvable, that finding an individual Nash equilibrium strategy is reduced to a bi-level optimization problem, and that all Nash equilibria have the same value.

Best Responses and Security Levels in Zero-Sum Normal-Form Games There always exist deterministic best responses in any zs-NFG, which can be found through simple enumeration over the player's actions. It also follows from Theorem 2.1.11 that security levels characterize Nash equilibria, as any security strategy is an individual Nash equilibrium strategy profile.

Example 2.1.12 (Application to matching pennies)**.** *For player* 1*, playing Head with probability* $p \neq 0.5$ *straightforwardly leads to* 2 *best-responding by (i) Tail if* $p > 0.5$ *and Head if* $p < 0.5$*, with negative expected payoff. However, if* $p = 0.5$*, every mixed strategy of* 2 *has a null expected outcome. The mixed strategy maximizing* 1*'s security level (which is also her*[17] *Nash equilibrium value) is thus playing Head or Tail with probability* 0.5*. Similar reasoning shows that* 2 *must also play Head or Tail with probability* 0.5*, with a null expected outcome. Consequently, playing uniformly at random for both players is both (i) their unique security strategy and (ii) the unique Nash equilibrium strategy profile of this game.*

Before presenting dynamic games, we first define Bayesian games, which introduce imperfect information.

2.1.2.4 Bayesian Games

When trying to tackle games with imperfect information such as POSGs, studying Bayesian games is an important step as they generalize normal-form games by introducing uncertainty about the state of the game. At the start of the game, all players are given a *type*, randomly chosen over a finite set of types, according to a predefined probability distribution p over players' type profiles. The payoff function depends on *all* players' types. Despite the other players' types remaining unknown to each other, the probability distribution used by nature to sample players' types is known to all.

Definition 2.1.13 (Bayesian Games (Harsanyi 1968))**.** *A Bayesian game is a tuple*

$$\langle n, \times_i \mathcal{A}^i, \times_i \Theta^i, p, f_1, \ldots, f_n \rangle,$$

where:

- n *is the number of players,*
- $\forall i \in \{1, \ldots, n\}$*,* $\mathcal{A}^i$ *is* i*'s set of* actions *(*$|\mathcal{A}^i| < \infty$*);*

[17] This game only admits one unique Nash equilibrium.

Table 2.2: Payoff functions for the Sheriff's dilemma game for both types of the person.

		Opponent: *Shoot*	Opponent: *Don't*
Sheriff	*Shoot*	$(0, 0)$	$(2, -2)$
	Don't	$(-2, -1)$	$(-1, 1)$

(a) Criminal type

		Opponent: *Shoot*	Opponent: *Don't*
Sheriff	*Shoot*	$(-3, -1)$	$(-1, -2)$
	Don't	$(-2, -1)$	$(0, 0)$

(b) Citizen type

- $\forall i \in \{1, \dots, n\}$, Θ^i *is i's set of possible* types *($|\Theta^i| < \infty$), and the type profiles* $\boldsymbol{\theta}$ *are elements of* $\Theta \stackrel{\text{def}}{=} \times_i \Theta^i$*;*
- $p \in \Delta(\Theta)$ *is the probability distribution over players' type profiles;*
- $\forall i \in \{1, \dots, n\}$, $f_i : \mathcal{A} \times \Theta \to \mathbb{R}$ *is i's payoff function.*

Players' behaviors can only vary depending on their own experienced type since their opponents' types remain unknown. As a consequence, players' decisions rules typically only take as input their own types θ^i.

Definition 2.1.14 (Vocabulary for the Description of Players' Decision Making)**.** *In such a game, a* decision rule *for i is a mapping* $\beta^i : \Theta^i \to \Delta(\mathcal{A}^i)$ *and the value for i of a* decision rule profile $\boldsymbol{\beta} = (\beta^i)_i$ *is* $\mathbb{E}_{\boldsymbol{a} \sim \boldsymbol{\beta}, \boldsymbol{\theta} \sim p}[f_i(\boldsymbol{a}, \boldsymbol{\theta})]$.

Example 2.1.15 (Sheriff's dilemma)**.** *The sheriff's dilemma is a game in which an armed sheriff faces an armed person. This person can be a criminal or a citizen, with respective probability* $p \in [0, 1]$ *and* $1 - p$*. Both the sheriff and the person have the choice of whether to shoot or not. Clearly, depending on the person's type, unknown to the sheriff, both players' payoff for shooting or not differs. The payoff functions for both types of the person are given in Table 2.2. While the sheriff only has a unique type, making the task easier for the person, the sheriff has to reason upon the expected payoff of her actions, depending on the probabilities over the person's type.*

Interestingly, von Neumann's minimax theorem still holds, and *the* optimal value function $p \mapsto V^*(p)$ of a zero-sum Bayesian game exhibits structure. The proposition below shows the concavity (resp. convexity) of V^* in the space of player 1 (resp. 2) *marginal distributions* that are derived from distributions p over type profiles.

Proposition 2.1.16 (Concavity w.r.t. marginal distributions (Harsanyi 1968))**.** *Let*

$$B = \langle 2, \times_{i=1}^{2} \mathcal{A}^i, \times_{i=1}^{2} \Theta^i, p, f_1, -f_1 \rangle$$

be a zero-sum Bayesian game. There exists an infinite collection $(\alpha_{\beta^2})_{\beta^2 \in \Delta(\mathcal{A}^2)}$ *such that* $\forall \beta^2$*,* $\alpha_{\beta^2} \in \mathbb{R}^{\Theta^1}$ *is a mapping and*

$$\min_{\beta^2} \max_{\beta^1} \mathbb{E}_{(a^1, a^2) \sim \boldsymbol{\beta}, \boldsymbol{\theta} \sim p} \left[f_1(a^1, a^2, \theta) \right] = \min_{\beta^2} \sum_{\theta^1 \in \theta^1} \Pr(\theta^1 \mid p) \alpha_{\beta^2}(\theta^1) \tag{2.4}$$

$$= \min_{\beta^2} \langle p^1, \alpha_{\beta^2} \rangle, \tag{2.5}$$

where $p^1 : \theta^1 \mapsto \sum_{\theta^2} p(\theta^1, \theta^2)$.

For a given β^2, the component of vector α_{β^2} attached to type θ^1 is the value of 1's best response to β^2 when 1's type is θ^1. The concavity holds due to the fact that each component of α_{β^2} can be computed independently.

2.1.3 Dynamic Games

In most real-life scenarios, games involve players at multiple time steps at which they must act, influencing the game's future. To capture such situations, game theory allows for considering *dynamic games*. They can consist in repeated games, such as playing matching pennies again and again; but also in games whose state (possibly stochastically) evolves with respect to players' actions. We are especially interested in the latter ones. The following sections first details fully and partially observable systems with only one player and then moves on to the multi-player case.

2.1.3.1 Overview on Evaluation Criteria

For now, the considered games involved only one time step, and the players have been searching for Nash equilibrium strategy profiles based on given strategy evaluation functions. However, in the context of dynamic games, the decisions made for all time steps in the game influence the total payoff. While the evaluation function for each state and time step of the game is part of the model, it is critical to also properly define one for the entire game, encompassing multiple (and possibly infinitely many) time steps. Below, we briefly introduce well-known criteria for games, while maintaining a high-level discussion.

In cases where the game stops after $H < \infty$ time steps, the expected finite sum of rewards is a natural and widely used criterion. It is referred to as the *finite criterion*. The sum of rewards can be discounted using a discount factor $\gamma \in [0, 1]$. For the sake of genericity, we always consider the finite criterion to be discounted, including the case $\gamma = 1$.

Furthermore, in certain real-life applications, players, having a finite lifetime, may be willing to accept a slight reduction in the expected sum of rewards if it also decreases the variance of the expectation. The *risk-sensitive criteria* (Markovitz 1959; Geibel 2001) are possible formalisms relevant to such situations.

When considering games played for infinitely many time steps, a discount factor $\gamma < 1$ is often given, to define the probability 1-γ that the game stops after each interaction. Conceptually, it reflects players being less confident about rewards obtained far away in the future. Besides, it ensures that the expected sum of discounted rewards (*i.e.,* step t's expected reward is multiplied by γ^t) remains finite under any possible behavior of players, which is of great technical help. The corresponding criterion is known as the *discounted criterion*.

Finally, when the horizon is infinite and whenever it makes no sense to prefer short-term rewards over long-term ones (*e.g.,* the effective horizon for optimization (which has to be finite in practice) is long compared to the decicions frequency), the *average criterion* is preferred to the discounted criterion. It evaluates the average reward obtained along trajectories.

2.1.3.2 Markov Decision Processes

Markov decision processes (MDPs) are among the simplest formalisms for single-player games. They describe dynamic systems that start at an initial state s_0, evolving according to dynamics induced by a transition function T until a time horizon H. At each time step $t \in \{0, \dots, H-1\}$, the player performs actions and receives rewards based on the current state, the action taken and the resulting next state.

Definition 2.1.17. *An MDP is defined by a tuple* $(\mathcal{S}, \mathcal{A}, T, r, \gamma, H, s_0)$ *where:*

- $\mathcal{S}$ *is the set of possible* states *for the system (*$|\mathcal{S}| < +\infty$*);*
- $\mathcal{A}$ *is the set of* actions *(*$|\mathcal{A}| < +\infty$*);*
- $T(s, a, s')$*, the* transition function*, gives the probability that the system moves from state* s *to* s' *upon taking action* a *(with* $\forall s, a, \sum_{s'} T(s, a, s') = 1$*);*
- r *is a* reward function $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \longmapsto \mathbb{R}$ *that associates an immediate payoff to each possible transition* (s, a, s')*;*

- $\gamma \in [0,1]$ *is a* discount factor*;*
- $H \in \mathbb{N} \cup \{\infty\}$ *is the* time horizon*;*
- s_0 *is the initial state.*

In the following, we rather use $\tilde{r} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ *such that* $\forall (s,a), \tilde{r}(s,a) \stackrel{\text{def}}{=} \sum_{s'} T(s,a,s')r(s,a,s')$ *and abuse notation to confound* $\tilde{r}$ *and* r*. Besides, we discard the case where* $H = \infty$ *and* $\gamma = 1$*, assuming that either* $H < \infty$ *or* $\gamma < 1$*.*

Strategies and optimization criteria We begin by stating the description of players' behavior of interest for this section.

Definition 2.1.18 (Vocabulary for the Description of Players' Decision Making)**.** *In coherence with the terminology introduced in Section 2.1.1, in the case where* H *is finite, a* pure strategy $\pi_{0:H-1}$ *in an* MDP *is a collection of mappings* $(\pi_t)_{t\in\{0,\dots,H-1\}}$*, for all time steps, taking as input* any[18] *state for the specific time step and yielding an action* $a \in \mathcal{A}$*.*

We assume that the player aims at maximizing the finite criterion $\mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t r(S_t, A_t) \mid s_0, (\pi_t)_t\right]$. The maximum exists as the set $(A^S)^H$ of all pure strategies is finite. This defines an *optimal pure strategy* of the player and the *optimal value* $V^*(s_0)$ for this criterion. Considering mixed strategies or behavioral strategies would not yield higher expected value (Puterman 2014).

Bellman's optimality equations Similarly to the optimal value for s_0 at time step 0, we define the optimal value at any time step t for any state s as:

$$V_t^*(s) \stackrel{\text{def}}{=} \max_{\pi_{t:H-1}} \mathbb{E}\left[\sum_{\tau=t}^{H-1} \gamma^{\tau-t} r(S_\tau, A_\tau) | S_t = s, \pi_{t:H-1}\right].$$

Assume that, at time step $H-1$, some state s was reached. Then, clearly, the player maximizes her expected reward by playing any action $\pi_{H-1}^*(s) \stackrel{\text{def}}{\in} \arg\max_{a\in\mathcal{A}} r(s,a)$.

Now, let us consider that some state s was reached at time step $H-2$. Knowing the optimal action $\pi_{H-1}^*(s')$ for each reachable state s' at time step $H-1$, the player can find the best trade-off, in expectancy, between maximizing the immediate reward and the value for time step $H-1$. More formally, the best way to act for the player at time step $H-2$ in state s is to play

$$\pi_{H-2}^*(s) \stackrel{\text{def}}{\in} \arg\max_a \left[r(s,a) + \gamma \sum_{s'} T(s,a,s') V_{H-1}^*(s')\right],$$
$$\text{where } V_{H-1}^*(s') \stackrel{\text{def}}{=} \max_a r(s',a) = r(s', \pi_{H-1}^*(s')).$$

We are starting to observe a nesting of the optimization problem of computing the best way to act for any time step t, which requires the most rewarding behavior for any reachable state at time step $t+1$, which is obtained through an optimizing process knowing the most rewarding behavior for any reachable state at time step $t+2$, and so on.

In fact, Bellman's optimality principle mathematically expresses this observation:

$$\forall t,\ \forall s,\ \max_{\pi_{t:H-1}} \mathbb{E}[\sum_{\tau=t}^{H-1} \gamma^{\tau-t} r(S_\tau, A_\tau) \mid s, \pi_{t:H-1}] = \max_a \left[r(s,a) + \gamma \sum_{s'} T(s,a,s') V_{t+1}^*(s')\right], \quad (2.6)$$

where $V_{t+1}^*(s') \stackrel{\text{def}}{=} \max_{\pi_{t+1:H-1}} \mathbb{E}[\sum_{\tau=t+1}^{H-1} \gamma^{\tau-(t+1)} r(S_\tau, A_\tau) \mid s, \pi_{t+1:H-1}]$. Note that computing $V_{t+1}^*(s')$ corresponds to a "subproblem" and solving all "subproblems" allows solving the original problem, *i.e.,* computing $\max_{\pi_{t:H-1}} \mathbb{E}[\sum_{\tau=t}^{H-1} \gamma^{\tau-t} r(S_\tau, A_\tau) \mid s, \pi_{t:H-1}]$.

[18] Note that only *reachable* states are important, but some algorithms (*e.g.,* backward induction) can not determine in advance the relevant states, while some others (*e.g.,* HSVI) take advantage of this observation to reduce the size of built strategies.

Infinite-horizon setting In the case where H is infinite, we consider the discounted criterion $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t) \mid \pi_{0:\infty}, s_0]$ for the process starting in s_0. Defining an optimal strategy is not as straightforward as for the finite criterion since there exist infinitely many pure strategies $\pi_{0:\infty}$.

Bellman's operator Fortunately, the optimal value function V^* is the *unique* fixed point of the γ-contracting Bellman operator $\mathcal{H}$ taking as input a value function V and yielding the value function $\mathcal{H}(V)$ defined as:

$$\forall s \in \mathcal{S}, \mathcal{H}(V)(s) = \max_a r(s,a) + \gamma \sum_{s'} T(s,a,s')V(s'). \tag{2.7}$$

It follows (Garcia et al. 2010) that a *pure decision rule* $d^* : \mathcal{S} \to \mathcal{A}$ obtained through greedy selection upon V^*, *i.e.,*

$$\forall s, d^*(s) \overset{\text{def}}{\in} \max_a \left[r(s,a) + \gamma \sum_{s'} T(s,a,s')V^*(s') \right],$$

is optimal. Its value $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{-t} r(S_t, A_t) \mid d^*, s_0\right]$ is as good as the value of any pure strategy $\pi_{0:\infty}$. Intuitively, this result might not appear completely surprising. For any state s reached at any time step t, there still exist infinitely many other time steps; therefore, there is no reason to play differently than if the same state were reached at another time step t'. One can consequently consider only *stationary strategies*, each corresponding to *a unique* pure decision rule.

Still, MDPs assume that the state of the game is always known to the player, limiting the use of this formalism when it comes to real-life situations.

2.1.3.3 Partially Observable Markov Decision Processes

We now consider that, at any time step, the current state of the game is unknown to the player, which only receives noisy observations informing her about the system's possible states. This leads to defining partially observable Markov decision processes (POMDPs), in which the player must in general take into account her uncertainty about the system state to act optimally for the specified criterion.

Definition 2.1.19. *A POMDP is defined by a tuple $\langle \mathcal{M}, \mathcal{Z}, O, b_0 \rangle$ where:*

- *$\mathcal{M} \equiv \langle \mathcal{S}, \mathcal{A}, T, r, H, \gamma \rangle$ are like in Definition 2.1.17;*
- *$\mathcal{Z}$ is the set of* observations *($|\mathcal{Z}| < +\infty$);*
- *$\mathcal{O}$ is the* observation function *and is such that for all z, s', $O(z,a,s')$ is the probability of receiving z when a is taken in some state, and the resulting state is s';*
- *$b_0 \in \Delta(\mathcal{S})$ is the* initial belief *over possible initial states.*

Again, we discard the case $H = \infty$ and $\gamma = 1$.

In a POMDP, the player does not observe the true state $s \in \mathcal{S}$ of the system. However, upon taking actions at each time step, she randomly receives an observation $z \in \mathcal{Z}$ from it. Probabilities to receive each observation depend on the last action and the state reached. Figure 2.3 provides an influence diagram describing the interactions of a player within a POMDP. Note that decisions (*i.e.,* actions) can depend on previous actions made and observations received.

Definition 2.1.20 (Action-observation histories)**.** *Sequences of actions and observations of length $t \geqslant 0$, $\theta_t \overset{\text{def}}{=} (a_0, z_1, \ldots, a_{t-1}, z_t) \in \Theta_t$ (for $t = 0$, $\theta_0 \overset{\text{def}}{=} \varnothing$), are called* action-observation histories *(AOH). The set of all histories is noted $\Theta \overset{\text{def}}{=} \cup_t \Theta_t$. If $H < \infty$, Θ is finite.*

Knowing the system's dynamics (b_0, T and $\mathcal{O}$) and her action-observation history, the player is able to infer a probability distribution over the possible current states of the system. The rewards depending on the system's state and the player's action, reasoning upon this distribution is mandatory to ensure optimality in general. For this reason, optimal strategies are history-dependent, providing the best actions to make in each reachable history.

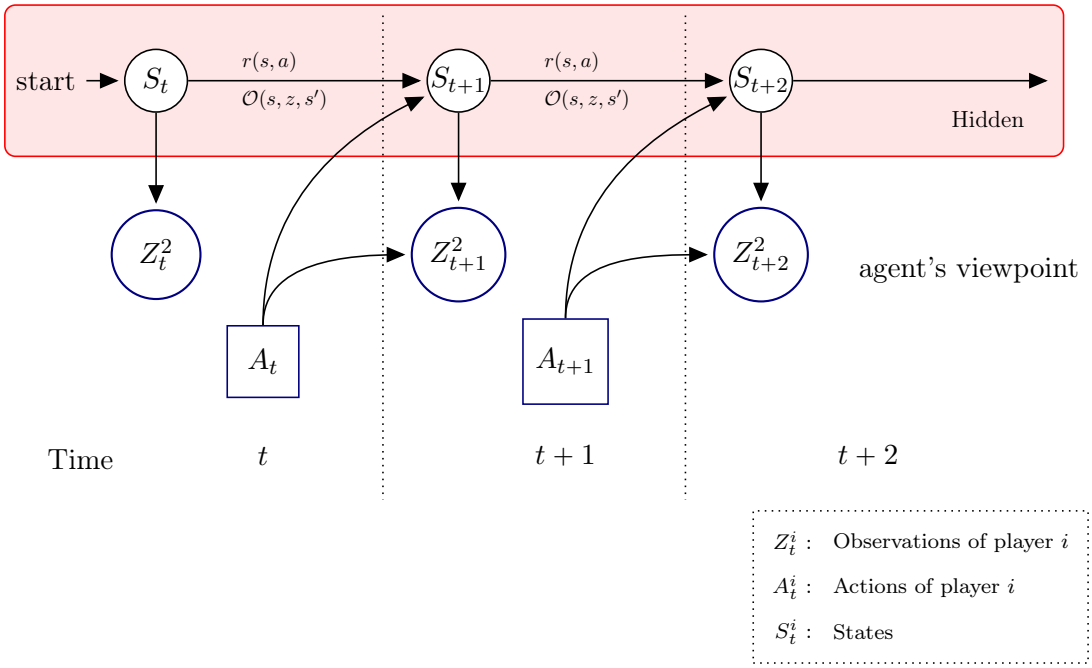


Figure 2.3: POMDP influence diagram

Example 2.1.21 (Rocksample (Smith et al. 2005))**.** *A classical benchmark used to test solving algorithms for POMDPs involves a rover exploring a planet. The rover's task is to collect valuable rocks, but it has imperfect knowledge about the value of each rock in the area. Each rock has a different scientific value, and the reward the rover receives for collecting a rock consequently depends on its value. Since sampling rocks costs energy, the rover is equipped with a sensor so that it can perform a Check$_i$ action to receive a noisy observation (the farther the rock, the noisier the observation) about the value of rock i. Finally, knowing a probability distribution b_0 of rocks value and after receiving any sequence of noisy observations about rocks value, the robot can infer a probability distribution about the value of all (the belief about non-sampled rocks is the one from b_0) rocks.*

Strategies and Optimization Criterion Here, we focus on the discounted finite criterion for finite-horizon games, and then move on to the discounted criterion for infinite-horizon games. The analysis for other criteria can be found in Garcia et al.'s (2008) textbook.

Definition 2.1.22 (Vocabulary for the description of players' decision making)**.** *In the case where a finite horizon $H \in \mathbb{N}$ is given, a* pure strategy $\pi_{0:H-1}$ *of the player is a collection $(\pi_t)_{t\in\{0,\dots,H-1\}}$ of mappings providing actions to make in each reachable history ($\forall t, \pi_t : \Theta_t \to \mathcal{A}$).*

The player aims at maximizing the finite criterion $\mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t r(S_t, A_t) \mid b_0, (\pi_t)_{t\in\{0,\dots,H-1\}}\right]$. The maximum exists as the set of pure strategies is finite so that the *optimal strategies* and the *optimal value* $V_0^*(\theta_0)$ for this criterion are correctly defined. As for MDPs, considering mixed strategies or behavioral strategies would not yield higher expected value (Åström 1965).

Bellman's Optimality Principle Similarly to MDPs, Bellman's optimality principle links the optimal value functions for each time step of a POMDP, considering the finite criterion. A significant difference, however, is that, in POMDPs, the player does not observe the current state of the game. As a consequence, she must, in general, remember everything she previously saw since any information in her AOH might be necessary to act optimally. Therefore, Bellman's optimality principle rather applies to the computation of optimal values of histories. The value of a history θ_t is $V_t^*(\theta_t) \stackrel{\text{def}}{=} \max_{\pi_{t:H-1}} \mathbb{E}\left[\sum_{\tau=t} \gamma^{\tau-t} r(S_t, A_t) \mid \theta_t, \pi_{t:H-1}\right]$. Bellman's optimality principle leads to showing that the computation of $V_t^*(\theta_t)$ is linked to the optimal values $V_{t+1}^*(\theta_{t+1})$ of reachable

histories θ_{t+1}, *i.e.,*

$$\forall t, \forall \theta_t, V_t^*(\theta_t) = \max_{a_t} \mathbb{E}[r(S_t, A_t) \mid \theta_t, A_t = a_t] + \gamma \sum_{z_{t+1}} \Pr(z_{t+1} \mid a_t, \theta_t) V_{t+1}^*(\theta_t \oplus \langle a_t, z_{t+1} \rangle), \tag{2.8}$$

where $\oplus$ denotes concatenation.

Infinite-Horizon Setting If the POMDP has infinite horizon, the number of pure strategies is infinite so that the discounted criterion does not straightforwardly admit a maximum. Besides, working with AOHs is difficult, especially since their number grows exponentially w.r.t. the time step. Instead, Section 2.2.3.3 will show that one can replace AOHs with a sufficient and Markovian statistic regarding optimal planning, namely *belief states*. The evolution of the latter statistic will describe a Markov decision process with a continuous state space (the set of all possible belief states) and an infinite horizon. This will unveil multiple new levers compared to working with AOHs (*e.g.,* the existence of stationary optimal policies and convexity properties of the optimal value function in the belief-state space).

We now leave the single-player setting to consider more players interacting within a stochastic environment, starting with the perfect information case.

2.1.3.4 Stochastic Games

In essence, stochastic games resemble Markov decision processes, with the difference that they involve an arbitrary number of players that take actions simultaneously. Additionally, each player has her own reward function. In the following, we focus on stochastic games with only two players.

Definition 2.1.23 (Two-Player Stochastic Game (Shapley 1953))**.** *A two-player stochastic game is a tuple* $\langle \mathcal{S}, \boldsymbol{\mathcal{A}} = \mathcal{A}^1 \times \mathcal{A}^2, T, \boldsymbol{r}, \gamma, s_0 \rangle$*, where:*

- $\mathcal{S}$ *is a finite set of* states*;*
- $\forall i,\ \mathcal{A}^i$ *is* i*'s finite set of* actions*;*
- $T : \mathcal{S} \times (\times_i \mathcal{A}^i) \times \mathcal{S}$ *is the* transition function*, giving the probability* $T(s, a^1, a^2, s')$ *to reach the state* s'*, starting from* s *if players play* a^1 *and* a^2*;*
- $\forall i,\ r^i : \mathcal{S} \times (\times_i \mathcal{A}^i) \to \mathbb{R}$ *is* i*'s* reward function*;*
- $\gamma \in [0, 1]$ *is a* discount factor*;*
- $H \in \mathbb{N} \cup \{\infty\}$ *is the* time horizon*;*
- s_0 *the* initial state *of the game.*

Again, we discard the case $H = \infty$ and $\gamma = 1$. Note that the actions of *both* players influence the dynamics of the game. Moreover, as in one-shot games, the rewards received by players depend on the others' actions. Consequently, to act optimally, i must take into account $\neg i$'s behavior, as for one-shot games, but also a trade off between the immediate reward for some actions and the game evolving in various possible (next) states.

In the following, we will assume that the game is zero-sum. Hence we note r a unique reward function, assuming that, by default, 1 tries to maximize its expected return and 2 tries to minimize it.

Definition 2.1.24 (Vocabulary for the Description of Players' Decision Making)**.** *In this definition, we assume that* $H < \infty$*. A* behavioral strategy *for* i *is a collection of mappings* $\beta^i_{0:H-1} \overset{\text{def}}{=} (\beta^i_t)_{t \in \{0,\dots,H-1\}}$*, each taking as input any state of the game at any time step and yielding an element of* $\Delta(\mathcal{A}^i)$*.*

Optimization Criterion If a finite horizon H is given, we assume that players' behavioral strategies are evaluated through the finite criterion $\mathbb{E}\left[\sum_{t=0}^{H-1}\gamma^t r(S_t, A_t) \mid s_0, \beta^1_{0:H-1}, \beta^2_{0:H-1}\right]$. Kuhn (1950) showed that this game can be turned into an equivalent normal-form game. In this NFG, players' "actions" are pure strategies ($\pi^i_{0:H-1}$), mapping any pair (s,t) to an action $a^i \in \mathcal{A}^i$ (for $t=0$, only the pair $(s_0,0)$ is relevant). Therefore, a Nash equilibrium strategy profile in mixed strategies $\boldsymbol{\mu} = (\mu^1, \mu^2)$ exists as solution of the NFG. μ^1 and μ^2 are probability distributions over pure strategies, *i.e.,* can be viewed as distributions over a finite number of decision trees. To execute μ^i, player i randomly samples a pure strategy (according to μ^i) and follows the decision tree until the end. Furthermore, while not mandatory, the mixed strategy profile $\boldsymbol{\mu}$ can be turned (relying on our assumption that games have perfect recall) into a behavioral strategy profile, which is also a Nash equilibrium strategy profile that achieves the minimax value.

Bellman's optimality principle The optimal value of any state s_t at time step t is

$$V^*_t(s_t) \stackrel{\text{def}}{=} \max_{\beta^1_{t:H-1}} \min_{\beta^2_{t:H-1}} \mathbb{E}\left[\sum_{\tau=t}^{H-1}\gamma^{\tau-t} r(S_\tau, A_\tau) \mid s_t, \beta^1_{t:H-1}, \beta^2_{t:H-1}\right].$$

Bellman's optimality principle applies, *i.e.,*

$$\forall s_t, \forall t < H-1, \max_{\beta^1_{t:H-1}} \min_{\beta^2_{t:H-1}} \mathbb{E}\left[\sum_{\tau=t}^{H-1}\gamma^{\tau-t} r(S_\tau, A_\tau) \mid s_t, \beta^1_{t:H-1}, \beta^2_{t:H-1}\right] \tag{2.9}$$

$$= \max_{\mu^1_t \in \Delta(\mathcal{A}^1)} \min_{\mu^2_t \in \Delta(\mathcal{A}^2)} \mathbb{E}[r(S_t, A_t) \mid \mu^1_t, \mu^2_t, s_t] \tag{2.10}$$

$$+ \gamma \max_{\beta^1_{t+1:H-1}} \min_{\beta^2_{t+1:H-1}} \sum_{s'} \Pr(s' \mid s_t, \mu^1_t, \mu^2_t) \mathbb{E}\left[\sum_{\tau=t+1}^{H-1}\gamma^{\tau-(t+1)} r(S_\tau, A_\tau) \mid \beta^1_{t+1:H-1}, \beta^2_{t+1:H-1}, s'\right], \tag{2.11}$$

or, by definition (V^* being the optimal value function),

$$= \max_{\mu^1_t \in \Delta(\mathcal{A}^1)} \min_{\mu^2_t \in \Delta(\mathcal{A}^2)} \mathbb{E}[r(S_t, A_t) \mid \mu^1_t, \mu^2_t, s_t] + \gamma \sum_{s'} \Pr(s' \mid s_t, \mu^1_t, \mu^2_t) V^*_{t+1}(s'). \tag{2.12}$$

As in MDPs, the problem of computing $V^*_{t+1}(s')$ for reachable states s' corresponds to a subproblem. Similarly, we define a *subgame* by (i) a state s', (ii) a time step (here, $t+1$), and (iii) the problem of finding a NES for the criterion $\mathbb{E}\left[\sum_{\tau=t+1}^{H-1}\gamma^{\tau-(t+1)} r(S_\tau, A_\tau) \mid \beta^1_{t+1:H-1}, \beta^2_{t+1:H-1}, s'\right]$. Interestingly, it can be shown that Bellman's optimality principle applies to the computation of Nash equilibrium strategy profiles, *i.e.,* knowing Nash equilibrium strategy profiles for all subgames at $t+1$, one can construct a Nash equilibrium strategy profile for the game starting at time step t. In other words, the problem of computing a Nash equilibrium strategy profile in a stochastic game with finite horizon has optimal substructure.

Stationary Strategies for Infinite-Horizon zs-SGs Again, when it comes to infinite-horizon games, we assume that $\gamma < 1$.

It was shown (Shapley 1953) that (i) min - max is equal to max - min in stationary strategies $\mathcal{S} \to \Delta(\mathcal{A}^i)$ (hence there exists a Nash equilibrium strategy profile in stationary strategies), and that (ii) the minimax value in stationary strategies is as good as any Nash equilibrium behavioral strategy profile.

One of the main ingredients to prove the previous statements is, again, Bellman's optimality principle applied to any subgame, *i.e.,*

$$\forall s, \forall t, \max_{\beta^1:\mathcal{S}\to\Delta(\mathcal{A}^1)} \min_{\beta^2:\mathcal{S}\to\Delta(\mathcal{A}^2)} \mathbb{E}\left[\sum_{\tau=t}^{\infty}\gamma^{\tau-t} r(S_\tau, A_\tau) \mid s, \beta^1, \beta^2\right] \tag{2.13}$$

$$= \max_{\mu^1 \in \Delta(\mathcal{A}^1)} \min_{\mu^2 \in \Delta(\mathcal{A}^2)} \mathbb{E}[\gamma^t r(S_t, A_t) \mid \mu^1, \mu^2] \tag{2.14}$$

$$+ \gamma \max_{\beta^1 : \mathcal{S} \to \Delta(\mathcal{A}^1)} \min_{\beta^2 : \mathcal{S} \to \Delta(\mathcal{A}^2)} \sum_{s'} \Pr(s' \mid s_0, \mu^1, \mu^2) \mathbb{E}\left[\sum_{\tau=t+1}^{\infty} \gamma^{\tau-t-1} r(S_\tau, A_\tau) \mid \beta^1, \beta^2, s' \right], \tag{2.15}$$

or, by definition of the optimal value function,

$$\stackrel{\text{def}}{=} \max_{\mu^1 \in \Delta(\mathcal{A}^1)} \min_{\mu^2 \in \Delta(\mathcal{A}^2)} \mathbb{E}[r(S_t, A_t) \mid s, \mu^1, \mu^2] + \gamma \sum_{s'} \Pr(s' \mid s_0, \mu^1, \mu^2) V^*(s'). \tag{2.16}$$

The resulting operator is γ-contracting and is known as *Shapley's operator* and its unique fixed point is the optimal value function V^*.

The existence of optimal stationary strategies is particularly relevant in practice as it highly decreases the dimensionality of the search space. For example, approximating a stochastic game with time-dependent strategies by an ϵ-close game with finite horizon and turning it into a normal-form game could yield a very large LP, that would be intractable in practice. Typical cases in which it would not are games with a small number of reachable states starting from an initial s_0. In such specific settings, a dynamic programming algorithm solving a finite-horizon approximation might find a solution more quickly than value iteration applied to the infinite-horizon game.

We shall now present partially observable stochastic games that combine both imperfect information and a multiplicity of players.

2.1.3.5 Partially Observable Stochastic Games

Here, we first give basic definitions about POSGs[19], including the solution concept we will work with.

The POSG formalism can be obtained by extending the SG one, through adding partial observability (similarly to our introduction of POMDPs as extensions of MDPs). Mainly, an observation function $\mathcal{O} : \mathcal{Z} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ would be defined. Still, for conciseness (and since it is slightly more general), the original definition of POSG (Hansen et al. 2004) merges the transition and observation functions into a state transition and observation function $P : \mathcal{A} \times \mathcal{S} \times \mathcal{Z} \times \mathcal{S} \to [0, 1]$.

Definition 2.1.25 (POSGs)**.** *A POSG is defined by a tuple* $\langle n, \mathcal{S}, \times_{i=1}^n \mathcal{A}^i, \times_{i=1}^n \mathcal{Z}^i, P, r, H, \gamma, b_0 \rangle$*, where*

- n *is the number of players;*
- $\mathcal{S}$ *is a finite set of states;*
- $\mathcal{A}^i$ *is (player)* i*'s finite set of actions and we denote* $\mathcal{A} = \times_i \mathcal{A}^i$*;*
- $\mathcal{Z}^i$ *is* i*'s finite set of observations and we denote* $\mathcal{Z} = \times_i \mathcal{Z}^i$*;*
- $\forall (\boldsymbol{a}, \boldsymbol{z}) \in \mathcal{A} \times \mathcal{Z}$, $P_{\boldsymbol{a}}^{\boldsymbol{z}}(s'|s)$ *is the probability of transiting to state* s'*, receiving observations* $\boldsymbol{z}$ *when actions* $\boldsymbol{a}$ *are performed in state* s[20]*;*
- $\forall i$, $r^i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ *is the reward function for player* i*;*
- $H \in \mathbb{N}$ *is a temporal horizon;*
- $\gamma \in [0, 1]$ *is a discount factor; and*
- b_0 *is the initial belief state,* i.e.*, a probability distribution over states at* $t = 0$*.*

[19] In the literature, one can also encounter the *partially observable Markov game (POMG) terminology (Liu et al. 2022; Kozuno et al. 2021).*

[20] Note that if we had defined a POSG with functions T and $\mathcal{O}$ it would hold that $\forall (s, \boldsymbol{a}, \boldsymbol{z}, s')$, $P_{\boldsymbol{a}}^{\boldsymbol{z}}(s'|s) = T(s, \boldsymbol{a}, s') \cdot O(\boldsymbol{z}, \boldsymbol{a}, s')$.

Note that, in this definition, the case $H = \infty$ is forbidden, contrary to before, where $H = \infty$ and $\gamma < 1$ was allowed. This is due to Chapter 3 studying the case $H < \infty$, because any discounted ($\gamma < 1$) infinite-horizon POSG can be approximated ϵ-closely by a finite-horizon one. For more details on this specific point, see Section 3.1.3.2, Proposition 3.1.30.

If $n = 1$, the game is a POMDP. If $n > 1$ and $\forall i, j \in \{1, \dots, n\}^2$, $r^i = r^j$, then the game is called a *common-payoff partially observable stochastic game* (cp-POSG), or equivalently a *decentralized partially observable Markov decision process* (Dec-POMDP). Finally, if $n = 2$ and $r^1 = -r^2$, the game is called a *zero-sum partially observable stochastic game* (zs-POSG), and, as usual, we denote $r \equiv r^1$.

We recall the following concepts and definitions[21]:

θ^i_τ $= (a^i_0, z^i_1, \dots, a^i_{\tau-1}, z^i_\tau)$ is a length-τ *action-observation history* (AOH) for i. We denote Θ^i_τ the set of all AOHs for player i at horizon τ such that any AOH θ^i_τ is in $\cup_{t=0}^{H-1}\Theta^i_t$.

β^i_τ is a *(behavioral) decision rule* (DR) at τ for i, *i.e.*, a mapping from private AOHs in Θ^i_τ to *distributions* over private actions. $\beta^i_\tau(\theta^i_\tau, a^i)$ is the probability to pick a^i when facing θ^i_τ.

$\beta^i_{\tau:\tau'} = (\beta^i_\tau, \dots, \beta^i_{\tau'})$ is a *behavioral strategy* for i from time step τ to τ' (included).

When considering both players, the last 3 concepts become:

$\boldsymbol{\theta}_\tau$ $= (\theta^1_\tau, \theta^2_\tau)$ $(\in \boldsymbol{\Theta} = \cup_{t=0}^{H-1}\boldsymbol{\Theta}_t)$, a *joint AOH* at τ,

$\boldsymbol{\beta}_\tau$ $= \langle \beta^1_\tau, \beta^2_\tau \rangle$ $(\in \mathcal{B} = \cup_{t=0}^{H-1}\mathcal{B}_t)$, a *decision rule profile*, and

$\boldsymbol{\beta}_{\tau:\tau'} = \langle \beta^1_{\tau:\tau'}, \beta^2_{\tau:\tau'} \rangle$, a *behavioral strategy profile*.

Example 2.1.26 (Matching Pennies as a zs-POSG)**.** *Without loss of generality, we here formalize the matching pennies game, introduced in Example 2.1.4, as a zs-POSG (as illustrated in Figure 2.4). For pedagogical purposes, we artificially see it as a "sequential-move" POSG by making player* 1 *pick her action at* $t = 0$*, and player* 2 *at* $t = 1$*, hence the tuple* $\langle \mathcal{S}, \mathcal{A}^1, \mathcal{A}^2, \mathcal{Z}^1, \mathcal{Z}^2, P, r, H, \gamma, b_0 \rangle$ *where:*

- $\mathcal{S} = \{s_i, s_h, s_t\}$*, where* s_i *is the initial state, and* s_h *and* s_t *represent a memory of* 1*'s last move: respectively "head" or "tail";*
- $\mathcal{A}^1 = \mathcal{A}^2 = \{a_h, a_t\}$ *for playing "head"* (a_h) *or "tail"* (a_t)*;*
- $\mathcal{Z}^1 = \mathcal{Z}^2 = \{z_n\}$ *a trivial "none" observation;*
- $P^{\boldsymbol{z}}_{\boldsymbol{a}}(s'|s) = T(s, \boldsymbol{a}, s') \cdot \mathcal{O}(\boldsymbol{a}, s', \boldsymbol{z})$*, using the next two definitions:*
  - T *is deterministic and such that* ($\cdot$ *is used to denote "for all")*
    - $T(\cdot, (a_h, \cdot), s_h) = 1.0$,
    - $T(\cdot, (a_t, \cdot), s_t) = 1.0$*;*
  - $\mathcal{O}$ *is deterministic and always returns 1.0 for the only possible observation "*z_n*";*
- r *is such that*
  - $r(s_i, \cdot, \cdot) = 0$,
  - $r(s_t, (\cdot, a_t)) = r(s_h, (\cdot, a_h)) = +1$,
  - $r(s_t, (\cdot, a_h)) = r(s_h, (\cdot, a_t)) = -1$*;*
- $H = 2$*;*
- $\gamma = 1$*;*
- b_0 *is such that* $b_0(s_i) = 1$.

[21] We also recall that an influence diagram of a two-player POSG was given in Figure 1.1

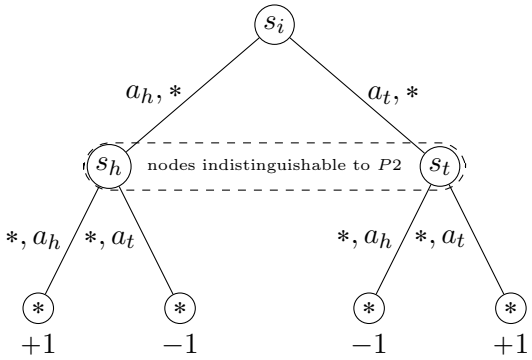


Figure 2.4: Simplified tree representation of the sequentialized Matching Pennies game. Irrelevant actions, noted $*$, allow merging edges with the same action for (i) player 2 at $t = 0$, and (ii) player 1 at $t = 1$. Notes: (a) Due to irrelevant actions, this game can be seen as an extensive form game, despite players acting simultaneously. (b) Players only know about their past action history (in this observation-free game).

In this game, only the action of player 1 *at* $t = 0$ *and the action of player* 2 *at* $t = 1$ *are relevant. Besides,* 1*'s action being hidden to* 2*, everything happens as if the two important actions (player* 2 *at* $t = 1$ *and player* 1 *at* $t = 0$*) were simultaneous (hence the equivalence with the usual* NFG*).*

Nash Equilibria for zero-sum POSGs

Since the game has finite horizon, we consider the finite criterion. Therefore, player 1 (respectively 2) wants to maximize (resp. minimize) the expected return, or *value*, of behavioral strategy profile $\boldsymbol{\beta}_{0:H-1}$, defined as the discounted sum of future rewards, *i.e.,*

$$V_0(\boldsymbol{\beta}_{0:H-1}) = E\left[\sum_{t=0}^{H-1} \gamma^t r(S_t, A_t) \mid \boldsymbol{\beta}_{0:H-1}\right].$$

This leads to the solution concept of Nash equilibrium strategy (NES). In such a game, all NESs have the same Nash-equilibrium value (NEV), $V_0^* \stackrel{\text{def}}{=} V_0(\beta^{1*}_{0:H-1}, \beta^{2*}_{0:H-1})$.

An interesting question here is whether the problem of computing a Nash equilibrium strategy profile possesses an optimal substructure (*i.e.,* can one construct a Nash equilibrium strategy profile $\boldsymbol{\beta}_{0:H-1}$ based on some Nash equilibrium strategy profiles of "subproblems", where the notion of "subproblem" is to be defined). For the zs-SG subclass, a subproblem consisted in subgames rooted at states, which makes sense as players have perfect information. Thus, any subgame can be experimented by players at execution phase (provided that the state has non-zero probability to be reached). But what is a subproblem in zs-POSGs that can indeed be experimented by players at execution time? For example, players' past strategies (as used as a root of subproblems in HSVI-like approaches for cp-POSGs) would never be known to a player at execution time since she does not know her opponent's past strategy. Then, can one define subproblems that are unrelated to "real" situations and still retrieve a valid Nash equilibrium strategy profile?

A positive answer to the previous concern was given for subclasses of zs-POSGs, *e.g.,* zero-sum one-sided partially observable stochastic games (zs-OS-POSG) which we introduce below.

2.1.3.6 Zero-Sum One-Sided Partially Observable Stochastic Games

A zero-sum one-sided partially observable stochastic game (Sorin 2003; Horák et al. 2017) (zs-OS-POSG) is a zs-POSG in which player 2's observations reveal the state of the game and 1's history. While such an assumption might appear to strongly lower the generality of zero-sum POSGs, it is actually relevant for many practical settings. The cases in which one player is perfectly aware of the state of the game while the other one has imperfect information are particularly present in security problems, as illustrated by the well-known Scotland Yard game.

Example 2.1.27 (Scotland Yard)**.** *In this pursuit-evasion game, a criminal tries to escape the police by moving in a graph representing London's streets. The locations of the patrolling policemen are known to the criminal, providing her full information about the current state of the game while her location remains unknown to the police.*

The structure in the observation function in a `zs-OS-POSG` translates into structure for players' optimal strategies and, thus, optimal value function. As the criminal is aware of the current state of the game and the policemen's belief about her location while playing, her strategies can depend on such concise information instead of reasoning upon every reachable history as in `zs-POSGs`.

We now move on to the (non-exhaustive) presentation of approaches to solve (*i.e.,* find a Nash equilibrium with highest value of) the games introduced during this section.

2.2 Solving Algorithms in Game Theory

In this section, we come back to three approaches mentioned in the introduction to optimally solve subclasses of `POSGs`: regret minimization, mathematical programming and Bellman's optimality principle-based approaches. Each section starts by detailing the algorithmic scheme for the simpler case (*e.g.,* `NFGs` or `MDPs`), then moves on to discussing the generalization of the scheme to `zs-POSGs`, and finally discusses the levers unveiled by each approach.

2.2.1 Regret Minimization

Regret minimization is a general algorithmic scheme that allows learning in games through *self-play* (see Section 3.3.3 for a high-level discussion). We below focus on the *regret-matching* rule (Blackwell 1956; Abernethy et al. 2011; Farina et al. 2021) that specifies the regrets computation and the update of the mixed strategy after each self-play iteration. Given a `zs-NFG`[22], Algorithm 2.1 is a pseudo-code implementation of the regret minimization principle using regret matching. The algorithm simulates a repetition of the `zs-NFG` over time. At each time step t, players sample an action $\boldsymbol{a}_t$ from a strategy profile $\boldsymbol{\mu}_t$, where $\boldsymbol{\mu}_0$ is a randomly initialized mixed strategy profile, and strategies $\boldsymbol{\mu}_t$ for $t > 0$ are computed using the procedure described below. At time step t, players alternatively study the impact of replacing action a_t^i by another action $\tilde{a}^i$ each time a_t^i was played in the past ($\tau < t$). The *regret* for this change is computed as the difference between the sum of rewards for time step $\tau \in \{0, \dots, t-1\}$ with and without replacement. This process is done for all alternatives $\tilde{a}^i$ and the probability to pick $\tilde{a}^i$ for player i (*i.e.,* $\mu_t^i(\tilde{a}^i)$) is updated by giving proportionally more weight to the actions whose computed regrets were positive (Line 7). The resulting algorithm converges almost surely towards a correlated equilibrium of the game (Hart et al. 2000). For the specific case of two-player zero-sum games, the algorithm converges towards a Nash equilibrium.

2.2.1.1 Regret Minimization for `zs-EFGs`

Later, Zinkevich et al. generalized this result in 2007 to dynamic games with imperfect information by applying a regret-matching update rule at each information state. Multiple changes were needed to adapt the regret matching scheme to the imperfect information setting and we refer for example to the very pedagogical article of Neller et al. (2013) for further details. The resulting regret-matching rule is based on a special type of regret, namely counterfactual regret, introduced by Zinkevich et al. in 2007. The resulting algorithm asymptotically converges towards a Nash equilibrium, with an error bound for the regret of player i in $\mathcal{O}(|I_i| \cdot \sqrt{|A^i|}/\sqrt{T})$, where T is the number of trials, I_i is the number of information sets of player i in the game and $|A^i|$ is the larger set of actions of player i (in the inclusion sense, with respect to every set of actions available to player i for all of her possible information sets in the game). Interestingly, this bound is linear in i's number of information sets.

[22] Again, we assume that the game is zero-sum, even though Hart et al.'s procedure was more general, dealing with N player `NFGs`, and converging towards a coarse correlated equilibrium.

Algorithm 2.1: Regret matching for `zs-NFGs` (Hart et al. 2000)

Input : Game matrix $(M_{i,j})_{i,j}$; players' action set $\mathcal{A}^1$ and $\mathcal{A}^2$
Input : α an hyperparameter (see Hart et al. (2000) for domain and interpretation of α); N the number of self plays

Initialize $\boldsymbol{\mu}_0 \in \Delta(\mathcal{A}^1) \times \Delta(\mathcal{A}^2)$ randomly
for $t \in \{0, \dots, N-1\}$ **do**
  Sample $\boldsymbol{a}_t \sim \boldsymbol{\mu}_t$
  for *each player* i **do**
    for *each action* k $(k \neq a_t^i)$ *of player* i **do**
      $R^i(a_t^i, k) \leftarrow \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{1}_{\{a_\tau^i = a_t^i\}} \cdot (u^i(\langle \boldsymbol{a}_\tau^{\neg i}, k \rangle) - u^i(\boldsymbol{a}_\tau))$ `/* regret of player` i `for changing` a_t^i `to action` k `*/`
      $\mu_{t+1}^i(k) \leftarrow \frac{1}{\alpha} \frac{1}{t} \max\{0, R^i(a_t^i, k)\}$ `/* probability of playing` k `at next round is proportional to regret if positive */`
    $\mu_{t+1}^i(a_t^i) \leftarrow 1 - \sum_{k \neq a_t^i} \mu_{t+1}^i(k)$
return $\boldsymbol{\mu}_N$

When considering a `zs-EFG` derived from a `zs-POSG`, the number of information sets for player i and for planning horizon H is $\sum_{t=0}^{H-1} (|\mathcal{A}^i||\mathcal{Z}^i|)^t = \mathcal{O}([|\mathcal{A}^i||\mathcal{Z}^i|]^{H-1})$. Such exponential complexity becomes prohibitive when H grows and represents the main bottleneck of running counterfactual regret minimization for large `EFGs`. Note that the complexity of one iteration over the whole game tree (*i.e.,* visiting all information sets) is roughly the same as computing a best-response given a strategy of the opponent. The latter optimization problem is known to be intractable in practice for some games (Johanson et al. 2011).

Tackling this prohibitive complexity was studied (*e.g.,* by Lanctot et al. (2009) using sampling methods) and is one possible lever offered by the counterfactuel regret minimization scheme. Other levers (*e.g.,* depth-limited subgame solving) are detailed in Section 3.3.3.

Overall, this scheme has been a building block for solving zero-sum imperfect information games since its discovery. Many approaches building on top of this simple algorithmic scheme were designed since (again, see Section 3.3.3 for further details).

2.2.2 Mathematical programming

Under linearity assumptions on players' payoff function with respect to their mixed strategies, a Nash equilibrium strategy profile can be computed in polynomial time in various settings. Throughout this section, we will discuss the computation of a Nash equilibrium through mathematical programming for various types of games, from normal-form games to dynamic games, possibly with imperfect information. We assume that games are zero-sum, but, when relevant, we present at a high level linear programs solving the general-sum version of the considered type of game.

2.2.2.1 (Mixed-Integer) Linear Programming for Normal-Form and Bayesian Games

Let $G = \langle \mathcal{A}^1, \mathcal{A}^2, M \rangle$ be a `zs-NFG`. As often in game theory, the zero-sum setting eases the search for optimal solutions of the game. In this case, the simple linear program given in Proposition 2.2.1 computes an exact Nash equilibrium strategy profile of the game G. It derives from von Neumann's minimax theorem, replacing the min (resp. max) operator over a finite number of pure strategies for player 2 (resp. 1) by a set of constraints, one per pure strategy. Nash equilibria being interchangeable offers the possibility to search for individual Nash equilibrium strategies for each player independently. The latter are given for player 1 (resp. 2) by solutions of the primal (resp. dual) linear program.

Proposition 2.2.1 (LP to solve `zs-NFGs` (Shoham et al. 2008))**.** *Let* $\langle \mathcal{A}^1, \mathcal{A}^2, M \rangle$ *be a* `zs-NFG`*. Then,* $V^* = \max_{\mu^1 \in \Delta(\mathcal{A}^1)} \min_{\mu^2 \in \Delta(\mathcal{A}^2)} \mu^{1,\top} \cdot M \cdot \mu^2$ *is also the value of the (primal) linear program*

$$\max_{V,\mu^1} \quad V \tag{2.17a}$$

$$\text{s.t.} \quad \sum_{a^1 \in \mathcal{A}^1} \mu^1(a^1) M(a^1, a^2) \geqslant V \quad \forall a^2 \in \mathcal{A}^2, \tag{2.17b}$$

$$\mu^1(a^1) \geqslant 0, \quad \forall a^1 \in \mathcal{A}^1, \quad \textit{and} \tag{2.17c}$$

$$\sum_{a^1} \mu^1(a^1) = 1, \tag{2.17d}$$

whose dual is

$$\min_{V,\mu^2} \quad V \tag{2.18a}$$

$$\text{s.t.} \quad \sum_{a^2 \in \mathcal{A}^2} \mu^2(a^2) M(a^1, a^2) \leqslant V \quad \forall a^1 \in \mathcal{A}^1, \tag{2.18b}$$

$$\mu^2(a^2) \geqslant 0 \quad \forall a^2 \in \mathcal{A}^2, \quad \textit{and} \tag{2.18c}$$

$$\sum_{a^2} \mu^2(a^2) = 1. \tag{2.18d}$$

Besides, any mixed strategy profile $(\mu^{1,*}, \mu^{2,*})$ where $\mu^{1,*}$ is a solution of the primal linear program and $\mu^{2,*}$ a solution of the dual is a Nash equilibrium strategy profile.

As a by-product, this also provides a complexity result: finding a `NES` in a `zs-NFG` has, at worst, polynomial time and linear space complexity w.r.t. players' set of action sizes.

Looking back at Example 2.1.4, the generic linear program of Proposition 2.2.1 is

$$\max_{V,\mu^1=(p_h,p_t)} \quad V \tag{2.19a}$$

$$\text{s.t.} \quad p_h \cdot 1 + p_t \cdot -1 \geqslant V, \tag{2.19b}$$

$$p_h \cdot -1 + p_t \cdot 1 \geqslant V, \tag{2.19c}$$

$$p_h + p_t = 1. \tag{2.19d}$$

Remark 2.2.2 (General-sum Bimatrix Game)**.** *The* Lemke-Howson *algorithm (Lemke et al. 1964) permits computing a Nash Equilibrium strategy profile of a 2-player "general-sum" normal-form game (*i.e., *with two matrices M^1 and M^2, player 1 (resp. player 2) aiming at maximizing the payoff functions derived from M^1 (resp. M^2)). The algorithm's worst case is to visit exponentially many vertices, and consequently may take exponential time before finding a Nash equilibrium.*

Remark 2.2.3 (Common-Payoff Normal-Form Games)**.** *Let $\langle n, \times_i \mathcal{A}^i, \times_i M^i \rangle$ be an n-player `NFG` (direct extension of Definition 2.1.9) where, $\forall i, j \in \{1, \ldots, n\}^2$, $M^i = M^j$ and $\forall i$, $M^i \in \times_i \mathbb{R}^{|\mathcal{A}^i|}$. The problem of finding a Nash equilibrium strategy profile with the highest value is a single-criterion optimization problem, and there exists at least one solution in pure strategy profiles (*i.e., *action profiles) (Oliehoek et al. 2008). Then, searching for a Nash equilibrium strategy profile with the highest value,* i.e., *a global optimum, amounts to a linear search over all elements of any of the matrix M^i.*

While zero-sum Bayesian games appear harder to solve than normal-form games as they introduce uncertainty regarding other players' types, Proposition 2.1.16 presents a piecewise linear and convex/concave property of the optimal value function w.r.t. some marginal distribution. This property also allows 1 to search for an optimal strategy by reasoning independently upon the possible types of her opponent, while the latter best responds with deterministic strategies. Since there is a finite number of deterministic strategies, solving a zero-sum Bayesian game can be turned into solving a linear program (and its dual) and the complexity of solving zero-sum Bayesian games is also polynomial.

Proposition 2.2.4 (Linear Program for a zs-BG (Harsanyi 1968)). *For any Bayesian game* $\langle \mathcal{A}^1, \mathcal{A}^2, \Theta^1, \Theta^2, p, u \rangle$, *the Nash equilibrium value* $\min_{\beta^2} \max_{\beta^1} \mathbb{E}_{\boldsymbol{a} \sim \boldsymbol{\beta}}[u(\boldsymbol{a})]$ *is also the optimal value of the linear program*

$$\min_{(V_{\theta^1})_{\theta^1}, \beta^2} \sum_{\theta^1} \Pr(\theta^1 \mid p) V_{\theta^1} \tag{2.20a}$$

$$\text{s.t.} \sum_{\theta^2} \sum_{a^2} \Pr(a^2 \mid \beta^2, \theta^2) \Pr(\theta^2 \mid \theta^1, p) u(a^1, a^2) \geqslant V_{\theta^1}, \ \forall \theta^1 \in \Theta^1, \ \forall a^1 \in \mathcal{A}^1, \tag{2.20b}$$

$$\sum_{a^2} \Pr(a^2 \mid \theta^2, \beta^2) = 1, \ \forall \theta^2 \in \Theta^2. \tag{2.20c}$$

In the linear program (2.20), variables

- β^2 is a family of probability distributions $(\beta^2_{\theta^2})_{\theta^2}$, one for each possible type θ^2; and
- V_{θ_1} each correspond to the value obtained if 2 plays β^2, assuming that 1 experienced θ^1.

Constraint (2.20b) represents 1 best responding (with a provably optimal *pure* action) to β^2, knowing θ^1, while (2.20c) ensures that for any type θ^2, $\beta^2_{\theta^2}$ is a valid probability distribution over $\mathcal{A}^2$.

2.2.2.2 Linear Programming for zs-EFGs

Moving on to dynamic games with imperfect information, we remind the reader that a player's behavioral strategy is a collection of mappings providing, for each time step, a probability distribution over the player's actions for each of her information sets in the EFG. Unfortunately, the finite criterion is not linear with respect to players' behavioral strategies (as EFGs and POSGs are very close, we refer to the same result proven for POSGs in Proposition 3.1.2 (page 43)). Interestingly however, the finite criterion is linear with respect to players' mixed strategies. This allows turning the extensive-form game into a normal-form game (the EFG's pure strategies serving as actions in the NFG) and applying the linear programs presented in Proposition 2.2.1. There is no loss in considering behavioral strategies or mixed strategies, as Kuhn et al. showed in 1953 that assuming perfect recall and timeability of EFGs (our default assumption) leads to mixed strategies being equivalent to behavioral ones.

Remark 2.2.5 (Mixed Strategies and Behavioral strategies). *Behavioral and mixed strategies are equivalent, but both concepts offer complementary benefices, as illustrated by the ability to construct an NFG to solve the zs-EFG using mixed strategies. Behavioral strategies, for their part, are, for example, more suited to (and easier to manipulate for) approaches performing local computations for intermediate decision points in the game, as behavioral strategies are defined as sequences of decision rules.*

However, the size of the resulting normal-form game is exponential with respect to the size of the game tree (Koller et al. 1996), so that such a transformation is intractable in practice. Indeed, the number of pure strategies for player i in an EFG derived from a POSG (see (Vojtěch Kovařík et al. 2022a) for a detailed presentation of the conversion) is[23]

$$|\mathcal{A}^i|^{\sum_{t=0}^{H-1} |\mathcal{Z}^i|^t} = |\mathcal{A}^i|^{\frac{|\mathcal{Z}^i|^H - 1}{|\mathcal{Z}^i| - 1}}. \tag{2.21}$$

Later on, in 1996, Koller et al. introduced another equivalent strategy concept, namely *realization weights*, and proved the equivalence between solving the previous normal-form game and an LP whose variables are vectors of realization weights, *i.e.,* a *realization plan*. Realization weights for player i correspond to every possible probability for a sequence of actions she picked in information sets, assuming that the corresponding information sets are reached. They discard redundancy contained in mixed strategies. The number of parameters needed to represent the

[23] Note: in EFGs, the number of pure strategies is seen as $|\mathcal{A}^i|^{\sum_{t=0}^{H-1} [|\mathcal{A}^i| \cdot |\mathcal{Z}^i|]^t}$, but as noted by Koller et al., the resulting normal-form game would suffer from redundant rows.

space of realization weights is linear in the number of information sets of a player, which, in a POSG is:

$$\sum_{t=0}^{H-1} (|\mathcal{A}^i| \cdot |\mathcal{Z}^i|)^t = \mathcal{O}([|\mathcal{A}^i| \cdot |\mathcal{Z}^i|]^{H-1}). \tag{2.22}$$

It is noteworthy that the dimension needed to represent the space of realization weights is exponentially lower than the dimension of distributions over pure strategies (*i.e.,* mixed strategies). Overall, the LP using realization weights, namely the sequence-form linear program (SFLP), has linear size (in sparse form) with respect to the size of the game tree.

2.2.2.3 Double Oracle Algorithmic Scheme

Linear programs using mixed strategies or even SFLPs can be prohibitively big, for example when they are obtained by flattening time in a large zs-EFG (*e.g.,* poker, (Bowling et al. 2015, Figure 2)). For such programs, we might want to design an anytime algorithm to be able to retrieve decent strategies in reasonable time without having to wait for the exact convergence. Even though the LP solvers are anytime once the problem is constructed, the LP itself might not even fit in memory (Bowling et al. 2015, Figure 2). Thereby, designing an anytime algorithm based on linear programming (McMahan et al. 2003) has been a major advance to tackle real-world games, especially when the approach was applied to SFLPs (Bošanský et al. 2014).

In essence, the algorithmic scheme starts by solving a restricted program (either in normal-form or in sequence form) with only one row and column (corresponding to players only having one possible pure strategy/realization plan) and keeps adding rows and columns (*i.e.,* possible pure strategies/realization plans) until the sets of rows and columns stabilize (which occurs in particular if the unrestricted game is reached). Rows and columns are selected according to some heuristic which typically relies on computing best responses. A pseudo-code for the double-oracle algorithm is given in Algorithm 2.2 (but, for simplicity, is only given for linear programs using mixed strategies). Empirically (Bošanský et al. 2014, Table 5), this approach often computes an exact Nash equilibrium strategy profile way before having constructed the complete linear program. The algorithm provably computed a Nash equilibrium strategy profile whenever the best responses of players already appear in their support set of pure strategies/realization plans (*i.e.,* rows or columns). The efficiency of double-oracle algorithms essentially comes from their ability to focus on the empirically small fractions of rows/columns needed to construct a Nash equilibrium strategy profile, discarding the irrelevant ones.

Still, this approach requires being able to compute multiple best responses, which might be intractable for very large games (Johanson et al. 2011). Recent developments include (i) new heuristics to populate rows and columns (including approximation of best-response computations by training oracles (Lanctot et al. 2017; McAleer et al. 2021)), (ii) different constructions of restricted games (McAleer et al. 2021), and (iii) replacing linear programming with other solving methods (*e.g.,* regret minimization) (McAleer et al. 2021; Lanctot et al. 2017).

2.2.3 Approaches Based on Bellman's Optimality Principle

Dynamic programming applied to the search for Nash equilibria in POSGs gave rise to various approaches (Hansen et al. 2004; Burch et al. 2014; Horák et al. 2017; Horák et al. 2019b). Related work section in Chapter 3 (p. 70) presents some of these complementary approaches. We particularly focus below on one particular scheme, namely heuristic search value iteration (HSVI), as it is the most relevant to the understanding of our contributions. The following presents the application of this algorithmic scheme to MDPs before moving on to more complicated settings, introducing (i) another (adversarial) player, (ii) partial observability, or (iii) both.

2.2.3.1 Single-Player Fully Observable Games: MDPs

In this section, we come back to MDPs to detail the dynamic programming solving algorithm resulting of an iterative application of Bellman's operator. Next, we introduce the heuristic

Algorithm 2.2: Double oracle for `zs-NFG`s

Data: G a `zs-NFG`

$\tilde{G}$ is a game initialized with 0 actions for 1 and 2
$\pi^1, \pi^2 \leftarrow$ two initial pure strategies for 1 and 2
$\tilde{\mathcal{A}}^1 \leftarrow \{\pi^1\}$, $\tilde{\mathcal{A}}^2 \leftarrow \{\pi^2\}$ /* initialize 1 and 2 set of available pure actions */
$\tilde{G} \leftarrow Game(\tilde{G}, \tilde{\mathcal{A}}^1, \tilde{\mathcal{A}}^2)$ /* construct a game $\tilde{G}$ by adding the sets $\tilde{\mathcal{A}}^1$ and $\tilde{\mathcal{A}}^2$ of pure actions to the set of pure actions for 1 and 2 in $\tilde{G}$ */
do
  $\mu^{1,*} \leftarrow PrimalLPSolution(\tilde{G})$
  $\mu^{2,*} \leftarrow DualLPSolution(\tilde{G})$
  $(\pi^{1,br}, value_1) \leftarrow bestResponse(\mu^{1,*})$
  $(\pi^{2,br}, value_2) \leftarrow bestResponse(\mu^{2,*})$
  $\tilde{\mathcal{A}}^1 \leftarrow \tilde{\mathcal{A}}^1 \cup \{\pi^{1,br}\}$
  $\tilde{\mathcal{A}}^2 \leftarrow \tilde{\mathcal{A}}^2 \cup \{\pi^{2,br}\}$
  $\tilde{G} \leftarrow Game(\tilde{G}, \tilde{\mathcal{A}}^1, \tilde{\mathcal{A}}^2)$
while $value_1 \neq value_2$
return $\langle \mu^{1,*}, \mu^{2,*} \rangle$

search value iteration scheme (`HSVI`), based on dynamic programming and on a heuristic selection of nodes in the game tree to update.

Finite-horizon `MDP`s In the case of a finite horizon `MDP`, Bellman's optimality principle leads to recursively computing V_0^*, the base case being the exact computation of $V_{H-1}^* : s \mapsto \max_a r(s,a)$. Still, such an algorithm would compute multiple times the optimal value of the same nodes, as a certain state s at time step $t \in \{1, \dots, H-1\}$ could be reached through multiple different paths $s_0, \dots, s_{t-1}$. Instead, a dynamic programming approach, given in Algorithm 2.3, keeping in memory the optimal values of future time steps, provides a more time-efficient algorithm.

Algorithm 2.3: Dynamic Programming for (finite-horizon) `MDP`s

Input : $H \in \mathbb{N}^*$
$\forall s \in \mathcal{S},\ V_H(s) \leftarrow 0$
$t \leftarrow H-1$
while $t \geqslant 0$ **do**
  for $s \in S$ **do**
    $V_t(s) \leftarrow \max_a r(s,a) + \gamma \sum_{s'} T(s,a,s') V_{t+1}(s')$
  $t \leftarrow t-1$
return $(V_k)_{k \in \{0,\dots,H-1\}}$

In a significant number of real-life applications, the initial state of the system is known to the player. Focusing on one specific initial configuration s_0 is of great computational help, as it allows focusing on the branches of the tree contributing the most to the optimal value at s_0. For example, if a state s_1 at depth 1 is not reachable under any action starting from s_0, studying s_1 is completely irrelevant. Various algorithms take advantage of this observation to implement efficient algorithms providing reasonably good strategies in way less time than value iteration. Heuristic search value iteration (Smith et al. 2005) is one of them (initially introduced for infinite-horizon `POMDP`s), and is given in Algorithm 2.4. A key difference between `HSVI` and dynamic programming is that `HSVI`'s goal is to compute ϵ-optimal solutions, where $\epsilon > 0$ is an input. It allows `HSVI` to leverage the knowledge of the initial state s_0 by (i) elegantly focusing on the next state s_{t+1} that contributes the most to the uncertainty at current state s_t, and (ii) guiding search optimistically. The price to pay, however, is the maintenance of upper-bound approximations, in addition to classical lower-bound approximations.

Algorithm 2.4: `HSVI` for (finite-horizon) `MDP`s

Input : $H \in \mathbb{N},\ \epsilon > 0$
Input : s_0 a state
Fct `Solve`
  $\forall t \in \{0, \dots, H-1\}$, Initialize $\overline{V}_t : S \to \mathbb{R}$ with optimistic value
  $\forall t \in \{0, \dots, H-1\}$, Initialize $\underline{V}_t : S \to \mathbb{R}$ with pessimistic value
  $\forall s, \underline{V}_H(s) \leftarrow 0$ and $\forall s, \overline{V}_H(s) \leftarrow 0$
  while $\overline{V}_0(s_0) - \underline{V}_0(s_0) \geqslant \epsilon$ **do**
    `Explore` $(s_0, 0)$
  return $(\underline{V}_t)_{t \in \{0,\dots,H-1\}}$
Fct `Explore`(s_t^*, t)
  if $\overline{V}_t(s_t^*) - \underline{V}_t(s_t^*) \geqslant \epsilon\gamma^{-t}$ **then**
    `Update` (s, t)
    $a^* \leftarrow \arg\max_a r(s,a) + \gamma \sum_{s'} T(s,a,s')\overline{V}_{t+1}(s')$
    $s^* \leftarrow \arg\max_{s'} T(s,a^*,s')\left[\overline{V}_{t+1}(s') - \underline{V}_{t+1}(s')\right]$
    `Explore` $(s^*, t+1)$
    `Update` (s, t)
  return
Fct `Update` (s, t)
  $\overline{V}_t(s) \leftarrow \max_a r(s,a) + \gamma \sum_{s'} T(s,a,s')\overline{V}_{t+1}(s')$
  $\underline{V}_t(s) \leftarrow \max_a r(s,a) + \gamma \sum_{s'} T(s,a,s')\underline{V}_{t+1}(s')$

Algorithm 2.5: Value Iteration (synchronous version) for (infinite-horizon) `MDP`s

Input : $\epsilon \in \mathbb{R}^{+,*}$
Initialize $V_0 : S \to \mathbb{R}$ with any values
$n \leftarrow 0$
do
  for $s \in S$ **do**
    $V_{n+1}(s) \leftarrow \max_a r(s,a) + \gamma \sum_{s'} T(s,a,s')V_n(s')$
    $n \leftarrow n + 1$
while $\|V_{n+1} - V_n\|_\infty \geqslant \epsilon$
return V

More formally, given a state s_t, the most promising action a^* is

$$a^* \overset{\text{def}}{\in} \arg\max_a r(s,a) + \sum_{s'} T(s,a,s')\overline{V}_{t+1}(s'),$$

where $\overline{V}_{t+1}$ is the upper-bound approximation at time step $t+1$. A next state s_{t+1}^* contributing the most to the uncertainty at state s_t for action a^*, defined as

$$s^* \overset{\text{def}}{\in} \arg\max_{s'}\ T(s,a^*,s')\left[\overline{V}_{t+1}(s') - \underline{V}_{t+1}(s')\right], \tag{2.23}$$

is selected. Overall, `HSVI` performs trajectories in the game tree. The action and next state selections described above essentially define the forward phase, while the backward phase mainly involves updating the upper-bound and lower-bound values of each encountered state through Bellman's equation.

Infinite-Horizon Case Moving on to the infinite-horizon setting, we again assume that strategies are evaluated through the discounted criterion. As mentioned in Section 2.1.3.2, stationary

Algorithm 2.6: `HSVI` for (infinite-horizon) `MDP`s

Input : s_0 a state

Fct `Solve`
  Initialize $\overline{V} : S \to \mathbb{R}$ with optimistic value
  Initialize $\underline{V} : S \to \mathbb{R}$ with pessimistic value
  while $\overline{V}(s_0) - \underline{V}(s_0) \geqslant \epsilon$ **do**
    `Explore` $(s_0, 0)$
  return $\underline{V}$
Fct `Explore`(s, t)
  if $\overline{V}(s) - \underline{V}(s) \geqslant \epsilon\gamma^{-t}$ **then**
    `Update` (s, t)
    $a^* \leftarrow \arg\max_a r(s, a) + \gamma \sum_{s'} T(s, a, s')\overline{V}(s')$
    $s^* \leftarrow \arg\max_{s'} T(s, a^*, s') \left[\overline{V}(s') - \underline{V}(s')\right]$
    `Explore` $(s^*, t + 1)$
    `Update` (s, t)
  return
Fct `Update` (s, t)
  $\overline{V}(s) \leftarrow \max_a r(s, a) + \gamma \sum_{s'} T(s, a, s')\overline{V}(s')$
  $\underline{V}(s) \leftarrow \max_a r(s, a) + \gamma \sum_{s'} T(s, a, s')\underline{V}(s')$

strategies are sufficient for this criterion and we consequently limit ourselves to searching for an optimal one among them.

The Bellman operator is γ-contracting so that, after applying it k times, $\|V_k - V_{k-1}\|_\infty \leqslant \epsilon$ implies that $\|V_k - V^*\|_\infty \leqslant \frac{\gamma}{1-\gamma}\epsilon$ (Russell et al. 2010). Since there always exists a k ensuring $\|V_k - V_{k-1}\|_\infty \leqslant \epsilon$, Algorithm 2.5 always terminates, which implies that an ϵ'-approximation of V^* can be obtained in finite time, where $\epsilon' = \frac{\gamma}{1-\gamma}\epsilon$. A similar argument proves that trajectories performed by `HSVI` (Algorithm 2.6) have a finite maximal depth t_{max} (Smith et al. 2005).

Remark 2.2.6. *γ being strictly inferior to 1 implies that rewards are more and more negligible, so that the sum of rewards obtained from time step K onward does not contribute more than ϵ to the total sum, for a sufficiently large K. It follows that there exists a finite-horizon `MDP` that closely approximates the infinite-horizon one. The corresponding horizon H_{eff} can be computed analytically and solving the resulting `MDP` ensures an error with respect to the original optimal value less than $\gamma^{H_{eff}} \cdot \frac{R_{max} - R_{min}}{1-\gamma}$, where $R_{max} \stackrel{\text{def}}{=} \max_{s,a} r(s, a)$ and $R_{min} \stackrel{\text{def}}{=} \min_{s,a} r(s, a)$. In practice, both formulations (the infinite-horizon and the finite one) can be relevant, depending, for example, on the games' dynamics and the solving algorithm scheme.*

2.2.3.2 Two-player Zero-sum Stochastic Games

Section 2.1.3.4 stated that Bellman's optimality equations hold for finite- and infinite-horizon `zs-SG`s. Still, as for `MDP`s and `POMDP`s, dynamic programming algorithms (Algorithm 2.3 and Algorithm 2.5, switching the Bellman's operator for Shapley's one) relying on applying Shapley's operator n times become intractable for games of reasonable size (Buffet et al. 2020). This is partly due to Shapley's operator having polynomial-time complexity (since it is essentially solving an `NFG`). For large games, repeating many times this operator becomes prohibitive. `HSVI` mitigates this burden by reducing the total number of computations of Shapley's operator, leveraging the search for ϵ-optimal solutions instead of optimal ones and knowledge of the initial state.

Upper and lower bounds required for the `HSVI` scheme can be defined by tabular functions, as for `MDP`s. The main adaptation required lies in the selection of the next state to study. In `MDP`s, deciding which next state to study involves the computation of an optimistic action a for the (single) player, and the selection of the next state that contributes the most to the uncertainty, given a. But what does it mean to optimistically play for both players in a `zs-SG`? Assuming

that the game is in state s and 1 (resp. 2) picks $\overline{\pi}^1$ (resp. $\underline{\pi}^2$) relatively to the upper bound (resp. lower bound):

$$\overline{\pi}^1(s) \in \arg\max_{\pi^1(s)} \min_{\pi^2(s)} \pi^1(s) \cdot \left[\sum_{s'} r(s,\cdot,\cdot) + \gamma T(s,\cdot,\cdot,s') \overline{V}(s') \right] \cdot \pi^2(s) \tag{2.24}$$

$$\overline{\pi}^2(s) \in \arg\min_{\pi^2(s)} \max_{\pi^1(s)} \pi^1(s) \cdot \left[\sum_{s'} r(s,\cdot,\cdot) + \gamma T(s,\cdot,\cdot,s') \underline{V}(s') \right] \cdot \pi^2(s), \tag{2.25}$$

one can exhibit one of the nodes contributing the most to the uncertainty:

$$s^* \in \arg\max_{s'} \overline{\pi}^1(s) \cdot \left[T(s,\cdot,\cdot,s') (\overline{V}(s') - \underline{V}(s')) \right] \cdot \underline{\pi}^2(s). \tag{2.26}$$

This strategy profile and node selection is provably relevant as the resulting `HSVI` algorithm converges in finite time to an ϵ-`NES` of the `zs-SGs`.

2.2.3.3 Introducing Partial Observability: `POMDPs`

Contrary to `MDPs`, value functions of a `POMDP` can no longer be defined on the states of the system as it is not known to the player anymore. We will here detail a transformation, turning the `POMDP` problem into an "equivalent" continuous-state `MDP`, in which states are a summary of the complete list of data known to the player, for any time step.

A Sufficient Statistic Smallwood et al. showed in 1973 that, under three criteria, a statistic can be sufficient to optimally plan in `POMDPs`. It is required that the statistic (i) is Markovian, (ii) correctly (compared to the estimation obtained with knowledge of all data) estimates immediate rewards and (iii) correctly estimates the probability for the next observation. Here, correctly refers to equality in comparison with knowledge of all data. Under those three conditions, solving the Markov decision process defined by the evolution of the statistic allows retrieving an optimal strategy for the original `POMDP`. In `POMDPs`, a sufficient statistic commonly used is the probability distribution over the hidden states of the system, namely the *belief state*.

Definition 2.2.7 (Belief States)**.** *Given any action-observation history θ_t and an initial belief b_0, we define the statistic called* belief state *induced by θ_t as an element of $\Delta(\mathcal{S})$ such that:*

$$\forall s,\ b_{\theta_t}(s) \overset{\text{def}}{=} \Pr(s \mid b_0, \theta_t). \tag{2.27}$$

Belief states satisfy Smallwood et al.'s assumptions for sufficient statistics for optimally planning in `POMDPs`.

Theorem 2.2.8 (Sufficiency of Belief States (Garcia et al. 2008))**.** *Belief states are sufficient statistics for optimally planning in `POMDPs`,* i.e., *it holds that:*

- *$(b_t)_t$ is Markovian, and we note τ the transition function computing $b_{\tau+1} = \tau(b_t, a, z)$;*

noting b_{θ_t} the belief state obtained for some history θ_t, it also holds that:

- *b_{θ_t} estimates correctly the immediate reward, compared to complete data known to the player: $\mathbb{E}[r(S,A) \mid b_{\theta_t}] = \mathbb{E}[r(S,A) \mid \theta_t]$; and*
- *probability of next observation z for action a is $\sum_s \sum_{s'} \mathcal{O}(z,a,s') T(s,a,s') b_{\theta_t}(s)$, which is equal to $\Pr(z \mid b_0, \theta_t)$.*

Corollary 2.2.9. *Let $M = \langle \mathcal{S}, \mathcal{A}, \mathcal{Z}, T, \mathcal{O}, r, H, \gamma, b_0 \rangle$ be a `POMDP` and let $M' = \langle \mathcal{B}, \mathcal{A}, \mathtt{T}, \rho, H, \gamma, b_0 \rangle$ be an `MDP`, where:*

- *$\mathcal{B} \overset{\text{def}}{=} \bigcup_{t=0}^{H} \{b_{\theta_t}\}_{\theta_t} \subset \Delta(\mathcal{S})$ is the set of beliefs induced by histories θ_t reachable for all time steps t;*

- $\mathtt{T} : \Delta(\mathcal{S}) \times \mathcal{A} \times \Delta(\mathcal{S}) \mapsto \mathbb{R}$ *gives the probability to reach a next belief;* $\forall b_t, \forall b_{t+1}, \mathtt{T}(b_t, a, b_{t+1}) = \sum_z \Pr(z \mid b_t, a) \delta_{b_{t+1}}^{\tau(b_t,a,z)}$ *(where* δ *is the Kronecker symbol);*
- $\rho : b, a \mapsto \sum_s b(s) r(s, a)$ *is the reward model; and*
- $\mathcal{A}, H, b_0$ *and* γ *are as in the* `POMDP`*.*

It holds that M *and* M' *are equivalent regarding optimal planning (*i.e., *an optimal solution* $\pi^* : \mathcal{B} \to \mathcal{A}$ *of* M' *can be translated into an optimal solution* $\pi^* : \Theta \to \mathcal{A}$ *of* M*, and conversely). The* `MDP` *is referred to as the* belief `MDP` (`b-MDP`) *derived from the* `POMDP`*.*

If the planning horizon H is finite, only a finite number of beliefs are reachable. The state space of the `b-MDP` is thus finite; and the hypotheses to apply the dynamic programming scheme, given in Algorithm 2.3, are met. Still, the number of reachable beliefs can be large for reasonable size problems so that it remains inefficient, in general.

If the planning horizon H is infinite, we typically apply Algorithm 2.6. As for `MDP`s, an important difference with the finite criterion lies in the stationarity of optimal strategies, which implies that values functions for all time steps are equal. The number of reachable beliefs is, however, infinite. Value functions with generalization properties are consequently required, but are also welcome, as they allow knowledge transfer between beliefs.

Exhibiting Continuity Properties of the Optimal Value Function Turning a `POMDP` into a belief `MDP` is lossless regarding ϵ-close optimal planning, but it also permits leveraging some structure present in the `POMDP`. For example, the reward function ρ is linear in the belief-state space $\Delta(\mathcal{S})$, which induces continuity properties of the optimal value function.

Lemma 2.2.10 (Structure in the Value Function for Finite-Horizon `b-MDP`s (Smallwood et al., 1973))**.** *The optimal value function of a finite-horizon belief* `MDP` *is piecewise-linear and convex in the belief state space as there exists a finite collection of vectors for each time step*[24] $\alpha_t : \mathcal{S} \to \mathbb{R}$ *such that:*

$$\forall t, \forall b_t \in \Delta(S),\ V_t^*(b) = \max_{\alpha_t} [b_t \cdot \alpha_t]\,. \tag{2.28}$$

Theorem 2.2.11 (Structure in the Value Function for Infinite-Horizon `b-MDP`s (Sondik 1971))**.** *The optimal value function of a belief* `MDP` *is convex in the belief-state space and can be approximated arbitrarily closely by a* `PWLC` *function.*

Remark 2.2.12. *In cases where an infinite-horizon* `POMDP` *is approximated using a finite-horizon one (Remark 2.2.6), an* ϵ-`PWLC` *approximation of the convex optimal value function of the* `POMDP` *can be obtained by applying* H_{eff} *times Bellman's operator.*

Clearly, the convexity property of the optimal value function is of particular interest. It allows fairly good value generalization from visited beliefs to unvisited ones (compared to approximations of Lipschitz-continuous optimal value functions (Fehr et al. 2018)). In other words, visiting only certain branches of the game tree provides a "reasonable"[25] estimation of the value of any node. On the contrary, the algorithms presented up to now were "tabular", and consequently did not offer generalization properties. Below, we consider finite-horizon `b-MDP`s and present (i) an iterative construction of the representation of optimal value functions as envelopes of hyperplanes, and (ii) an `HSVI` scheme that leverages the structure exhibited above in Lemma 2.2.10.

[24] Usually, algorithmic schemes computing the α-vectors also compute a strategy whose value is at worst $b \cdot \alpha$ as a by-product. Algorithms either store the strategies using book-keeping or leave the strategy extraction as another step to do after convergence.

[25] "Reasonable" is to be understood empirically. The number of branches required to get an ϵ-close approximation of the optimal value of all nodes depends on both the problem and ϵ.

Iterative Construction of Optimal Value Functions Lemma 2.2.10 showed that optimal value functions of finite-horizon `b-MDP`s can be viewed as envelopes of hyperplanes. We present below an iterative construction of such hyperplanes, yielding an exact representation of optimal value functions in the whole space $\Delta(\mathcal{S})$, for all time steps.

The construction starts from the last time step and is based on the following proposition.

Proposition 2.2.13. *Let Γ_{t+1} be a set of α-vectors, such that $\forall \alpha$, $\forall b$, $\alpha \cdot b \leqslant V^*_{t+1}(b)$. Then one can construct*

$$\Gamma_t \overset{\text{def}}{=} \{\alpha : \mathcal{S} \to \mathbb{R} \ s.t., \quad \forall s \in \mathcal{S},\ \alpha(s) = r(s,a) + \sum_{\tilde{s},z} T(s,a,\tilde{s})\mathcal{O}(z,a,\tilde{s})\alpha_{a,z}(\tilde{s}) \mid \forall a \in \mathcal{A}, \forall (\alpha_{a,z})_{z\in\mathcal{Z}} \in (\Gamma_{t+1})^{|\mathcal{Z}|}\}. \tag{2.29}$$

Besides, it holds that $b \to \max_{\alpha_t} [b \cdot \alpha_t]$ lower bounds V^*_t in the whole space $\Delta(\mathcal{S})$.

At time step $H-1$, only the immediate reward is left to be optimized, so that one can easily compute a finite number of hyperplanes representing V^*_{H-1} exactly. An iterative application of Proposition 2.2.13 starting from time step $H-1$ until reaching time step 0 defines a dynamic programming scheme that is exact, but constructs a very large set of α-vectors in general, even when using pruning techniques. Indeed, it follows from Equation (2.29) that the number of created vectors at time step t is $|\Gamma_t| = |A| \cdot |\Gamma_{t+1}|^{|\mathcal{Z}|}$.

The previous result can be alternatively interpreted by reasoning upon all possible strategies. By induction, assume that each vector $\alpha_{a,z} \in \Gamma_{t+1}$ is associated to a strategy $\pi^{a,z}_{t+1:H-1}$ whose value in any belief b_{t+1} is at worst $\alpha_{a,z} \cdot b_{t+1}$. Strategies $\pi_{t:H-1}$ can be created by selecting (i) one action $a \in \mathcal{A}$ to make at time step t and a collection $(\pi^{a,z}_{t+1:H-1})_{z\in\mathcal{Z}}$ of strategies to follow from $t+1$ on, after observing each possible z. There are $|\mathcal{A}| \cdot |\Gamma_{t+1}|^{|\mathcal{Z}|}$ possible strategies $\pi_{t:H-1}$, each corresponding to one particular vector α_t and whose value is, at worst, $b_t \cdot \alpha_t$, for any belief b_t.

This iterative construction exactly computes the optimal value functions in the whole space $\Delta(\mathcal{S})$. On the contrary, `HSVI` (i) has access to the initial belief b_0 and (ii) only searches for an ϵ-close approximation at b_0. In particular, `HSVI` does not require exact computation of the optimal value functions in the whole space $\Delta(\mathcal{S})$, but only good enough approximations of "relevant" beliefs for b_0.

Heuristic Search Value Iteration The following presents new upper and lower bounds to replace the tabular ones in the `HSVI` scheme given in Algorithm 2.4.

The optimal value functions $(V^*_t)_{t\in\{0,\dots,H-1\}}$ being convex, at each time step t, a finite number of upper-bounding points allows constructing a convex hull that upper bounds V^*_t. New points are added by simply computing an upper-bounding value for each belief b_t encountered during trajectories, by applying Bellman's operator on $\overline{V}_{t+1}$:

$$\overline{V}_t(b_t) = \max_a \left[\rho(b,a) + \sum_z \Pr(z \mid b_t, a)\overline{V}_{t+1}(\tau(b_t,a,z)) \right]. \tag{2.30}$$

Remark 2.2.14. *In practice, however, computing the convex hull given a finite set of upper-bounding points is intractable as it involves solving `LP`s. Instead, we prefer upper-bounding techniques that are less precise, but way faster to compute, such as the* sawtooth approximation (Hauskrecht 2000; Smith 2007).

The construction of lower-bound approximations of optimal value functions is similar to the iterative construction presented in the previous paragraph. The key difference is that, at any belief b_t encountered during a trajectory, `HSVI` only adds one vector $\alpha_t \overset{\text{def}}{\in} \arg\max_{\alpha_t\in\Gamma_t} b_t \cdot \alpha_t$, where Γ_t is the set of α-vectors constructed in Proposition 2.2.13.

We significantly modified `HSVI`'s upper and lower bounds, so that it is not straightforward that the resulting version still possesses the convergence properties of the tabular one. Smith (2007) showed in his `Ph.D.` that it is the case, and even proposed criteria to decide whether some approximations are guaranteed not to break the convergence properties.

Theorem 2.2.15 (Convergence of HSVI (Smith 2007)). *The finite-time convergence of "tabular" HSVI (Algorithm 2.6) to an ϵ-optimum is maintained with the aforementioned upper and lower bounds.*

Remark 2.2.16 (Tools to Improve Efficiency). *Many tools improve the empirical efficiency of HSVI (e.g., smart initialization of lower and upper bounds, pruning, anytime versions), but we did not present them to maintain a concise discussion. The interested reader can find more details in Smith's Ph.D. dissertation.*

2.2.3.4 General Algorithmic Scheme: Summary of Specifications

Since several versions of the HSVI scheme were mentioned up to now (for MDPs, POMDPs and zs-SGs), Algorithm 2.7 provides a general implementation, while Table 2.3 specializes the relevant functions depending on the class of games considered. This table could be completed, but is not for the sake of simplicity, by adding columns for cp-POSGs (Dibangoye et al. 2016), his-POSGs (Chapter 4), zs-POSGs and subclasses (Horák et al. 2017; Horák et al. 2019b).

Algorithm 2.7: Generic HSVI for (infinite-horizon) problems

Input : s_0 a state
Fct Solve
  Initialize $\overline{V} : S \to \mathbb{R}$ with optimistic value
  Initialize $\underline{V} : S \to \mathbb{R}$ with pessimistic value
  while $\overline{V}(s_0) - \underline{V}(s_0) \geqslant \epsilon$ **do**
    Explore $(s_0, 0)$
  return $(\overline{V}, \underline{V})$
Fct Explore(s, t)
  if $\overline{V}(s) - \underline{V}(s) \geqslant \epsilon\gamma^{-t}$ **then**
    Update (s, t)
    $a^* \leftarrow$ Greedy$(\overline{V}, \underline{V}, s)$ // to be specified
    $s^* \leftarrow$ SelectMostUncertain$(\overline{V}, \underline{V}, s, a^*)$ // to be specified
    Explore (s^*)
    Update (s, t)
  return
Fct Update (s)
  Update $\overline{V}$ // to be specified
  Update $\underline{V}$ // to be specified

Table 2.3: Summary of specified procedures in the HSVI algorithmic scheme, for the different classes of games considered up to now.

Function	MDP	b-MDP	zs-SG
"state"	public state	belief	public state
"action"	$a \in \mathcal{A}$	$a \in \mathcal{A}$	$(\pi^1, \pi^2) \in \Delta(\mathcal{A}^1) \times \Delta(\mathcal{A}^2)$
Greedy	Equation (2.6) (page 18)	Equation (2.6) (page 18)	Equation (2.24) (page 34)
SelectMostUncertain	Equation (2.23) (page 32)	Equation (2.23) (page 32)	Equation (2.26) (page 34)
Update $\overline{V}$	Bellman$(\overline{V})$	Equation (2.30) (page 36)	Shapley$(\overline{V})$
Update $\underline{V}$	Bellman$(\underline{V})$	Proposition 2.2.13 (page 36)	Shapley$(\underline{V})$

2.2.3.5 POSGs

Solving zero-sum or common-payoff partially observable stochastic games through sufficient statistic was subject to recent extensive study (Dibangoye et al. 2016; Horák et al. 2017; Horák

et al. 2019b; Brown et al. 2020; Sokota et al. 2023; Vojtěch Kovařík et al. 2022b). As we shall see in the following, the interests are threefold. Firstly, it enables (after having proven that Bellman's optimality principle holds) tools from the fully observable settings to apply. Secondly, it allows taking advantage of continuity properties of the optimal value function in the continuous state space. Finally, lossy or lossless compression of the state to reduce its dimensionality highly improves the solving algorithms' scalability with respect to (i) players' numbers of actions and observations and (ii) the planning horizon.

Continuous-state Markov games (including continuous-state MDPs) As often in assumptions for multi-player planning under partial observability (Wiggers et al. 2016a; Dibangoye et al. 2016), let us formally define an *occupancy state* (OS) $\sigma_{\boldsymbol{\beta}_{0:\tau-1}}$ as the probability distribution over joint AOHs $\boldsymbol{\theta}_\tau$ given partial behavioral strategy profile $\boldsymbol{\beta}_{0:\tau-1}$. This statistic exhibits the usual Markov and sufficiency properties:

Proposition 2.2.17 (Adapted from Dibangoye et al. 2016, Thm. 1)**.** *Let G be a POSG. In G, $\sigma_{\boldsymbol{\beta}_{0:\tau-1}}$, together with $\boldsymbol{\beta}_\tau$, is a sufficient statistic to compute:*

1. *the next OS, $T(\sigma_{\boldsymbol{\beta}_{0:\tau-1}}, \boldsymbol{\beta}_\tau) \overset{\text{def}}{=} \sigma_{\boldsymbol{\beta}_{0:\tau}}$, and*
2. *the expected reward at τ for any player:*

$$\forall i, r^i(\sigma_{\boldsymbol{\beta}_{0:\tau-1}}, \boldsymbol{\beta}_\tau) \overset{\text{def}}{=} \mathbb{E}\left[r^i(S_\tau, A_\tau) \mid \boldsymbol{\beta}_{0:\tau-1} \oplus \boldsymbol{\beta}_\tau\right], \tag{2.31}$$

where $\oplus$ denotes a concatenation.

Proof. Let us first derive a recursive way of computing $\sigma_{\boldsymbol{\beta}_{0:\tau}}(\boldsymbol{\theta}_\tau \oplus \boldsymbol{a}_\tau \oplus \boldsymbol{z}_{\tau+1})$:

$$\begin{aligned}
&\sigma_{\boldsymbol{\beta}_{0:\tau}}(\boldsymbol{\theta}_\tau, \boldsymbol{a}_\tau, \boldsymbol{z}_{\tau+1}) \overset{\text{def}}{=} Pr(\boldsymbol{\theta}_\tau, \boldsymbol{a}_\tau, \boldsymbol{z}_{\tau+1} \mid \boldsymbol{\beta}_{0:\tau}) && (2.32)\\
&= \sum_{s_\tau, s_{\tau+1}} Pr(\boldsymbol{\theta}_\tau, \boldsymbol{a}_\tau, \boldsymbol{z}_{\tau+1}, s_\tau, s_{\tau+1} \mid \boldsymbol{\beta}_{0:\tau}) && (2.33)\\
&= \sum_{s_\tau, s_{\tau+1}} Pr(\boldsymbol{z}_{\tau+1}, s_{\tau+1} \mid \boldsymbol{\theta}_\tau, \boldsymbol{a}_\tau, s_\tau, \boldsymbol{\beta}_{0:\tau}) Pr(\boldsymbol{a}_\tau \mid \boldsymbol{\theta}_\tau, s_\tau, \boldsymbol{\beta}_{0:\tau}) Pr(s_\tau \mid \boldsymbol{\theta}_\tau, \boldsymbol{\beta}_{0:\tau}) Pr(\boldsymbol{\theta}_\tau \mid \boldsymbol{\beta}_{0:\tau}) && (2.34)\\
&= \sum_{s_\tau, s_{\tau+1}} \underbrace{Pr(\boldsymbol{z}_{\tau+1}, s_{\tau+1} \mid \boldsymbol{a}_\tau, s_\tau)}_{=P_{\boldsymbol{a}_\tau}^{\boldsymbol{z}_{\tau+1}}(s_{\tau+1} \mid s_\tau)} \underbrace{Pr(\boldsymbol{a}_\tau \mid \boldsymbol{\theta}_\tau, \boldsymbol{\beta}_\tau)}_{=\boldsymbol{\beta}(\boldsymbol{\theta}_\tau, \boldsymbol{a}_\tau)} \underbrace{Pr(s_\tau \mid \boldsymbol{\theta}_\tau, \boldsymbol{\beta}_{0:\tau})}_{=b(s_\tau \mid \boldsymbol{\theta}_\tau)} \underbrace{Pr(\boldsymbol{\theta}_\tau \mid \boldsymbol{\beta}_{0:\tau-1})}_{=\sigma_{\boldsymbol{\beta}_{0:\tau-1}}(\boldsymbol{\theta}_\tau)}, && (2.35)
\end{aligned}$$

(where $b(s \mid \boldsymbol{\theta}_\tau)$ is the belief over states obtained by a usual HMM filtering process)

$$= \sum_{s_\tau, s_{\tau+1}} P_{\boldsymbol{a}_\tau}^{\boldsymbol{z}_{\tau+1}}(s_{\tau+1} \mid s_\tau) \boldsymbol{\beta}(\boldsymbol{\theta}_\tau, \boldsymbol{a}_\tau) b(s_\tau \mid \boldsymbol{\theta}_\tau) \sigma_{\boldsymbol{\beta}_{0:\tau-1}}(\boldsymbol{\theta}_\tau). \tag{2.36}$$

$\sigma_{\boldsymbol{\beta}_{0:\tau}}$ can thus be computed from $\sigma_{\boldsymbol{\beta}_{0:\tau-1}}$ and $\boldsymbol{\beta}_\tau$ without explicitly using $\boldsymbol{\beta}_{0:\tau-1}$ or earlier occupancy states.

Then, let us compute player i's expected reward at τ given $\boldsymbol{\beta}_{0:\tau}$:

$$\begin{aligned}
&E[r^i(S_\tau, A_\tau^1, A_\tau^2) \mid \boldsymbol{\beta}_{0:\tau}] && (2.37)\\
&= \sum_{s_\tau, \boldsymbol{a}_\tau} r^i(s_\tau, \boldsymbol{a}_\tau) Pr(s_\tau, \boldsymbol{a}_\tau \mid \boldsymbol{\beta}_{0:\tau}) && (2.38)\\
&= \sum_{s_\tau, \boldsymbol{a}_\tau} \sum_{\boldsymbol{\theta}_\tau} r^i(s_\tau, \boldsymbol{a}_\tau) Pr(s_\tau, \boldsymbol{a}_\tau, \boldsymbol{\theta}_\tau \mid \boldsymbol{\beta}_{0:\tau}) && (2.39)\\
&= \sum_{s_\tau, \boldsymbol{a}_\tau} \sum_{\boldsymbol{\theta}_\tau} r^i(s_\tau, \boldsymbol{a}_\tau) Pr(s_\tau, \boldsymbol{a}_\tau \mid \boldsymbol{\theta}_\tau, \boldsymbol{\beta}_{0:\tau}) Pr(\boldsymbol{\theta}_\tau \mid \boldsymbol{\beta}_{0:\tau}) && (2.40)\\
&= \sum_{s_\tau, \boldsymbol{a}_\tau} \sum_{\boldsymbol{\theta}_\tau} r^i(s_\tau, \boldsymbol{a}_\tau) \underbrace{Pr(\boldsymbol{a}_\tau \mid \boldsymbol{\theta}_\tau, \boldsymbol{\beta}_{0:\tau})}_{\boldsymbol{\beta}_\tau(\boldsymbol{\theta}_\tau, \boldsymbol{a}_\tau)} \underbrace{Pr(s_\tau \mid \boldsymbol{\theta}_\tau, \boldsymbol{\beta}_{0:\tau})}_{b(s_\tau \mid \boldsymbol{\theta}_\tau)} \underbrace{Pr(\boldsymbol{\theta}_\tau \mid \boldsymbol{\beta}_{0:\tau})}_{\sigma_{\boldsymbol{\beta}_{0:\tau-1}}(\boldsymbol{\theta}_\tau)} && (2.41)
\end{aligned}$$

$$= \sum_{s_\tau, \boldsymbol{a}_\tau} \sum_{\boldsymbol{\theta}_\tau} r^i(s_\tau, \boldsymbol{a}_\tau) \boldsymbol{\beta}_\tau(\boldsymbol{\theta}_\tau, \boldsymbol{a}_\tau) b(s_\tau \mid \boldsymbol{\theta}_\tau) \sigma_{\boldsymbol{\beta}_{0:\tau-1}}(\boldsymbol{\theta}_\tau). \tag{2.42}$$

Player i's expected reward at τ can thus be computed from $\sigma_{\boldsymbol{\beta}_{0:\tau-1}}$ and $\boldsymbol{\beta}_\tau$ without explicitly using $\boldsymbol{\beta}_{0:\tau-1}$ or earlier occupancy states. □

From now on, we will write σ_τ as short for $\sigma_{\boldsymbol{\beta}_{0:\tau-1}}$, the OS associated with some prefix strategy profile $\boldsymbol{\beta}_{0:\tau-1}$.

Occupancy Markov Games We can then derive, from a POSG, the game induced by the Markov process of occupancy-states.

Definition 2.2.18 (Occupancy Markov Games (oMGs))**.** *An* occupancy Markov game *(oMG)*[26] *is defined by the tuple* $\langle n, \mathcal{O}^\sigma, \mathcal{B}, T, r, H, \gamma \rangle$*, where:*

- n *is the number of players;*
- $\mathcal{O}^\sigma (= \cup_{t=0}^{H-1} \mathcal{O}_t^\sigma)$ *is the set of* OSs *induced by the* POSG*;*
- $\mathcal{B}$ *is the set of decision rule profiles of the* POSG*;*
- T *is the deterministic transition function as defined in Equation* (2.32)*;*
- $\forall i,\ r^i$ *is the reward function of player* i *as defined in Equation* (2.42)*; and*
- H *and* γ *are as in the* POSG*.*

Note that b_0 *is not in the tuple but serves to define* T *and* r^i *for all players* i*.*

The following discussion discards the case $n = 1$ that corresponds to a POMDP, which is rather different as the occupancy state (*i.e.,* the belief state (Definition 2.2.7)) is known to the player at execution phase. On the contrary, players in a POSG do not observe the occupancy state at execution phase as it depends on their opponent's strategies, which remains unknown in general. Consequently, to ensure the equivalence between the original POSG and the derived oMG, we follow similar path to Oliehoek (2013) and define the Markov process induced by occupancy states as a non-observable problem.

In both common-payoff and zero-sum cases, the game being deterministic implies that there exists a solution that can be executed in open-loop, *i.e.,* that consists in a sequence of decision rule profiles. Consequently, any Nash equilibrium strategy profile of the derived oMG can be translated into a Nash equilibrium strategy profile of the original zero-sum or common-payoff POSG. But this equivalence comes at a price: solving a non-observable zero-sum or common-payoff Markov game is non-standard and we do not know yet if Bellman's optimality principle applies to such game. In particular, both following questions are worth answering:

- Does Bellman's optimality principle apply to the computation of optimal value functions?
- Does Bellman's optimality principle apply to the computation of Nash equilibrium strategy profiles?

Despite occupancy states not being accessible to players during execution phase, let us define a subgame by

- a time step $\tau \in \{0, \dots, H-1\}$;
- an occupancy state σ_τ;
- the objective of finding a Nash equilibrium strategy profile $\boldsymbol{\beta}_{\tau:H-1}$ for the criterion

$$\mathbb{E}\left[\sum_{t=\tau}^{H-1} \gamma^{t-\tau} r(S_t, A_t) \mid \sigma_\tau, \boldsymbol{\beta}_{\tau:H-1} \right].$$

Answers to the previous questions for cp-POSGs exist in the literature (Dibangoye et al. 2016), while Chapter 3 of this Ph. D. provides positive answers for the zs-POSG case.

[26] We use (i) "Markov game" instead of "stochastic game" because the dynamics are not stochastic, and (ii) "partially observable stochastic game" to stick with the literature.

`cp-POSGs` The value of a common-payoff occupancy Markov game is

$$\max_{\beta^1_{0:H-1}} \cdots \max_{\beta^n_{0:H-1}} \mathbb{E}\left[r(S_t, A_t) \mid \boldsymbol{\beta}_{0:H-1}\right]. \tag{2.43}$$

The problem is a nesting of only max operators. Bellman's optimality principle "easily" applies to the computation of the `NEV` of the game. Besides, the concatenation of the arg max for each time step is an optimal solution, so that Bellman's optimality principle also applies to the computation of a `NES` with the highest value. At any time step, it holds that:

$$V^*(\sigma_\tau) = \max_{\boldsymbol{\beta}_\tau} r(\sigma_\tau, \boldsymbol{\beta}_\tau) + \gamma V^*_{\tau+1}(T(\sigma_\tau, \boldsymbol{\beta}_\tau)), \tag{2.44}$$

and a `NES` with highest value for the subgame rooted at σ_τ can be obtained:

$$\boldsymbol{\beta}^*_\tau \oplus \boldsymbol{\beta}^*_{\tau+1:} \in \arg\max_{\boldsymbol{\beta}_{\tau:}} \mathbb{E}\left[\sum_{t=\tau}^{H-1} \gamma^{t-\tau} r(S_t, A_t) \mid \sigma_\tau, \boldsymbol{\beta}_{\tau:}\right], \tag{2.45}$$

where:

$$\boldsymbol{\beta}^*_\tau \in \arg\max_{\boldsymbol{\beta}_\tau} \mathbb{E}\left[r(\sigma_\tau, \boldsymbol{\beta}_\tau) + \gamma V^*_{\tau+1}(T(\sigma_\tau, \boldsymbol{\beta}_\tau))\right], \qquad \text{and} \tag{2.46}$$

$$\boldsymbol{\beta}^*_{\tau+1:} \in \arg\max_{\boldsymbol{\beta}_{\tau+1:}} \mathbb{E}\left[\sum_{t=\tau+1}^{H-1} \gamma^{t-(\tau+1)} r(S_t, A_t) \mid T(\sigma_\tau, \boldsymbol{\beta}^*_\tau), \boldsymbol{\beta}_{\tau+1:}\right]. \tag{2.47}$$

Note that $\boldsymbol{\beta}^*_{\tau+1:}$ is to be obtained applying again Bellman's optimality principle, until reaching $\tau = H$.

`zs-POSGs` The value of a zero-sum occupancy Markov game is

$$\max_{\beta^1_{0:H-1}} \min_{\beta^2_{0:H-1}} \mathbb{E}\left[r(S_t, A_t) \mid \boldsymbol{\beta}_{0:H-1}\right]. \tag{2.48}$$

The optimization process incorporates both max operators and min operators. As a consequence, its is not straightforward to understand whether Bellman's optimality principle applies to the computation of the `NEV`. Even though we give in Chapter 3 a positive answer to the latter question, the concatenation of solutions at each time step has no reason to be a Nash equilibrium strategy profile for the whole game. In other words, a classic dynamic programming algorithm would not be able to compute a solution of the `zs-POSG`, in general. In the matching pennies game for example, 1's unique Nash equilibrium strategy is to play head or tail with equal probability 0.5 at time step 0 and any strategy thereafter. But then, any strategy (including deterministic ones) of player 2 is a `NES` for the subgame reached at $t = 1$. Consequently, concatenating strategy for 2 at time steps 0 (at time step 0, 2's strategies are irrelevant) and 1 can yield arbitrarily bad strategies $\beta^2_{0:1}$ for 2.

What If We Assume Full Observability? If the `zs-oMG` is considered to be fully observable, then the answers to the applicability or not of Bellman's optimality principle to compute `NEV` and `NES`s differ. The game resembles a deterministic `zs-SG` (but with infinite state space, and infinite action space), so that Bellman's optimality principle might apply both for the computation of the `NEV` and Nash equilibrium strategy profiles. But this comes at a price: the ability to retrieve a solution to the `zs-POSG` is unclear. Let us assume only for the discussion below that the occupancy state is publicly available to players. Nash equilibrium strategy profiles for the matching pennies game satisfy for time step 0:

$$\beta^1_0(b_0) = 0.5\delta_{a_h} + 0.5\delta_{a_t}, \qquad \text{and} \tag{2.49}$$

$$\beta^2_0(b_0) = *, \tag{2.50}$$

and for time step 1:

$$\beta_1^1(*) = *, \qquad \text{and} \tag{2.51}$$

$$\beta_1^2(0.5\delta_{s_h} + 0.5\delta_{s_t}) = * \tag{2.52}$$

$$\beta_1^2(p\delta_{s_h} + (1-p)\delta_{s_t}) = a_t \qquad \text{if } p < 0.5, \tag{2.53}$$

$$\beta_1^2(p\delta_{s_h} + (1-p)\delta_{s_t}) = a_h \qquad \text{if } p > 0.5. \tag{2.54}$$

In particular, as in the previous paragraph, player 2 at time step 1 can play anything if player 1's strategy at time step 0 is uniformly random. Then, it is non trivial to retrieve a solution to the original `zs-POSG`.

3

Solving zs-POSGs Through Dynamic Programming

Contents

We now present the contribution of the manuscript dealing with the computation of an ϵ-Nash equilibrium strategy profile for any zero-sum partially observable stochastic game, through a dynamic programming approach. In the first section (Section 3.1), we start with general results on zero-sum occupancy Markov games that suggest a potential applicability of Bellman's optimality principle to solve them. Then, we build on these results to design necessary tools (*e.g.,* approximation functions, backup and update operators) that permit the implementation of an HSVI scheme (Section 3.1.2). Finally, the resulting algorithm is described; its convergence properties are discussed (Section 3.1.3); and experimental validations are presented (Section 3.2).

Note: To help the reader, Appendix A.1 provides three synthetic tables: Table A.1 (p. 120) to sum up various theoretical properties (assuming a finite temporal horizon), Table A.1 (p. 120) and Table A.2 (p. 122) to respectively sum up the notation and the abbreviations used in this chapter.

Also, for convenience, we may replace in the following:

- *subscript "$\tau : H-1$" with "$\tau :$",*
- *any function $f(\boldsymbol{x})$ linear in vector $\boldsymbol{x}$ with either $f(\cdot)\cdot\boldsymbol{x}$ or $\boldsymbol{x}^\top\cdot f(\cdot)$,*
- *a full tuple with its few elements of interest, and*
- *an element (a "field") x of a specific tuple t by $x[t]$.*

3.1 Theoretical Contributions

3.1.1 Properties of zs-oMGs

This first section of the theoretical contributions provides mandatory results for the understanding of our solving algorithm. We study oMGs induced by the occupancy-state Markov process (Section 2.2.3.5). These games are shown to exhibit interesting properties, especially regarding the ability to apply dynamic programming (Theorems 3.1.5 and 3.1.6). Finally, technical continuity results for the transition functions of the Markov process are given (Lemmas 3.1.8 to 3.1.10).

3.1.1.1 "Subgames" and their properties

Until stated otherwise (*i.e.,* until Section 3.1.1.1), the discussion below considers general-sum oMGs with any number of players. Let us recall that an oMG was defined in Definition 2.2.18 (page 39) and that the state of the game σ_τ is hidden to players. Let us recall that oMGs and their attached notion of subgames were defined in Definition 2.2.18 definition 2.2.18) and that the state of the game σ_τ is hidden to players.

Despite the OS at $\tau > 0$ not being accessible to any player, let us recall that we defined (definition 2.2.18) a *subgame* at σ_τ to be the restriction to reachable (*i.e.,* with non-zero probability) individual histories, starting from time step τ under this particular occupancy state, meaning that we are seeking strategies $\beta^1_{\tau:H-1}$ and $\beta^2_{\tau:H-1}$. σ_τ tells us which AOHs each player could be facing with non-zero probability, and are thus relevant for planning. We can then define the value function in any OS σ_τ for any strategy profile $\boldsymbol{\beta}_{\tau:H-1}$.

Definition 3.1.1 (Value of Strategy Profile)**.** *The value of any strategy profile $\boldsymbol{\beta}_{\tau:H-1}$ for any occupancy-state σ_τ is defined as*

$$V_\tau(\sigma_\tau, \boldsymbol{\beta}_{\tau:H-1}) \stackrel{\text{def}}{=} E[\sum_{t=\tau}^{H-1} \gamma^{t-\tau} r(S_t, A_t) | \sigma_\tau, \boldsymbol{\beta}_{\tau:H-1}]. \tag{3.1}$$

However, it is not clear here if a Nash equilibrium for the previous criterion can be easily characterized as a minmax value as in Theorem 2.1.11. Indeed, the previous criterion involving behavioral strategies, it is not bilinear in players' strategies and thus does not define a normal-form game.

Proposition 3.1.2 (Value Functions $V_\tau(\sigma_\tau, \boldsymbol{\beta}_{\tau:H-1})$ are not Linear in Individual Behavioral Strategies)**.** *There exists a zs-POSG such that, for some time step τ, $(\beta^1_{\tau:}, \beta^2_{\tau:}) \mapsto V_\tau(\sigma_\tau, \boldsymbol{\beta}_{\tau:H-1})$ is not linear with respect to either $\beta^1_{\tau:H-1}$ or $\beta^2_{\tau:H-1}$.*

Proof. Let us consider the following (finite-horizon, deterministic) non-observable MDP (modeled as a POMDP):

$$\begin{aligned}
\mathcal{S} &\stackrel{\text{def}}{=} \{-2, -1, 0, +1, +2\}, \quad b_0(0) = 1, && \text{(always start in } s = 0) \\
\mathcal{A} &\stackrel{\text{def}}{=} \{-1, +1\}, && \text{(moves = add or subtract 1)} \\
T(s, a) &\stackrel{\text{def}}{=} \min\{+2, \max\{-2, s + a\}\}, && \text{(+1 or } -1 \text{ move in } \mathcal{S}) \\
\mathcal{Z} &\stackrel{\text{def}}{=} \{none\}, \quad O(none) \stackrel{\text{def}}{=} 1, && \text{(no observation)} \\
r(s) &\stackrel{\text{def}}{=} \begin{cases} +1 & \text{if } s \in \{-2, +2\} \\ 0 & \text{otherwise,} \end{cases} && (|s| = 2 : \text{success!}) \\
\gamma &\stackrel{\text{def}}{=} 1, \quad H \stackrel{\text{def}}{=} 2.
\end{aligned}$$

In this game, a single player moves either to the left or to the right at each time step along a finite 1D line, aiming to reach one of the two extremities.

Let us then consider two particular behavioral strategies:

$$\forall \theta, \ \beta^+(A = +1 | \theta) = 1 \qquad \text{(always +1), and}$$

$$\forall\theta,\ \beta^-(A=-1|\theta)=1 \qquad \text{(always } -1\text{)}.$$

These two strategies are optimal, with an expected return of $+1$, because, at $t=H=2$, β^+ reaches $+2$ w.p. 1, and β^- reaches -2 w.p. 1:

$$V(\beta^+)=V(\beta^-)=+1.$$

Let us now consider their linear combination $\beta^\pm \overset{\text{def}}{=} \frac{1}{2}\beta^+ + \frac{1}{2}\beta^-$:

$$\forall\theta,\ \beta^\pm(A=-1|\theta)=0.5,$$
$$\forall\theta,\ \beta^\pm(A=+1|\theta)=0.5.$$

Here, the probability to reach $s=-2$ or $s=+2$ at the last time step is much lower, and gives the value of that strategy:

$$\begin{aligned}V(\beta^\pm) &= Pr(s_2=+2|\beta^\pm)+Pr(s_2=-2|\beta^\pm)\\ &= Pr(a_0=+1|\beta^\pm)\cdot Pr(a_1=+1|\beta^\pm)\\ &+ Pr(a_0=-1|\beta^\pm)\cdot Pr(a_1=-1|\beta^\pm)\\ &= \underbrace{0.5\cdot 0.5}_{0.25}+\underbrace{0.5\cdot 0.5}_{0.25}=0.5.\end{aligned}$$

This confirms that, in a `POMDP`, the value is not linear in the space of behavioral strategies. As a consequence, in a `zs-POSG`, the value is not (bi)linear in the spaces of behavioral strategies of both players. □

The proof of the minimax theorem only applies to games whose payoff functions exhibit bilinearity w.r.t. players' strategies. As a consequence, the previous proposition exhibits a potential barrier to the relevance of the aforementioned subgames. Indeed, establishing the unicity of subgames' value (*i.e.,* showing that $\max-\min$ equals $\min-\max$ with behavioral strategies) can reasonably be expected to be key to approaches combining dynamic programming and heuristic search to solve `zs-POSG`s. When considering dynamic games, bilinearity of payoff functions typically holds w.r.t. mixed strategies. Yet, the concept of mixed strategies "rooted" at some subgames defined in particular by a distribution probability over players' history profiles does not exist, to the best of our knowledge.

The following fills that gap by extending (i) the definition of mixed strategies to mixed strategies "compatible" with an associated occupancy state σ_τ, and (ii) Kuhn's equivalence theorem between mixed strategies and behavioral ones.

Back to Mixed Strategies We now generalize mixed strategies as a mathematical tool to handle subgames of a `zs-oMG` as normal-form games, and give some preliminary results.

First, for a given σ_τ and $\tau \leqslant \tau'$, let $\boldsymbol{\mu}_{0:\tau'-1|\sigma_\tau\rangle}$ denote a mixed strategy profile that is defined over $0:\tau'-1$, and induces σ_τ at time τ. We will also write that this strategy is *compatible* with σ_τ.

From now on, we consider that either $\tau'=\tau$ or $\tau'=H$.

To complete a given mixed *prefix* strategy $\boldsymbol{\mu}_{0:\tau-1|\sigma_\tau\rangle}$, the solver should provide each player with a different *suffix* strategy to execute for each θ^i_τ it could be facing. We now detail how to build an equivalent set of mixed *full* strategies for i. Each of the pure *prefix* strategies $\pi^i_{0:\tau-1}$ used in $\mu^i_{0:\tau-1|\sigma_\tau\rangle}$ (belonging to a set denoted $\Pi^i_{0:\tau-1|\sigma_\tau\rangle}$) can be extended by appending a different pure *suffix* strategy $\pi^i_{\tau:H-1}$ at each of its leaf nodes, which leads to a large set of pure strategies $\Pi^i_{0:H-1}(\pi^i_{0:\tau-1})$. Then, let $M^i_{0:H-1|\sigma_\tau\rangle}$ be the set of mixed *full* strategies $\mu^i_{0:H-1|\sigma_\tau\rangle}$ obtained by considering the distributions over $\bigcup_{\pi^i_{0:\tau-1}\in\Pi^i_{0:\tau-1|\sigma_\tau\rangle}} \Pi^i_{0:H-1}(\pi^i_{0:\tau-1})$ that verify, $\forall\pi^i_{0:\tau-1}$,

$$\sum_{\substack{\pi^i_{0:H-1}\in\\ \Pi^i_{0:H-1}(\pi^i_{0:\tau-1})}} \mu^i_{0:H-1|\sigma_\tau\rangle}(\pi^i_{0:H-1}) = \mu^i_{0:\tau-1|\sigma_\tau\rangle}(\pi^i_{0:\tau-1}). \tag{3.2}$$

Note that, even though $M^i_{0:H-1|\sigma_\tau\rangle}$ is bound to a particular prefix mixed strategy, this arbitrary choice is not limiting for the following discussion.

Lemma 3.1.3. *The set $M^i_{0:H-1|\sigma_\tau\rangle}$ is convex.*

Proof. Let $\mu^i_{0:H-1|\sigma_\tau\rangle}$ and $\nu^i_{0:H-1|\sigma_\tau\rangle}$ be two mixed strategies in $M^i_{0:H-1|\sigma_\tau\rangle}$, *i.e.,* which are both full and compatible with occupancy state σ_τ at time step τ, and let $\alpha \in [0,1]$. Then, for any $\pi^i_{0:\tau-1}$,

$$\sum_{\pi^i_{0:H-1}\in\Pi^i_{0:H-1}(\pi^i_{0:\tau-1})} \left[\alpha\cdot\mu^i_{0:H-1|\sigma_\tau\rangle}(\pi^i_{0:H-1}) + (1-\alpha)\cdot\nu^i_{0:H-1|\sigma_\tau\rangle}(\pi^i_{0:H-1})\right]$$
$$= \alpha\left[\sum_{\pi^i_{0:H-1}\in\Pi^i_{0:H-1}(\pi^i_{0:\tau-1})} \mu^i_{0:H-1|\sigma_\tau\rangle}(\pi^i_{0:H-1})\right] + (1-\alpha)\left[\sum_{\pi^i_{0:H-1}\in\Pi^i_{0:H-1}(\pi^i_{0:\tau-1})} \nu^i_{0:H-1|\sigma_\tau\rangle}(\pi^i_{0:H-1})\right]$$

(because both mixed strategies are compatible with σ_τ (eq. 3.2, p. 44):)

$$= \alpha\cdot\mu^i_{0:\tau-1|\sigma_\tau\rangle}(\pi^i_{0:\tau-1}) + (1-\alpha)\cdot\mu^i_{0:\tau-1|\sigma_\tau\rangle}(\pi^i_{0:\tau-1})$$
$$= \mu^i_{0:\tau-1|\sigma_\tau\rangle}(\pi^i_{0:\tau-1}).$$

Eq. 3.2 thus also applies to $\alpha\cdot\mu^i_{0:H-1|\sigma_\tau\rangle}+(1-\alpha)\cdot\nu^i_{0:H-1|\sigma_\tau\rangle}$, proving that it belongs to $M^i_{0:H-1|\sigma_\tau\rangle}$ and, as a consequence, that this set is convex.

□

Corollary 3.1.4 (Equivalence between behavioral and mixed strategies)**.** *The set $M^i_{0:H-1|\sigma_\tau\rangle}$ is equivalent to the set of behavioral strategies $\beta^i_{0:H-1|\sigma_\tau\rangle}$, and is thus sufficient to search for a Nash equilibrium strategy profile in σ_τ.*

Proof. The equivalence with the set of behavioral strategies simply relies on the fact that all mixed strategies over $\tau : H-1$ can be independently generated at each action-observation history $\theta^i_{0:\tau-1}$. □

While only future rewards are relevant when making a decision at τ, reasoning with mixed strategies defined from $t=0$ will be convenient because $V_\tau(\sigma_\tau,\cdot,\cdot)$ is linear in $\mu^i_{0:H-1|\sigma_\tau\rangle}$, which allows coming back to a standard normal-form game and applying known results.

In the remainder, we simply denote μ^i (without index) the mixed strategies in $M^i_{0:H-1|\sigma_\tau\rangle}$, set which we now denote $M^i_{|\sigma_\tau\rangle}$. Also, since we shall work with local game $Q^*_\tau(\sigma_\tau,\boldsymbol{\beta}_\tau)$, let us define $M^i_{|\sigma_\tau,\beta^j_\tau\rangle}$ the set of i's mixed strategies compatible with occupancy states reachable given σ_τ and β^j_τ (with either $j=i$ or $j=\neg i$). Then, $M^i_{|\sigma_\tau,\boldsymbol{\beta}_\tau\rangle} \subseteq M^i_{|\sigma_\tau,\beta^j_\tau\rangle} \subseteq M^i_{|\sigma_\tau\rangle}$ (inclusion relying on the perfect recall hypothesis and the latter sets being less constrained in their definition). As a consequence, if maximizing some function f over i's mixed strategies compatible with a given σ_τ:

$$\max_{\mu^i\in M^i_{|\sigma_\tau\rangle}} f(\sigma_\tau,\mu^i,\dots) \geqslant \max_{\mu^i\in M^i_{|\sigma_\tau,\beta^j_\tau\rangle}} f(\sigma_\tau,\mu^i,\dots) \geqslant \max_{\mu^i\in M^i_{|\sigma_\tau,\boldsymbol{\beta}_\tau\rangle}} f(\sigma_\tau,\mu^i,\dots).$$

Von Neumann's Minimax Theorem for Subgames From now on, we focus on zero-sum games (with only two players).

Using the previous results, one can show that von Neumann's minimax theorem applies in any subgame, allowing to swap max and min operators.

Theorem 3.1.5 (Minimax theorem)**.** *The subgame defined in Eq. (3.1) admits a unique* ***NEV***

$$V^*_\tau(\sigma_\tau) \overset{\text{def}}{=} \max_{\beta^1_{\tau:H-1}} \min_{\beta^2_{\tau:H-1}} V_\tau(\sigma_\tau,\beta^1_{\tau:H-1},\beta^2_{\tau:H-1}) = \min_{\beta^2_{\tau:H-1}} \max_{\beta^1_{\tau:H-1}} V_\tau(\sigma_\tau,\beta^1_{\tau:H-1},\beta^2_{\tau:H-1}). \tag{3.3}$$

Proof. For any occupancy state σ_τ,

$$\max_{\beta^1_{\tau:}}\min_{\beta^2_{\tau:}} V(\sigma_\tau,\boldsymbol{\beta}_{\tau:}) = \max_{\mu^1_{\tau:}}\min_{\mu^2_{\tau:}} V(\sigma_\tau,\boldsymbol{\mu}_{\tau:}) \qquad \text{(Kuhn's theorem (generalized))} \tag{3.4}$$

$$= \min_{\mu^2_{\tau:}} \max_{\mu^1_{\tau:}} V(\sigma_\tau, \boldsymbol{\mu}_{\tau:}) \qquad \text{(von Neumann's theorem)} \tag{3.5}$$

$$= \min_{\beta^2_{\tau:}} \max_{\beta^1_{\tau:}} V(\sigma_\tau, \boldsymbol{\beta}_{\tau:}). \qquad \text{(again Kuhn's theorem (generalized))} \tag{3.6}$$

□

3.1.1.2 Bellman's Optimality Principle; a Recursive Expression of V^*

In this occupancy Markov game, the state of the game is unknown to players (*i.e.,* no player has access to σ_τ). Therefore, `zs-oMG`s are different from `zs-SG`s and the usual result stating Bellman's optimality equations for `zs-SG`s might not be straighforwardly applicable. Still, we now show that it does hold, which justifies reasoning on subgames despite the non-observability as the value of a subgame is related to the value of its nested subgames.

Theorem 3.1.6 (Bellman optimality equation)**.** *$V^*_\tau(\sigma_\tau)$ satisfies the following functional equation:*

$$V^*_\tau(\sigma_\tau) = \max_{\beta^1_\tau} \min_{\beta^2_\tau} \left[r(\sigma_\tau, \boldsymbol{\beta}_\tau) + \gamma V^*_{\tau+1}(T(\sigma_\tau, \boldsymbol{\beta}_\tau)) \right].$$

Proof. Focusing, without loss of generality, on player 1, we have (complementary explanations follow for numbered lines in particular):

$$\max_{\beta^1_\tau} \min_{\beta^2_\tau} \left[r(\sigma_\tau, \beta^1_\tau, \beta^2_\tau) + \gamma V^*_{\tau+1}(T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau)) \right]$$

($V^*_{\tau+1}(T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau))$ being the Nash equilibrium value of normal-form game $V_{\tau+1}(T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau), \mu^1, \mu^2)$:)

$$= \max_{\beta^1_\tau} \min_{\beta^2_\tau} \left[r(\sigma_\tau, \beta^1_\tau, \beta^2_\tau) + \gamma \max_{\mu^1 \in M^1_{|\sigma_\tau, \boldsymbol{\beta}_\tau\rangle}} \min_{\mu^2 \in M^2_{|\sigma_\tau, \boldsymbol{\beta}_\tau\rangle}} V_{\tau+1}(T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau), \mu^1, \mu^2) \right]$$

$$= \max_{\beta^1_\tau} \min_{\beta^2_\tau} \max_{\mu^1 \in M^1_{|\sigma_\tau, \boldsymbol{\beta}_\tau\rangle}} \min_{\mu^2 \in M^2_{|\sigma_\tau, \boldsymbol{\beta}_\tau\rangle}} \left[r(\sigma_\tau, \beta^1_\tau, \beta^2_\tau) + \gamma V_{\tau+1}(T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau), \mu^1, \mu^2) \right]$$

(using the equivalence between maximin and minimax values for the (constrained normal-form) game at $\tau + 1$, the last two max and min operators can be swapped:)

$$= \max_{\beta^1_\tau} \min_{\beta^2_\tau} \min_{\mu^2 \in M^2_{|\sigma_\tau, \boldsymbol{\beta}_\tau\rangle}} \max_{\mu^1 \in M^1_{|\sigma_\tau, \boldsymbol{\beta}_\tau\rangle}} \left[r(\sigma_\tau, \beta^1_\tau, \beta^2_\tau) + \gamma V_{\tau+1}(T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau), \mu^1, \mu^2) \right]$$

(merging both mins (and with explanations thereafter):)

$$= \max_{\beta^1_\tau} \min_{\mu^2 \in M^2_{|\sigma_\tau, \beta^1_\tau\rangle}} \max_{\mu^1 \in M^1_{|\sigma_\tau, \beta^1_\tau, \beta^2_\tau(\mu^2)\rangle}} \left[r(\sigma_\tau, \beta^1_\tau, \beta^2_\tau(\mu_2)) + \gamma V_{\tau+1}(T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau(\mu_2)), \mu^1, \mu^2) \right] \tag{3.7}$$

(since ignoring the opponent's decision rule does not influence the expected return:)

$$= \max_{\beta^1_\tau} \min_{\mu^2 \in M^2_{|\sigma_\tau\rangle}} \max_{\mu^1 \in M^1_{|\sigma_\tau, \beta^1_\tau\rangle}} \left[r(\sigma_\tau, \beta^1_\tau, \beta^2_\tau(\mu_2)) + \gamma V_{\tau+1}(T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau(\mu_2)), \mu^1, \mu^2) \right]$$

(using again the minimax theorem's equivalence between maximin and minimax on an appropriate game:)

$$= \max_{\beta^1_\tau} \max_{\mu^1 \in M^1_{|\sigma_\tau, \beta^1_\tau\rangle}} \min_{\mu^2 \in M^2_{|\sigma_\tau\rangle}} \left[r(\sigma_\tau, \beta^1_\tau, \beta^2_\tau(\mu_2)) + \gamma V_{\tau+1}(T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau(\mu_2)), \mu^1, \mu^2) \right] \tag{3.8}$$

(merging both maxes (and with explanations thereafter):)

$$= \max_{\mu^1 \in M^1_{|\sigma_\tau\rangle}} \min_{\mu^2 \in M^2_{|\sigma_\tau\rangle}} \left[r(\sigma_\tau, \beta^1_\tau(\mu_1), \beta^2_\tau(\mu_2)) + \gamma V_{\tau+1}(T(\sigma_\tau, \beta^1_\tau(\mu^1), \beta^2_\tau(\mu_2)), \mu^1, \mu^2) \right] \tag{3.9}$$

(again with the equivalence property discussed before the lemma:)

$$= \max_{\mu^1 \in M^1_{|\sigma_\tau\rangle}} \min_{\mu^2 \in M^2_{|\sigma_\tau\rangle}} V_\tau(\sigma_\tau, \mu^1, \mu^2)$$
$$= \max_{\beta^1_{\tau:H-1|\sigma_\tau\rangle}} \min_{\beta^2_{\tau:H-1|\sigma_\tau\rangle}} V_\tau(\sigma_\tau, \beta^1_{\tau:H-1}, \beta^2_{\tau:H-1})$$
$$\stackrel{\text{def}}{=} V^*_\tau(\sigma_\tau).$$

More precisely, line 3.7 (and, similarly, line 3.9) is obtained by observing that

- minimizing over both (i) β^2_τ and (ii) μ^2 constrained by σ_τ and $\boldsymbol{\beta}_\tau$ is equivalent to minimizing over μ^2 constrained by σ_τ and β^1_τ; and
- in the reminder of the formula, decision rule β^2_τ at time τ can be retrieved as a function of μ^2 (noted $\beta^2_\tau(\mu^2)$).

Also, line 3.8 results from the observation that, while $M^1_{|\sigma_\tau,\beta^1_\tau\rangle}$ and $M^2_{|\sigma_\tau\rangle}$ allow to actually make decisions over different time intervals, we are here minimizing over μ^2 while maximizing over μ^1 a function that is linear in both input spaces. This amounts to solving some 2-player zero-sum normal-form game, hence the applicability of von Neumann's minimax theorem.

The above derivation tells us that the maximin value (the best outcome player 1 can guarantee whatever player 2's strategy) in the one-time-step game is thus the Nash equilibrium value (NEV) for the complete subgame from τ onwards. □

Concavity and Convexity Results Theorems 3.1.5 and 3.1.6 together imply that the following Theorem 3.1.7, originally stated in a game with public strategies by Wiggers et al. (2016), also holds in our setting where strategies remain private.

As a preliminary step, an occupancy state σ_τ can be decomposed into a *marginal term* $\sigma^{m,1}_\tau$ and a *conditional term* $\sigma^{c,1}_\tau$ (Wiggers et al. 2016a), where

- $\sigma^{m,1}_\tau(\theta^1_\tau) = \sum_{\theta^2_\tau} \sigma_\tau(\theta^1_\tau, \theta^2_\tau)$ is the probability of 1 facing θ^1_τ under σ_τ, and
- $\sigma^{c,1}_\tau(\theta^2_\tau|\theta^1_\tau) = \frac{\sigma_\tau(\theta^1_\tau,\theta^2_\tau)}{\sigma^{m,1}_\tau(\theta^1_\tau)}$ is the probability of 2 facing θ^2_τ under σ_τ given that 1 faces θ^1_τ,

so that $\sigma_\tau(\theta^1_\tau, \theta^2_\tau) = \sigma^{m,1}_\tau(\theta^1_\tau) \cdot \sigma^{c,1}_\tau(\theta^2_\tau|\theta^1_\tau)$. (Symmetric definitions apply by swapping players 1 and 2.) In addition, let us denote $T^1_m(\sigma_\tau, \boldsymbol{\beta}_\tau)$ and $T^1_c(\sigma_\tau, \boldsymbol{\beta}_\tau)$ the marginal and conditional terms associated to $T(\sigma_\tau, \boldsymbol{\beta}_\tau)$.

Now, if 1 faces AOH θ^1_τ, knows 2's future strategy $\beta^2_{\tau:H-1}$, and has access to $\sigma^{c,1}_\tau(\theta^2_\tau|\theta^1_\tau)$ for any θ^2_τ, then she faces a POMDP whose optimal value we denote $\nu^2_{[\sigma^{c,1}_\tau, \beta^2_{\tau:H-1}]}(\theta^1_\tau)$. This leads to defining the best-response value vector $\nu^2_{[\sigma^{c,1}_\tau, \beta^2_{\tau:H-1}]}$, which contains one component per AOH θ^1_τ, and writing the value of 1's best response against $\beta^2_{\tau:H-1}$ under σ_τ as $\sigma^{m,1}_\tau \cdot \nu^2_{[\sigma^{c,1}_\tau, \beta^2_{\tau:H-1}]}$. But then, because 2 also knows σ_τ, she can in fact pick $\beta^2_{\tau:H-1}$ to minimize this value, so that we get the following theorem.

Theorem 3.1.7 (Concavity and convexity (CC) of V^*_τ (Wiggers et al. 2016a, Thm. 2))**.** *For any $\tau \in \{0\,..\,H-1\}$, V^*_τ is (i) concave w.r.t. $\sigma^{m,1}_\tau$ for a fixed $\sigma^{c,1}_\tau$, and (ii) convex w.r.t. $\sigma^{m,2}_\tau$ for a fixed $\sigma^{c,2}_\tau$. More precisely,*

$$V^*_\tau(\sigma_\tau) = \min_{\beta^2_{\tau:H-1}} \left[\sigma^{m,1}_\tau \cdot \nu^2_{[\sigma^{c,1}_\tau, \beta^2_{\tau:H-1}]}\right] = \max_{\beta^1_{\tau:H-1}} \left[\sigma^{m,2}_\tau \cdot \nu^1_{[\sigma^{c,2}_\tau, \beta^1_{\tau:H-1}]}\right], \textit{where}$$

$$\nu^2_{[\sigma^{c,1}_\tau, \beta^2_{\tau:H-1}]}(\theta^1_\tau) \stackrel{\text{def}}{=} \max_{\beta^1_{\tau:H-1}} \mathbb{E}_{\theta^2_\tau \sim \sigma^{c,1}_\tau(\theta^1_\tau)} \left\{ \sum_{t=\tau}^{H-1} \gamma^{t-\tau} r(S_t, A^1_t, A^2_t) \mid \beta^1_{\tau:H-1}, \beta^2_{\tau:H-1} \right\}, \quad \textit{and} \tag{3.10}$$

$$\nu^1_{[\sigma^{c,2}_\tau, \beta^1_{\tau:H-1}]}(\theta^2_\tau) \stackrel{\text{def}}{=} \min_{\beta^2_{\tau:H-1}} \mathbb{E}_{\theta^1_\tau \sim \sigma^{c,2}_\tau(\theta^2_\tau)} \left\{ \sum_{t=\tau}^{H-1} \gamma^{t-\tau} r(S_t, A^1_t, A^2_t) \mid \beta^1_{\tau:H-1}, \beta^2_{\tau:H-1} \right\}. \tag{3.11}$$

Proof. We start from Theorem 3.1.5:

$$V^*_\tau(\sigma_\tau) = \min_{\beta^2_{\tau:H-1}} \max_{\beta^1_{\tau:H-1}} \left[V_\tau(\sigma_\tau, \beta^1_{\tau:H-1}, \beta^1_{\tau:H-1})\right]$$
$$= \min_{\beta^2_{\tau:H-1}} \max_{\beta^1_{\tau:H-1}} \left[\mathbb{E}\left\{\sum_{t=\tau}^{H-1} \gamma^{t-\tau} r(S_t, A^1_t, A^2_t) \mid \theta^1_\tau, \beta^1_{\tau:H-1}, \beta^2_{\tau:H-1}, \sigma^{c,1}_\tau\right\}\right],$$

then, observing that 1's best response to $\beta^2_{\tau:}$ can be computed for each AOH θ^1_τ independently, we can swap the max operator and part of the expectation one ($\mathbb{E}$) as follows:[27]

$$= \min_{\beta^2_{\tau:H-1}} \mathbb{E}_{\theta^1_\tau \sim \sigma^{m,1}_\tau} \underbrace{\left\{\max_{\beta^1_{\tau:H-1}} \mathbb{E}_{\theta^2_\tau \sim \sigma^{c,1}_\tau(\theta^1)} \left[\sum_{t=\tau}^{H-1} \gamma^{t-\tau} r(S_t, A^1_t, A^2_t) \mid \beta^1_{\tau:H-1}, \beta^2_{\tau:H-1}\right]\right\}}_{\text{best-response of 1 to } \beta^2_{\tau:} \text{ under } \theta^1_\tau}$$

and, recognizing the components of vector $\nu^2_{[\sigma^{c,1}_\tau, \beta^2_{\tau:H-1}]}$ and writing the expectation over AOHs $\theta^1_{\tau:}$ as a scalar product:

$$= \min_{\beta^2_{\tau:H-1}} \left[\sigma^{m,1}_\tau \cdot \nu^2_{[\sigma^{c,1}_\tau, \beta^2_{\tau:H-1}]}\right].$$

□

In practice, however, such convexity/concavity properties alone only allow upper-bounding V^*_τ for finitely many conditional terms $\sigma^{c,i}_\tau$, thus *not* for the whole occupancy space, as required to enable DP and HS in our game.

To address this limitation, we propose to seek a continuity property in the conditional dimension (w.r.t. $\sigma^{c,1}_\tau$), which requires to first study continuity properties of transition functions.

Continuity Properties of the Transition Functions

Lemma 3.1.8 (Linearity of T^1_m (Wiggers et al. 2016a, Lemma 4.2.3))**.** *$T^1_m(\sigma_\tau, \boldsymbol{\beta}_\tau)$ is linear in σ_τ, β^1_τ, and β^2_τ.*

Proof.

$$T^1_m(\sigma_\tau, \boldsymbol{\beta}_\tau)(\theta^1_\tau, a^1, z^1) \tag{3.12}$$
$$= \sum_{\theta^2_\tau, a^2, z^2} T(\sigma_\tau, \boldsymbol{\beta}_\tau)((\theta^1_\tau, a^1, z^1), (\theta^2_\tau, a^2, z^2))$$
$$= \sum_{s', \theta^2_\tau, a^2, z^2} \beta^1_\tau(\theta^1_\tau, a^1) \beta^2_\tau(\theta^2_\tau, a^2) \sum_s P^{z^1, z^2}_{a^1, a^2}(s'|s) b(s|\theta^1_\tau, \theta^2_\tau) \sigma_\tau(\theta^1_\tau, \theta^2_\tau)$$
$$= \beta^1_\tau(\theta^1_\tau, a^1) \sum_{\theta^2_\tau, a^2} \beta^2_\tau(\theta^2_\tau, a^2) \sum_{s, s', z^2} P^{z^1, z^2}_{a^1, a^2}(s'|s) b(s|\theta^1_\tau, \theta^2_\tau) \sigma_\tau(\theta^1_\tau, \theta^2_\tau). \tag{3.13}$$

□

Lemma 3.1.9 (Independance properties of T^1_c (Wiggers et al. 2016a))**.** *$T^1_c(\sigma_\tau, \boldsymbol{\beta}_\tau)$ is independent of β^1_τ and $\sigma^{m,1}_\tau$.*

Proof.

$$T^1_c(\sigma_\tau, \boldsymbol{\beta}_\tau)((\theta^2_\tau, a^2, z^2)|(\theta^1_\tau, a^1, z^1)) = \frac{T(\sigma_\tau, \boldsymbol{\beta}_\tau)((\theta^1_\tau, a^1, z^1), (\theta^2_\tau, a^2, z^2))}{\sum_{\tilde{\theta}^2_\tau, a^2, z^2} T(\sigma_\tau, \boldsymbol{\beta}_\tau)((\theta^1_\tau, a^1, z^1), (\tilde{\theta}^2_\tau, a^2, z^2))}$$

[27] Note that this property is well known in Bayesian games, where AOHs correspond to *types*, *cf.* Harsanyi 1968, Th. 1, p. 321.

$$
\begin{aligned}
&= \frac{\beta^1_\tau(\theta^1_\tau, a^1)\beta^2_\tau(\theta^2_\tau, a^2)\sum_{s,s'} P^{z^1,z^2}_{a^1,a^2}(s'|s)b(s|\theta^1_\tau,\theta^2_\tau)\sigma_\tau(\theta^1_\tau,\theta^2_\tau)}{\beta^1_\tau(\theta^1_\tau, a^1)\sum_{\tilde\theta^2,a^2}\beta^2_\tau(\tilde\theta^2_\tau, a^2)\sum_{s,s',z^2} P^{z^1,z^2}_{a^1,a^2}(s'|s)b(s|\theta^1_\tau,\tilde\theta^2_\tau)\sigma_\tau(\theta^1_\tau,\tilde\theta^2_\tau)} \\
&= \frac{\beta^2_\tau(\theta^2_\tau, a^2)\sum_{s,s'} P^{z^1,z^2}_{a^1,a^2}(s'|s)b(s|\theta^1_\tau,\theta^2_\tau)\sigma_\tau(\theta^1_\tau,\theta^2_\tau)}{\sum_{\tilde\theta^2,a^2}\beta^2_\tau(\tilde\theta^2_\tau, a^2)\sum_{s,s',z^2} P^{z^1,z^2}_{a^1,a^2}(s'|s)b(s|\theta^1_\tau,\tilde\theta^2_\tau)\sigma_\tau(\theta^1_\tau,\tilde\theta^2_\tau)} \\
&= \frac{\beta^2_\tau(\theta^2_\tau, a^2)\sum_{s,s'} P^{z^1,z^2}_{a^1,a^2}(s'|s)b(s|\theta^1_\tau,\theta^2_\tau)\overbrace{\sigma^{c,1}_\tau(\theta^2_\tau|\theta^1_\tau)\sigma^{m,1}_\tau(\theta^1_\tau)}}{\sum_{\tilde\theta^2,a^2}\beta^2_\tau(\tilde\theta^2_\tau, a^2)\sum_{s,s',z^2} P^{z^1,z^2}_{a^1,a^2}(s'|s)b(s|\theta^1_\tau,\tilde\theta^2_\tau)\underbrace{\sigma^{c,1}_\tau(\tilde\theta^2_\tau|\theta^1_\tau)\sigma^{m,1}_\tau(\theta^1_\tau)}} \\
&= \frac{\left(\beta^2_\tau(\theta^2_\tau, a^2)\sum_{s,s'} P^{z^1,z^2}_{a^1,a^2}(s'|s)b(s|\theta^1_\tau,\theta^2_\tau)\sigma^{c,1}_\tau(\theta^2_\tau|\theta^1_\tau)\right)\sigma^{m,1}_\tau(\theta^1_\tau)}{\left(\sum_{\tilde\theta^2,a^2}\beta^2_\tau(\tilde\theta^2_\tau, a^2)\sum_{s,s',z^2} P^{z^1,z^2}_{a^1,a^2}(s'|s)b(s|\theta^1_\tau,\tilde\theta^2_\tau)\sigma^{c,1}_\tau(\tilde\theta^2_\tau|\theta^1_\tau)\right)\sigma^{m,1}_\tau(\theta^1_\tau)} \\
&= \frac{\beta^2_\tau(\theta^2_\tau, a^2)\sum_{s,s'} P^{z^1,z^2}_{a^1,a^2}(s'|s)b(s|\theta^1_\tau,\theta^2_\tau)\sigma^{c,1}_\tau(\theta^2_\tau|\theta^1_\tau)}{\sum_{\tilde\theta^2,a^2}\beta^2_\tau(\tilde\theta^2_\tau, a^2)\sum_{s,s',z^2} P^{z^1,z^2}_{a^1,a^2}(s'|s)b(s|\theta^1_\tau,\tilde\theta^2_\tau)\sigma^{c,1}_\tau(\tilde\theta^2_\tau|\theta^1_\tau)}.
\end{aligned}
$$

□

Lemma 3.1.10 (Lipschitz continuity of T)**.** *At depth τ, $T(\sigma_\tau, \boldsymbol{\beta}_\tau)$ is linear in β^1_τ, β^2_τ, and σ_τ, where $\boldsymbol{\beta}_\tau = \langle \beta^1_\tau, \beta^2_\tau \rangle$. It is more precisely* 1*-Lipschitz-continuous (*1*-LC) in σ_τ (in* 1*-norm),* i.e., *for any σ_τ, σ'_τ:*

$$\|T(\sigma'_\tau, \boldsymbol{\beta}_\tau) - T(\sigma_\tau, \boldsymbol{\beta}_\tau)\|_1 \leqslant 1 \cdot \|\sigma'_\tau - \sigma_\tau\|_1.$$

Proof. Let σ be an occupancy state at time τ and $\boldsymbol{\beta}_\tau$ be a decision rule. Then, as seen in the proof of Proposition 2.2.17, the next occupancy state $\sigma' = T(\sigma, \boldsymbol{\beta}_\tau)$ satisfies, for any s' and $(\boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{z})$:

$$
\begin{aligned}
\sigma'(\boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{z}) &\overset{\text{def}}{=} Pr(\boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{z} | \sigma, \beta^1_\tau, \beta^2_\tau) \\
&= \beta^1_\tau(\theta^1, a^1)\beta^2_\tau(\theta^2, a^2)\left[\sum_{s',s\in\mathcal{S}} P^{\boldsymbol{z}}_{\boldsymbol{a}}(s'|s)b(s|\boldsymbol{\theta})\right]\sigma(\boldsymbol{\theta}).
\end{aligned}
$$

The probability $b(s|\boldsymbol{\theta})$ depending only on the model (transition function and initial belief), the next occupancy state σ' thus evolves linearly w.r.t. (i) *private* decision rules β^1_τ and β^2_τ, and (ii) the occupancy state σ.

1-Lipschitz-continuity holds because each component of vector σ_τ is distributed over multiple components of σ'. Indeed, let us view two occupancy states as vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, and their corresponding next states under $\boldsymbol{\beta}_\tau$ as $M\boldsymbol{x}$ and $M\boldsymbol{y}$, where $M \in \mathbb{R}^{m\times n}$ is the corresponding transition matrix (*i.e.,* which turns σ into $\sigma' \overset{\text{def}}{=} T(\sigma_\tau, \boldsymbol{\beta}_\tau)$). Then,

$$
\begin{aligned}
\|M\boldsymbol{x} - M\boldsymbol{y}\|_1 &\overset{\text{def}}{=} \sum_{j=1}^{m} |\sum_{i=1}^{n} M_{i,j}(x_i - y_i)| \\
&\leqslant \sum_{j=1}^{m}\sum_{i=1}^{n} |M_{i,j}(x_i - y_i)| && \text{(convexity of } |\cdot|) \\
&= \sum_{j=1}^{m}\sum_{i=1}^{n} M_{i,j}|x_i - y_i| && (\forall i,j,\ M_{i,j} \geqslant 0) \\
&= \sum_{i=1}^{n}\underbrace{\sum_{j=1}^{m} M_{i,j}}_{=1}|x_i - y_i| && (M \text{ is a transition matrix}) \\
&\overset{\text{def}}{=} \|\boldsymbol{x} - \boldsymbol{y}\|_1.
\end{aligned}
$$

□

3.1.2 Towards Solving zs-oMGs

This section aims at providing the second tool for DP and HS with continuous state spaces, *i.e.,* bounding approximators of optimal value functions which will allow generalization across the occupancy space. Their update and selection operators are written as linear programs, and they turn out to come with solution strategies.

3.1.2.1 Bounding value functions

So far, several issues prevented from applying the HSVI scheme to zs-POSGs, starting with the continuous spaces of

1. occupancy states (zs-oMG states) and
2. decision rules (zs-oMG actions).

One can address (1) by introducing upper- and lower- bounding functions $\overline{V}_\tau(\sigma_\tau)$ and $\underline{V}_\tau(\sigma_\tau)$ of $V^*_\tau(\sigma_\tau)$. The following starts with a lemma that allows proving a key Lipschitz-continuity property of V^*.

Lemma 3.1.11. *At depth τ, $V_\tau(\sigma_\tau, \boldsymbol{\beta}_{\tau:})$ is linear w.r.t. σ_τ.*

Note: This result in fact applies to any reward function of a general-sum POSG with any number of agents (here N), e.g., *to a cp-POSG.*

Proof. This property trivially holds for $\tau = H-1$ because

$$\begin{aligned} V_{H-1}(\sigma_{H-1}, \boldsymbol{\beta}_{H-1:}) &= r(\sigma_{H-1}, \boldsymbol{\beta}_{H-1}) \\ &= \sum_{s,\boldsymbol{a}} \left(\sum_{\boldsymbol{\theta}} Pr(s, \boldsymbol{a}|\boldsymbol{\theta}) \sigma_{H-1}(\boldsymbol{\theta}) \right) r(s, \boldsymbol{a}) \\ &= \sum_{s,\boldsymbol{a}} \left(\sum_{\boldsymbol{\theta}} b(s|\boldsymbol{\theta}) \boldsymbol{\beta}_\tau(\boldsymbol{a}|\boldsymbol{\theta}) \sigma_{H-1}(\boldsymbol{\theta}) \right) r(s, \boldsymbol{a}) \\ &= \sum_{s,\boldsymbol{\theta}} b(s|\boldsymbol{\theta}) \sigma_{H-1}(\boldsymbol{\theta}) \left(\sum_{\boldsymbol{a}} \boldsymbol{\beta}_\tau(\boldsymbol{a}|\boldsymbol{\theta}) r(s, \boldsymbol{a}) \right). \end{aligned}$$

Now, let us assume that the property holds for $\tau + 1 \in \{1 \,..\, H-1\}$. Then,

$$\begin{aligned} V_\tau(\sigma_\tau, \boldsymbol{\beta}_{\tau:}) &= \sum_{s,\boldsymbol{a}} \Big(\sum_{\boldsymbol{\theta}} b(s|\boldsymbol{\theta}) \boldsymbol{\beta}_\tau(\boldsymbol{a}|\boldsymbol{\theta}) \sigma_\tau(\boldsymbol{\theta}) \Big) r(s, \boldsymbol{a}) + \gamma V_{\tau+1} \left(T(\sigma_\tau, \boldsymbol{\beta}_\tau), \boldsymbol{\beta}_{\tau+1:} \right) \\ &= \sum_{s,\boldsymbol{\theta}} b(s|\boldsymbol{\theta}) \sigma_\tau(\boldsymbol{\theta}) \Big(\sum_{\boldsymbol{a}} \boldsymbol{\beta}_\tau(\boldsymbol{a}|\boldsymbol{\theta}) r(s, \boldsymbol{a}) \Big) + \gamma V_{\tau+1} \left(T(\sigma_\tau, \boldsymbol{\beta}_\tau), \boldsymbol{\beta}_{\tau+1:} \right). \end{aligned}$$

As

- $T(\sigma_\tau, \boldsymbol{\beta}_\tau)$ is linear in σ_τ (Lemma 3.1.10) and
- $V_{\tau+1}(\sigma_{\tau+1}, \boldsymbol{\beta}_{\tau+1:})$ is linear in $\sigma_{\tau+1}$ (induction hypothesis),

their composition, $V_{\tau+1}(T(\sigma_\tau, \boldsymbol{\beta}_\tau), \boldsymbol{\beta}_{\tau+1:})$, is also linear in σ_τ, and so is $V_\tau(\sigma_\tau, \boldsymbol{\beta}_{\tau:})$. □

Theorem 3.1.12 (Lipschitz-Continuity of V^*)**.** *Let $h_\tau \overset{\text{def}}{=} \frac{1-\gamma^{H-\tau}}{1-\gamma}$ (or $h_\tau \overset{\text{def}}{=} H - \tau$ if $\gamma = 1$). Then $V^*_\tau(\sigma_\tau)$ is λ_τ-Lipschitz continuous in σ_τ at any depth $\tau \in \{0 \,..\, H-1\}$, where $\lambda_\tau = \frac{1}{2} h_\tau \left(r_{\max} - r_{\min} \right)$.*

Proof. At depth τ, the value of any behavioral strategy $\boldsymbol{\beta}_{\tau:}$ is bounded, independently of σ_τ, by

$$V_\tau^{\max} \overset{\text{def}}{=} h_\tau r_{\max}, \quad \text{where } r_{\max} \overset{\text{def}}{=} \max_{s,\boldsymbol{a}} r(s,\boldsymbol{a}), \text{ and}$$
$$V_\tau^{\min} \overset{\text{def}}{=} h_\tau r_{\min}, \quad \text{where } r_{\min} \overset{\text{def}}{=} \min_{s,\boldsymbol{a}} r(s,\boldsymbol{a}).$$

Thus, $V_{\boldsymbol{\beta}_{\tau:}}$ being a linear function defined over a probability simplex ($\mathcal{O}_\tau^\sigma$) (*cf.* Section 3.1.2.1) and bounded by $[V_\tau^{\min}, V_\tau^{\max}]$, we can apply Horák's PhD thesis' Lemma 3.5 (p. 33) (Horák 2019) to establish that it is also λ_τ-LC, *i.e.*,

$$|V_{\boldsymbol{\beta}_{\tau:}}(\sigma) - V_{\boldsymbol{\beta}_{\tau:}}(\sigma')| \leqslant \lambda_\tau \|\sigma - \sigma'\|_1 \quad (\forall \sigma, \sigma'),$$
$$\text{with } \lambda_\tau = \frac{V_\tau^{\max} - V_\tau^{\min}}{2}.$$

Considering now optimal solutions, this means that, at depth τ and for any $(\sigma, \sigma') \in \mathcal{O}_\tau^\sigma$:

$$\begin{aligned} V_\tau^*(\sigma) - V_\tau^*(\sigma') &= \max_{\beta_{\tau:}^1} \min_{\beta_{\tau:}^2} V_\tau(\sigma, \beta_{\tau:}^1, \beta_{\tau:}^2) - \max_{\beta_{\tau:}'^1} \min_{\beta_{\tau:}'^2} V_\tau(\sigma', \beta_{\tau:}'^1, \beta_{\tau:}'^2) \\ &\leqslant \max_{\beta_{\tau:}^1} \min_{\beta_{\tau:}^2} \left[V_\tau(\sigma', \beta_{\tau:}^1, \beta_{\tau:}^2) + \lambda_\tau \|\sigma - \sigma'\|_1 \right] - \max_{\beta_{\tau:}'^1} \min_{\beta_{\tau:}'^2} V_\tau(\sigma', \beta_{\tau:}'^1, \beta_{\tau:}'^2) \\ &= \lambda_\tau \|\sigma - \sigma'\|_1. \end{aligned}$$

Symmetrically,

$$V_\tau^*(\sigma) - V_\tau^*(\sigma') \geqslant -\lambda_\tau \|\sigma - \sigma'\|_1,$$

hence the expected result:

$$|V_\tau^*(\sigma) - V_\tau^*(\sigma')| \leqslant \lambda_\tau \|\sigma - \sigma'\|_1.$$

□

Combining the Lipschitz continuity of V^* and Theorem 3.1.7, we can derive bounding functions $\overline{V}_\tau$ and $\underline{V}_\tau$, for each time step. These approximators allow generalizing knowledge from a finite number of subgames to any subgame.

Theorem 3.1.13 (Upper-bouding V_τ^*)**.** *Let $\mathcal{J}$ be a set of tuples, each defined by (i) a conditional occupancy state $\sigma_\tau^{c,1}$ and (ii) a vector $\overline{\mu}_\tau^2$.*

$$\forall \sigma_\tau,\ V^*(\sigma_\tau) \leqslant \overline{V}_\tau(\sigma_\tau) \overset{\text{def}}{=} \min_{\langle \tilde{\sigma}_\tau^{c,1}, \langle \overline{\nu}_\tau^2, \beta_{\tau:}^2 \rangle \rangle \in \overline{\mathcal{J}}_\tau} \left[\sigma_\tau^{m,1} \cdot \overline{\nu}_\tau^2 + \lambda_\tau \|\sigma_\tau - \sigma_\tau^{m,1} \tilde{\sigma}_\tau^{c,1}\|_1 \right],$$

where $\overline{\nu}_\tau^2$ component-wise upper-bounds $\nu^2_{[\tilde{\sigma}_\tau^{c,1}, \beta_{\tau:}^2]}$ for some $\beta_{\tau:}^2$.

Proof. To find a form that could be appropriate for an upper-bound approximation of V_τ^*, let us consider an OS σ_τ and a single tuple $\langle \tilde{\sigma}_\tau, \nu^2_{[\tilde{\sigma}_\tau^{c,1}, \beta_{\tau:}^2]} \rangle$, and define $\zeta_\tau \overset{\text{def}}{=} \sigma_\tau^{m,1} \tilde{\sigma}_\tau^{c,1}$. Then,

$$\begin{aligned} V^*(\sigma_\tau) &\leqslant V^*(\zeta_\tau) + \lambda_\tau \|\sigma_\tau - \zeta_\tau\|_1 && \text{(LC, } \textit{cf.} \text{ Theorem 3.1.12)} \\ &= V^*(\sigma_\tau^{m,1} \tilde{\sigma}_\tau^{c,1}) + \lambda_\tau \|\sigma_\tau - \zeta_\tau\|_1 \\ &\leqslant \sigma_\tau^{m,1} \cdot \nu^2_{[\tilde{\sigma}_\tau^{c,1}, \beta_{\tau:}^2]} + \lambda_\tau \|\sigma_\tau - \sigma_\tau^{m,1} \tilde{\sigma}_\tau^{c,1}\|_1. && \text{(Cvx, } \textit{cf.} \text{ Theorem 3.1.7)} \end{aligned}$$

Notes:

- $\tilde{\sigma}_\tau^{m,1}$ does not appear in the resulting upper bound, thus will not need to be specified.
- For $\tau = H-1$, $\nu^2_{[\tilde{\sigma}_\tau^{c,1}, \beta_{\tau:}^2]}$ is a simple function of r, $\tilde{\sigma}_\tau^{c,1}$, $\beta_{\tau:}^2$, and the dynamics of the system, as described by Wiggers et al. (2016), Eq. (9).

From this, we can deduce the following appropriate forms of upper and (symmetrically) lower bound function approximations for V^*_τ:

$$\overline{V}_\tau(\sigma_\tau) = \min_{\langle\tilde{\sigma}^{c,1}_\tau,\langle\overline{\nu}^2_\tau,\beta^2_{\tau:}\rangle\rangle\in\overline{\mathcal{J}}_\tau} \left[\sigma^{m,1}_\tau \cdot \overline{\nu}^2_\tau + \lambda_\tau\|\sigma_\tau - \sigma^{m,1}_\tau\tilde{\sigma}^{c,1}_\tau\|_1\right], \text{ and} \tag{3.14}$$

$$\underline{V}_\tau(\sigma_\tau) = \max_{\langle\tilde{\sigma}^{c,2}_\tau,\langle\underline{\nu}^1_\tau,\beta^1_{\tau:}\rangle\rangle\in\underline{\mathcal{J}}_\tau} \left[\sigma^{m,2}_\tau \cdot \underline{\nu}^1_\tau - \lambda_\tau\|\sigma_\tau - \sigma^{m,2}_\tau\tilde{\sigma}^{c,2}_\tau\|_1\right], \tag{3.15}$$

which are respectively concave in $\sigma^{m,1}_\tau$ and convex in $\sigma^{m,2}_\tau$, and which both exploit the Lipschitz continuity. □

Yet, this yields (generally non-convex) Lipschitz-continuous functions whose max-min optimization would be intractable, so that the issue (2) of continuous decision rules remains unsolved. Also, we do not know how to retrieve valid solution strategies. In particular, and as illustrated in Example 2.1.26 (example 2.1.26), simply concatenating decision rules backwards from $\tau = H-1$ to 0 would not guarantee globally-consistent solutions, and could result in exploitable strategies.

But then, combining Theorems 3.1.6 and 3.1.7 leads to introducing a novel value function (denoted $W^{1,*}_\tau$) through writing, for any OS σ_τ:

$$V^*_\tau(\sigma_\tau) = \max_{\beta^1_\tau} \underbrace{\min_{\beta^2_{\tau:H-1}\in\mathcal{B}^2_\tau} \left[r(\sigma_\tau,\boldsymbol{\beta}_\tau) + \gamma\sigma^{m,1}_{\tau+1}\cdot\nu^2_{[\sigma^{c,1}_{\tau+1},\beta^2_{\tau+1:H-1}]}\right]}_{\stackrel{\text{def}}{=}W^{1,*}_\tau(\sigma_\tau,\beta^1_\tau)}.$$

Assuming that player 2 can only respond with one of finitely many stored strategies, the concavity and λ_τ-Lipschitz-continuity of $W^{1,*}_\tau$ allow upper-bounding it with finitely many tuples $w = \langle\tilde{\sigma}_\tau, \beta^2_\tau, \langle\overline{\nu}^2_{\tau+1}, \beta^2_{\tau+1:}\rangle\rangle$ stored in sets $\overline{\mathcal{I}}_\tau$, and where $\overline{\nu}^2_{\tau+1}$ upper-bounds $\nu^2_{[\tilde{\sigma}^{c,1}_{\tau+1},\beta^2_{\tau+1:}]}$. To prove this result, we start with two lemmas, that respectively aim at:

1. describing $W^{1,*}_\tau$ as a lower envelope of linear functions w.r.t. β^1_τ; and
2. showing the Lipschitz-continuity of vectors ν w.r.t. conditional terms of occupancy states.

Lemma 3.1.14. *Considering that vectors $\nu^2_{[\sigma^{c,1}_H,\beta^2_{H:}]}$ are null vectors, we have, for all $\tau \in \{0\,..\,H-1\}$:*

$$W^{1,*}_\tau(\sigma_\tau,\beta^1_\tau) = \min_{\beta^2_\tau,\langle\beta^2_{\tau+1:},\nu^2_{[T^1_c(\sigma_\tau,\beta^2_\tau),\beta^2_{\tau+1:}]}\rangle} \beta^1_\tau\cdot\left[r(\sigma_\tau,\cdot,\beta^2_\tau) + \gamma T^1_m(\sigma_\tau,\cdot,\beta^2_\tau)\cdot\nu^2_{[T^1_c(\sigma_\tau,\beta^2_\tau),\beta^2_{\tau+1:}]}\right].$$

Proof. Considering that vectors $\nu^2_{[\sigma^{c,1}_H,\beta^2_{H:}]}$ are null vectors, we have, for all $\tau \in \{0\,..\,H-1\}$:

$$W^{1,*}_\tau(\sigma_\tau,\beta^1_\tau) = \min_{\beta^2_\tau} Q^*_\tau(\sigma_\tau,\beta^1_\tau,\beta^2_\tau) = \min_{\beta^2_\tau}\left[r(\sigma_\tau,\boldsymbol{\beta}_\tau) + \gamma V^*_{\tau+1}(T(\sigma_\tau,\boldsymbol{\beta}_\tau))\right]$$

(Line below exploits Theorem 3.1.7 (p. 47) and T^1_c's independence from β^1_τ (Lemma 3.1.9).)

$$= \min_{\beta^2_\tau}\left[r(\sigma_\tau,\boldsymbol{\beta}_\tau) + \gamma \min_{\langle\beta^2_{\tau+1:},\nu^2_{[T^1_c(\sigma_\tau,\beta^2_\tau),\beta^2_{\tau+1:}]}\rangle}\left[T^1_m(\sigma_\tau,\boldsymbol{\beta}_\tau)\cdot\nu^2_{[T^1_c(\sigma_\tau,\beta^2_\tau),\beta^2_{\tau+1:}]}\right]\right]$$

$$= \min_{\beta^2_\tau,\langle\beta^2_{\tau+1:},\nu^2_{[T^1_c(\sigma_\tau,\beta^2_\tau),\beta^2_{\tau+1:}]}\rangle}\left[r(\sigma_\tau,\boldsymbol{\beta}_\tau) + \gamma T^1_m(\sigma_\tau,\boldsymbol{\beta}_\tau)\cdot\nu^2_{[T^1_c(\sigma_\tau,\beta^2_\tau),\beta^2_{\tau+1:}]}\right]$$

(Line below exploits r and T^1_m's linearity in β^1_τ (Lemma 3.1.8).)

$$= \min_{\beta^2_\tau,\langle\beta^2_{\tau+1:},\nu^2_{[T^1_c(\sigma_\tau,\beta^2_\tau),\beta^2_{\tau+1:}]}\rangle} {\beta^1_\tau}^\top\cdot\left[r(\sigma_\tau,\cdot,\beta^2_\tau) + \gamma T^1_m(\sigma_\tau,\cdot,\beta^2_\tau)\cdot\nu^2_{[T^1_c(\sigma_\tau,\beta^2_\tau),\beta^2_{\tau+1:}]}\right].$$

□

Lemma 3.1.15. *Let us consider $\tau \in \{0\,..\,H-1\}$, θ^1_τ, and $\beta^2_{\tau:}$. Then $\nu^2_{[\sigma^{c,1}_\tau, \beta^2_{\tau:}]}(\theta^1_\tau)$ is λ_τ-LC in $\sigma^{c,1}_\tau(\cdot|\theta^1_\tau)$.*

Equivalently, we will also write that $\nu^2_{[\sigma^{c,1}_\tau, \beta^2_{\tau:}]}$ is λ_τ-LC in $\sigma^{c,1}_\tau$ in vector-wise *1-norm,* i.e.,

$$\overrightarrow{|\nu^2_{[\sigma^{c,1}_\tau, \beta^2_{\tau:}]} - \nu^2_{[\tilde{\sigma}^{c,1}_\tau, \beta^2_{\tau:}]}|_1} \leqslant \lambda_\tau \overrightarrow{\|\sigma^{c,1}_\tau - \tilde{\sigma}^{c,1}_\tau\|_1},$$

where (i) the absolute value of a vector is obtained by taking the absolute value of each component; and (ii) the vector-wise 1-norm of a matrix is a vector made of the 1-norm of each of its component vectors.

Proof. For any θ^1_τ, $\sigma^{c,1}_\tau$ and $\beta^2_{\tau:}$ induce a POMDP for player 1 from τ on, where (i) the state at any $t \in \{\tau\,..\,H-1\}$ corresponds to a pair $\langle s, \theta^2_t\rangle$, and (ii) the initial belief is derived from $\sigma^{c,1}_\tau(\cdot|\theta^1_t)$. The belief state at t thus gives:

$$b_{\theta^1_t}(s, \theta^2_t) \stackrel{\text{def}}{=} Pr(s, \theta^2_t|\theta^1_t) = \underbrace{Pr(s|\theta^2_t, \theta^1_t)}_{b^{\text{HMM}}_{\theta^2_t, \theta^1_t}(s)} \cdot \underbrace{Pr(\theta^2_t|\theta^1_t)}_{\sigma^{c,1}_t(\theta^2_t|\theta^1_t)}.$$

So,

- the value function of any behavioral strategy $\beta^1_{\tau:}$ is linear at t in $b_{\theta^1_t}$, thus (in particular) in $\sigma^{c,1}_t(\cdot|\theta^1_t)$; and
- the optimal value function is LC at t also in $b_{\theta^1_t}$ (with the same depth-dependent upper-bounding Lipschitz constant λ_t as in the proof of Theorem 3.1.12),[28] thus (in particular) in $\sigma^{c,1}_t(\cdot|\theta^1_t)$.

Using $t = \tau$, the optimal value function is $\nu^2_{[\sigma^{c,1}_\tau, \beta^2_{\tau:}]}(\theta^1_\tau)$, which is thus λ_τ-LC in $\sigma^{c,1}_\tau(\cdot|\theta^1_\tau)$. □

Proposition 3.1.16. *Let $\overline{\mathcal{I}}_\tau$ be a set of tuples $w = \langle \tilde{\sigma}_\tau, \beta^2_\tau, \langle \overline{\nu}^2_{\tau+1}, \beta^2_{\tau+1:}\rangle\rangle$, where vectors $\overline{\nu}^2_{\tau+1}$ are upper bounding particular vectors $\nu^2_{\tau+1}$ (details are given in the proof). Then,*

$$\overline{W}_\tau(\sigma_\tau, \beta^1_\tau) \stackrel{\text{def}}{=} \min_{\langle \tilde{\sigma}_\tau, \beta^2_\tau, \langle \overline{\nu}^2_{\tau+1}, \beta^2_{\tau+1:}\rangle\rangle \in \overline{\mathcal{I}}_\tau} \big[r(\sigma_\tau, \beta^1_\tau, \beta^2_\tau) + \gamma T^1_m(\sigma_\tau, \beta^1_\tau, \beta^2_\tau) \cdot \overline{\nu}^2_{\tau+1} + \lambda_{\tau+1} \|T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau) - T^1_m(\sigma_\tau, \beta^1_\tau, \beta^2_\tau) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau)\|_1 \big] \tag{3.16}$$

upper-bounds $W^{1,*}_\tau$ over the whole space $\mathcal{O}^\sigma_\tau \times \mathcal{B}^1_\tau$. Similarly, lower bounds $\underline{W}_\tau$ of $W^{2,*}_\tau$ can be defined.

Proof. Note that, since $V^*_H = 0$, $\tau = H-1$ is a particular case which can be simply re-written:

$$W^{1,*}_\tau(\sigma_\tau, \beta^1_\tau) = \min_{\beta^2_\tau} {\beta^1_\tau}^\top \cdot r(\sigma_\tau, \cdot, \beta^2_\tau).$$

To find a form that could be appropriate for an upper bound approximation of $W^{*,1}_\tau$, let us now consider an OS σ_τ and a single tuple $\langle \tilde{\sigma}_\tau, \tilde{\beta}^2_\tau, \nu^2_{[T^1_c(\tilde{\sigma}_\tau, \tilde{\beta}^2_\tau), \tilde{\beta}^2_{\tau+1:}]}\rangle$. Then,

$$W^{1,*}_\tau(\sigma_\tau, \beta^1_\tau) \tag{3.17}$$
$$= \min_{\beta^2_\tau} \big[r(\sigma_\tau, \boldsymbol{\beta}_\tau) + \gamma V^*_{\tau+1}(T(\sigma_\tau, \boldsymbol{\beta}_\tau)) \big] \tag{3.18}$$
$$\leqslant r(\sigma_\tau, \beta^1_\tau, \tilde{\beta}^2_\tau) + \gamma V^{BR,1}_{\tau+1}(T(\sigma_\tau, \beta^1_\tau, \tilde{\beta}^2_\tau)|\tilde{\beta}^2_{\tau+1:}) \qquad \text{(for any } \tilde{\beta}^2_{\tau:} \text{ in } \mathcal{I}) \tag{3.19}$$

(where $V^{BR,1}_{\tau+1}(T(\sigma_\tau, \beta^1_\tau, \tilde{\beta}^2_\tau)|\tilde{\beta}^2_{\tau+1:})$ is the value of 1's best response to $\tilde{\beta}^2_{\tau+1:}$ if in $T(\sigma_\tau, \beta^1_\tau, \tilde{\beta}^2_\tau)$)

$$= r(\sigma_\tau, \beta^1_\tau, \tilde{\beta}^2_\tau) + \gamma T^1_m(\sigma_\tau, \beta^1_\tau, \tilde{\beta}^2_\tau) \cdot \underbrace{\nu^2_{[T^1_c(\sigma^{c,1}_\tau, \tilde{\beta}^2_\tau), \tilde{\beta}^2_{\tau+1:}]}}_{} \tag{3.20}$$

[28] The proof process is similar. The only difference lies in the space at hand, but without any impact on the resulting formulas.

(Wiggers et al. 2016a, Lemma 3)

$$\leqslant r(\sigma_\tau, \beta^1_\tau, \tilde{\beta}^2_\tau) + \gamma T^1_m(\sigma_\tau, \beta^1_\tau, \tilde{\beta}^2_\tau) \cdot \left(\nu^2_{[T^1_c(\tilde{\sigma}^{c,1}_\tau, \tilde{\beta}^2_\tau), \tilde{\beta}^2_{\tau+1:}]} + \lambda_{\tau+1} \cdot \overrightarrow{\|T^1_c(\sigma^{c,1}_\tau, \tilde{\beta}^2_\tau) - T^1_c(\tilde{\sigma}^{c,1}_\tau, \tilde{\beta}^2_\tau)\|_1}\right) \tag{3.21}$$

(Lemma 3.1.15: $\lambda_{\tau+1}$-LC of $\nu^2_{[T^1_c(\sigma^{c,1}_\tau, \tilde{\beta}^2_\tau), \tilde{\beta}^2_{\tau+1:}]}$)

$$= {\beta^1_\tau}^\top \cdot \left[r(\sigma_\tau, \cdot, \tilde{\beta}^2_\tau) + \gamma T^1_m(\sigma_\tau, \cdot, \tilde{\beta}^2_\tau) \cdot \left(\nu^2_{[T^1_c(\tilde{\sigma}^{c,1}_\tau, \tilde{\beta}^2_\tau), \tilde{\beta}^2_{\tau+1:}]} + \lambda_{\tau+1} \cdot \overrightarrow{\|T^1_c(\sigma^{c,1}_\tau, \tilde{\beta}^2_\tau) - T^1_c(\tilde{\sigma}^{c,1}_\tau, \tilde{\beta}^2_\tau)\|_1}\right)\right] \tag{3.22}$$

(Linearity in β^1_τ)

$$\tag{3.23}$$

$$= {\beta^1_\tau}^\top \cdot \Big[r(\sigma_\tau, \cdot, \tilde{\beta}^2_\tau) + \gamma T^1_m(\sigma_\tau, \cdot, \tilde{\beta}^2_\tau) \cdot \nu^2_{[T^1_c(\tilde{\sigma}^{c,1}_\tau, \tilde{\beta}^2_\tau), \tilde{\beta}^2_{\tau+1:}]} \tag{3.24}$$

$$+\gamma\lambda_{\tau+1} \cdot \|T(\sigma_\tau, \cdot, \tilde{\beta}^2_\tau) - T^1_m(\sigma_\tau, \cdot, \tilde{\beta}^2_\tau) T^1_c(\tilde{\sigma}^{c,1}_\tau, \tilde{\beta}^2_\tau)\|_1\Big] \tag{3.25}$$

(Alternative writing)

$$\tag{3.26}$$

From this, we can deduce the following appropriate forms (i) of upper-bounding approximation for $W^{1,*}_\tau$ and (ii) (symmetrically) of lower-bounding approximation for $W^{2,*}_\tau$:

$$\begin{aligned}
\overline{W}_\tau(\sigma_\tau, \beta^1_\tau) = \min_{\langle \tilde{\sigma}^{c,1}_\tau, \beta^2_\tau, \overline{\nu}^2_{\tau+1} \rangle \in \overline{\mathcal{I}}_\tau} {\beta^1_\tau}^\top \cdot \Big[&r(\sigma_\tau, \cdot, \beta^2_\tau) + \gamma T^1_m(\sigma_\tau, \cdot, \beta^2_\tau) \cdot \overline{\nu}^2_{\tau+1} \\
&+ \gamma\lambda_{\tau+1} \cdot \|T(\sigma_\tau, \cdot, \beta^2_\tau) - T^1_m(\sigma_\tau, \cdot, \beta^2_\tau) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau)\|_1\Big], \text{ and} \\
\underline{W}_\tau(\sigma_\tau, \beta^2_\tau) = \max_{\langle \tilde{\sigma}^{c,2}_\tau, \beta^1_\tau, \underline{\nu}^1_{\tau+1} \rangle \in \underline{\mathcal{I}}_\tau} {\beta^2_\tau}^\top \cdot \Big[&r(\sigma_\tau, \beta^1_\tau, \cdot) + \gamma T^2_m(\sigma_\tau, \beta^1_\tau, \cdot) \cdot \underline{\nu}^1_{\tau+1} \\
&- \gamma\lambda_{\tau+1} \cdot \|T(\sigma_\tau, \beta^1_\tau, \cdot) - T^2_m(\sigma_\tau, \beta^1_\tau, \cdot) T^2_c(\tilde{\sigma}^{c,2}_\tau, \beta^1_\tau)\|_1\Big],
\end{aligned}$$

where $\overline{\nu}^2_{\tau+1}$ and $\underline{\nu}^1_{\tau+1}$ respectively upper- and lower-bound the actual vectors associated to the players' future strategies (resp. of 2 and 1).

Again, $\tau = H - 1$ is a particular case where only the reward term is preserved. □

As explained in the next two sections, $\overline{W}_\tau$ will be easier to deal with compared to $\overline{V}_\tau$, allowing 1 to seek for decision rules optimistically, and providing valid solution strategies for 2 for the subgame at τ, *i.e.,* ignoring consistency with higher-level subgames.

3.1.2.2 Action Selection and Backup Operators

We now describe the decision rule selection for 1 using $\overline{W}_\tau$ to optimistically guide a trajectory in occupancy space, and how to update $\overline{W}_\tau$ by providing backup operators.

First, note that linearities in β^1_τ within Eq. (3.16) allow writing $\overline{W}_\tau(\sigma_\tau, \beta^1_\tau) = \min_{w \in \overline{\mathcal{I}}_\tau} {\beta^1_\tau}^\top \cdot M^{\sigma_\tau}_{(\cdot,w)}$, where β^1_τ and $M^{\sigma_\tau}_{(\cdot,w)}$ (for each w) are column vectors of dimension $|\Theta^1 \times \mathcal{A}^1|$. M^{σ_τ} is thus a $|\Theta^1_\tau \times \mathcal{A}^1| \times |\overline{\mathcal{I}}_\tau|$ matrix. But then, the optimization problem $\max_{\beta^1_\tau} \min_{w \in \overline{\mathcal{I}}_\tau} {\beta^1_\tau}^\top \cdot M^{\sigma_\tau}_{(\cdot,w)}$ corresponds to the search for a Nash equilibrium strategy profile in a zero-sum Bayesian game (Definition 2.1.13). In this Bayesian game, player 1 has one type per history θ^1_τ, and 2 has a single type.

The following lemma details the payoff matrix $M^{\sigma_\tau}_{(\cdot,\cdot)}$ of this Bayesian game and formulates (according to Proposition 2.2.4, proposition 2.2.4) the search for Nash equilibrium strategies as an LP and its dual (Corollary 3.1.18).

Lemma 3.1.17. *Using now a distribution ψ^2_τ over tuples $w = \langle \tilde{\sigma}^{c,1}_\tau, \beta^2_\tau, \overline{\nu}^2_{\tau+1} \rangle \in \overline{\mathcal{I}}^1_\tau$, the corresponding upper-bounding value for "profile" $\langle \beta^1_\tau, \psi^2_\tau \rangle$ when in σ_τ can be written as an expectancy:*

$$\beta^{1\top}_\tau \cdot M^{\sigma_\tau} \cdot \psi^2_\tau,$$

where M^{σ_τ} is an $|\Theta^1_\tau \times \mathcal{A}^1| \times |\overline{\mathcal{I}}^1_\tau|$ matrix.

Proof. From the right-hand side term in Equation (3.22), the upper-bounding value associated to σ_τ, β^1_τ and a tuple $\langle \tilde{\sigma}^{c,1}_\tau, \beta^2_\tau, \overline{\nu}^2_{\tau+1} \rangle \in \overline{\mathcal{I}}^1_\tau$ can be written:

$$\beta^{1\top}_\tau \cdot \left[r(\sigma_\tau, \cdot, \beta^2_\tau) + \gamma T^1_m(\sigma_\tau, \cdot, \beta^2_\tau) \cdot \left(\overline{\nu}^2_{\tau+1} + \lambda_{\tau+1} \cdot \overrightarrow{\|T^1_c(\sigma^{c,1}_\tau, \tilde{\beta}^2_\tau) - T^1_c(\tilde{\sigma}^{c,1}_\tau, \tilde{\beta}^2_\tau)\|_1} \right) \right].$$

Using now a distribution ψ^2_τ over tuples $w = \langle \tilde{\sigma}^{c,1}_\tau, \beta^2_\tau, \overline{\nu}^2_{\tau+1} \rangle \in \overline{\mathcal{I}}^1_\tau$, the corresponding upper-bounding value for "profile" $\langle \beta^1_\tau, \psi^2_\tau \rangle$ when in σ_τ can be written as an expectancy:

$$\sum_{w \in \overline{W}_\tau} \beta^{1\top}_\tau \cdot \left[r(\sigma_\tau, \cdot, \beta^2_\tau[w]) + \gamma T^1_m(\sigma_\tau, \cdot, \beta^2_\tau[w]) \cdot \left(\overline{\nu}^2_{\tau+1}[w] + \lambda_{\tau+1} \cdot \overrightarrow{\|T^1_c(\sigma^{c,1}_\tau, \beta^2_\tau[w]) - T^1_c(\tilde{\sigma}^{c,1}_\tau[w], \beta^2_\tau[w])\|_1} \right) \right] \cdot \psi^2_\tau(w)$$

(where $x[w]$ denotes the field x of tuple w)

$$= \beta^{1\top}_\tau \cdot M^{\sigma_\tau} \cdot \psi^2_\tau,$$

where M^{σ_τ} is an $|\Theta^1_\tau \times \mathcal{A}^1| \times |\overline{\mathcal{I}}^1_\tau|$ matrix. □

For implementation purposes, using Eqs. (2.42) and (3.13) (to develop respectively $r(\cdot,\cdot,\cdot)$ and $T^1_m(\cdot,\cdot,\cdot)$), we can derive the expression of a component, *i.e.,* the upper-bounding value if a^1 is applied in θ^1_τ while w is chosen:

$$M^{\sigma_\tau}_{(\langle \theta^1_\tau, a^1 \rangle, w)} \tag{3.27}$$

$$\stackrel{\text{def}}{=} r(\sigma_\tau, \cdot, \beta^2_\tau[w]) + \gamma T^1_m(\sigma_\tau, \cdot, \beta^2_\tau[w]) \cdot \tag{3.28}$$

$$\left(\overline{\nu}^2_{\tau+1}[w] + \lambda_{\tau+1} \cdot \overrightarrow{\|T^1_c(\sigma^{c,1}_\tau, \beta^2_\tau[w]) - T^1_c(\tilde{\sigma}^{c,1}_\tau[w], \beta^2_\tau[w])\|_1} \right) \tag{3.29}$$

$$= \sum_{s, \theta^2_\tau, a^2} \sigma_\tau(\boldsymbol{\theta}_\tau) b(s|\boldsymbol{\theta}_\tau) \beta^2_\tau[w](a^2|\theta^2) r(s, \boldsymbol{a}) + \gamma \sum_{z^1} \left[\sum_{\theta^2_\tau, a^2} \beta^2_\tau[w](a^2|\theta^2_\tau) \sum_{s, s', z^2} P^{\boldsymbol{z}}_{\boldsymbol{a}}(s'|s) b(s|\boldsymbol{\theta}_\tau) \sigma_\tau(\boldsymbol{\theta}_\tau) \right] \cdot \tag{3.30}$$

$$\left(\overline{\nu}^2_{\tau+1}[w](\theta^1_\tau, a^1, z^1) + \lambda_{\tau+1} \cdot \overrightarrow{\|T^1_c(\sigma^{c,1}_\tau, \beta^2_\tau[w]) - T^1_c(\tilde{\sigma}^{c,1}_\tau[w], \beta^2_\tau[w])\|_1}(\theta^1_\tau, a^1, z^1) \right) \tag{3.31}$$

$$= \sum_{\theta^2_\tau} \sigma_\tau(\boldsymbol{\theta}_\tau) \sum_{a^2} \beta^2_\tau[w](a^2|\theta^2_\tau) \cdot \left(\sum_s b(s|\boldsymbol{\theta}_\tau) r(s, \boldsymbol{a}) + \gamma \sum_{z^1} \left[\sum_{s, s', z^2} P^{\boldsymbol{z}}_{\boldsymbol{a}}(s'|s) b(s|\boldsymbol{\theta}_\tau) \right] \cdot \left(\overline{\nu}^2_{\tau+1}[w](\theta^1_\tau, a^1, z^1) \right. \right. \tag{3.32}$$

$$\left. \left. + \lambda_{\tau+1} \cdot \overrightarrow{\|T^1_c(\sigma^{c,1}_\tau, \beta^2_\tau[w]) - T^1_c(\tilde{\sigma}^{c,1}_\tau[w], \beta^2_\tau[w])\|_1}(\theta^1_\tau, a^1, z^1) \right) \right). \tag{3.33}$$

Then, solving $\max_{\beta^1_\tau} \overline{W}_\tau(\sigma_\tau, \beta^1_\tau)$ can be rewritten as solving a zero-sum game where pure strategies are:

- for player 1, the choice of not 1, but $|\Theta^1_\tau|$ actions (among $|\mathcal{A}^1|$) and,
- for player 2, the choice of 1 element of $\overline{\mathcal{I}}^1_\tau$.

With our upper bound, $\max_{\beta^1_\tau} \overline{W}_\tau(\sigma_\tau, \beta^1_\tau)$ can thus be solved as a LP.

Corollary 3.1.18. *For any given σ_τ and any set $\overline{\mathcal{I}}_\tau$ of tuples $w = \langle \tilde{\sigma}_\tau, \beta^2_\tau, \langle \overline{\nu}^2_{\tau+1}, \beta^2_{\tau+1:} \rangle \rangle$, $\max_{\beta^1_\tau} \overline{W}_\tau(\sigma_\tau, \beta^1_\tau)$ is equivalent to the* LP *and dual LP:*

$$\begin{aligned}
&\text{LP}\overline{W}_\tau(\sigma_\tau) \;:\; \max_{\beta^1_\tau, v} v \quad s.t. \quad (i) \quad \forall w \in \overline{\mathcal{I}}_\tau, \quad v \leqslant {\beta^1_\tau}^\top \cdot M^{\sigma_\tau}_{(\cdot,w)} \qquad and \\
&\qquad\qquad\qquad\qquad\qquad\quad (ii) \quad \forall \theta^1_\tau \in \Theta^1_\tau, \quad \sum_{a^1} \beta^1_\tau(a^1|\theta^1_\tau) = 1, \\
&\text{DLP}\overline{W}_\tau(\sigma_\tau) \;:\; \min_{\psi^2_\tau, v} v \quad s.t. \quad (i) \quad \forall (\theta^1_\tau, a^1), \quad v \geqslant M^{\sigma_\tau}_{((\theta^1_\tau,a^1),\cdot)} \cdot \psi^2_\tau \quad and \\
&\qquad\qquad\qquad\qquad\qquad\quad (ii) \qquad\qquad\quad \sum_{w \in \overline{\mathcal{I}}_\tau} \psi^2_\tau(w) = 1.
\end{aligned} \tag{3.34}$$

Remark 3.1.19 (Interpretation of M^{σ_τ})**.** *The content of this matrix can be interpreted by noting that, a given w containing a behavioral strategy $\beta^2_{\tau:H-1}$ and an* OS *$\tilde{\sigma}_\tau$, a pair $\langle w, \theta^1_\tau \rangle$ induces a* POMDP *for player 1 whose state space is made of pairs $\langle s, \theta^2_t \rangle$, and whose initial belief b_τ depends on $\tilde{\sigma}_\tau$ and θ^1_τ. Solving this* POMDP *amounts to finding a best response of player 1 to $\beta^2_{\tau:H-1}$. In this setting, an element $M^{\sigma_\tau}_{((\theta^1_\tau,a^1),w)}$ is an upper-bound of the optimal (*POMDP*) Q-value when player 1 performs a^1 while facing b_τ ($Q^*_{\text{POMDP}}(b_\tau, a^1)$).*

As can be noted, M^{σ_τ}'s columns corresponding to 0-probability histories θ^1_τ in $\sigma^{m,1}_\tau$ are empty (full of zeros), so that the corresponding decision rules (for these histories) are not relevant and can be set arbitrarily. The actual implementation thus ignores these histories, whose corresponding decision rules also do not need to be stored.

Remark 3.1.20 (Outcomes of this game)**.** *Since $\overline{W}_\tau$ upper-bounds $W^{1,*}_\tau$, solving this* LP *provides* 1 *with an* optimistically *selected immediate decision rule β^1_τ. For* 2*, ψ^2_τ is a probability distribution over tuples containing strategies $\beta^2_\tau \oplus \beta^2_{\tau+1:H-1}$, thus recursively induces a strategy, as illustrated by Fig. 3.1, which can be turned into a behavioral strategy $\beta^2_{\tau:H-1}$ (more details in Appendix A.2), and whose value is* at worst *(from* 2*'s viewpoint) the* LP*'s value,* i.e., *against* 1*'s best response to it. These strategies are of particular interest as the ones obtained at σ_0 will later be used to obtain "safe" solution strategy profiles.*

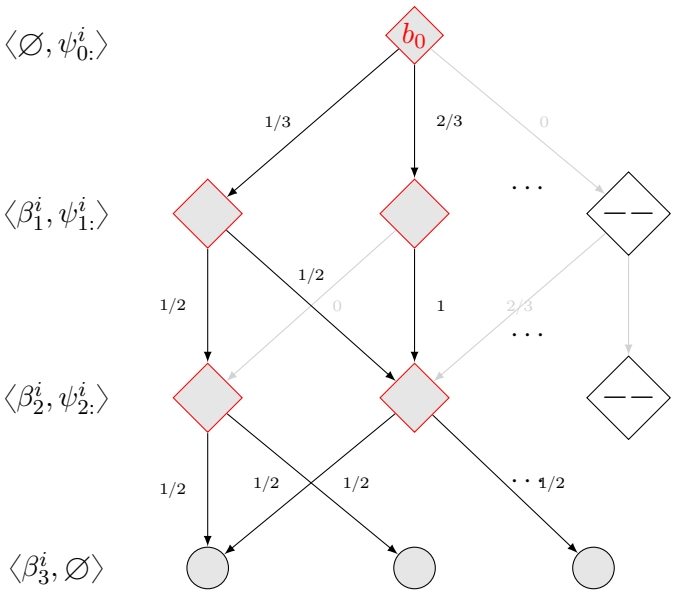


Figure 3.1: Representation of the strategy recursively induced by some ψ^1_0. At each time step τ, one must (i) sample a next tuple/node w^1_τ from current distribution ψ^1_τ, (ii) apply DR $\beta^1_\tau[w^1_\tau]$, and (iii) make $\psi^1_{\tau+1}[w^1_\tau]$ the new current distribution (unless reaching a leaf).

Then, the following properties allow performing backups, *i.e.,* filling up the set $\overline{\mathcal{I}}_{\tau-1}$ with new tuples w containing, in particular, vectors $\overline{\nu}^2_\tau$.

Proposition 3.1.21. *For each ψ^2_τ obtained as the solution of the aforementioned (dual) LP in σ_τ, and each θ^1_τ, the value $\nu^2_{[\sigma^{c,1}_\tau,\psi^2_\tau]}(\theta^1_\tau)$ is upper-bounded by a value $\overline{\nu}^2_\tau(\theta^1_\tau)$ that depends on vectors $\overline{\nu}^2_{\tau+1}$ in the support of ψ^2_τ. In particular, if $\theta^1_\tau \in \mathit{supp}(\sigma^{m,1}_\tau)$, we have:*

$$\overline{\nu}^2_\tau(\theta^1_\tau) \stackrel{\text{def}}{=} \frac{1}{\sigma^1_{\tau,m}(\theta^1_\tau)} \max_{a^1 \in \mathcal{A}^1} M^{\sigma_\tau}_{((\theta^1_\tau,a^1),.)} \cdot \psi^2_\tau.$$

Proof. For a newly derived ψ^2_τ, as $\nu^2_{[\sigma^{c,1}_\tau,\psi^2_\tau]}(\theta^1_\tau)$ is the value of 1's best action ($\in \mathcal{A}^1$) if (i) 1 observes θ^1_τ while in $\sigma^{c,1}_\tau$ and (ii) 2 plays ψ^2_τ, we have:

$$\begin{aligned}
&\nu^2_{[\sigma^{c,1}_\tau,\psi^2_\tau]}(\theta^1_\tau) \stackrel{\text{def}}{=} V^\star_{[\sigma^{c,1}_\tau,\psi^2_\tau]}(\theta^1_\tau) \qquad \text{(optimal POMDP value function)}\\
&= \max_{\beta^1_{\tau:}} \mathbb{E}\left[\sum_{t=\tau}^{H} \gamma^{t-\tau} R_t \mid \beta^1_{\tau:}, \theta^1_\tau, \sigma^{c,1}_\tau, \psi^2_\tau\right]\\
&= \max_{a^1} \mathbb{E}\left[R_\tau + \gamma \max_{\beta^1_{\tau+1:}} \mathbb{E}\left[\sum_{t=\tau+1}^{H} \gamma^{t-(\tau+1)} R_t \mid \beta^1_{\tau+1:}, \langle \theta^1_\tau, a^1, Z^1\rangle, \sigma^{c,1}_{\tau+1}, \psi^2_{\tau+1}\right] \,\middle|\, a^1, \theta^1_\tau, \sigma^{c,1}_\tau, \psi^2_\tau\right]\\
&= \max_{a^1} \mathbb{E}\left[R_\tau + \gamma V^\star_{[\sigma^{c,1}_{\tau+1},\psi^2_{\tau+1}]}(\theta^1_\tau, a^1, Z^1) \mid a^1, \theta^1_\tau, \sigma^{c,1}_\tau, \psi^2_\tau\right] \qquad (3.35)\\
&= \max_{a^1} \sum_{w,\theta^2_\tau,a^2,z^1} \underbrace{Pr(w, \theta^2_\tau, z^1, a^2 \mid a^1, \theta^1_\tau, \sigma^{c,1}_\tau, \psi^2_\tau)} \cdot \left(r(\boldsymbol{\theta}_\tau, \boldsymbol{a}_\tau) + \gamma \nu^2_{[\sigma^{c,1}_{\tau+1},\psi^2_{\tau+1}[w]]}(\theta^1_\tau, a^1, z^1)\right)
\end{aligned}$$

(where $\sigma^{c,1}_{\tau+1} = T^1_c(\sigma^{c,1}_\tau, \beta^2_\tau[w])$ (Lemma 3.1.9, p. 48))

$$\begin{aligned}
&= \max_{a_1} \sum_{w,\theta^2_\tau,a^2,z^1} \underbrace{Pr(w|\psi^2_\tau)} \cdot \underbrace{Pr(\theta^2_\tau|\theta^1_\tau, \sigma^{c,1}_\tau)} \cdot \underbrace{Pr(a^2|\beta^2_\tau[w], \theta^2_\tau)} \cdot \underbrace{Pr(z^1|\boldsymbol{\theta}_\tau, \boldsymbol{a}_\tau)}\\
&\quad \cdot \left(r(\boldsymbol{\theta}_\tau, \boldsymbol{a}) + \gamma \nu^2_{[\sigma^{c,1}_{\tau+1},\psi^2_{\tau+1}[w]]}(\theta^1_\tau, a^1, z^1)\right) \qquad (3.36)\\
&= \max_{a_1} \sum_w \psi^2_\tau(w) \sum_{\theta^2_\tau} \sigma^{c,1}_\tau(\theta^2_\tau|\theta^1_\tau) \sum_{a^2} \beta^2_\tau[w](a^2|\theta^2_\tau) \cdot \left(r(\boldsymbol{\theta}_\tau, \boldsymbol{a}) + \gamma \sum_{z^1} Pr(z^1|\boldsymbol{\theta}_\tau, \boldsymbol{a}) \underbrace{\nu^2_{[\sigma^{c,1}_{\tau+1},\psi^2_{\tau+1}[w]]}(\theta^1_\tau, a^1, z^1)}\right) \qquad (3.37)
\end{aligned}$$

then, as $\nu^2_{[\sigma^{c,1}_{\tau+1},\psi^2_{\tau+1}[w]]}$ is $\lambda_{\tau+1}$-LC in (any) $\sigma^{c,1}_{\tau+1}$ (Lemma 3.1.15),

$$\begin{aligned}
&\leqslant \max_{a_1} \sum_w \psi^2_\tau(w) \sum_{\theta^2_\tau} \sigma^{c,1}_\tau(\theta^2_\tau|\theta^1_\tau) \sum_{a^2} \beta^2_\tau[w](a^2|\theta^2_\tau) \cdot \left(r(\boldsymbol{\theta}_\tau, \boldsymbol{a}) + \gamma \sum_{z^1} Pr(z^1|\boldsymbol{\theta}_\tau, \boldsymbol{a})\right.\\
&\quad \left. \cdot \left[\overbrace{\underbrace{\nu^2_{[\tilde{\sigma}^{c,1}_{\tau+1}[w],\psi^2_{\tau+1}[w]]}(\theta^1_\tau, a^1, z^1)} + \lambda_{\tau+1} \overrightarrow{\|\sigma^{c,1}_{\tau+1} - \tilde{\sigma}^{c,1}_{\tau+1}[w]\|_1}(\theta^1_\tau, a^1, z^1)}\right]\right) \qquad (3.38)\\
&\leqslant \max_{a_1} \sum_w \psi^2_\tau(w) \sum_{\theta^2_\tau} \sigma^{c,1}_\tau(\theta^2_\tau|\theta^1_\tau) \sum_{a^2} \beta^2_\tau[w](a^2|\theta^2_\tau) \cdot \left(\underbrace{r(\boldsymbol{\theta}_\tau, \boldsymbol{a})} + \gamma \sum_{z^1} \underbrace{Pr(z^1|\boldsymbol{\theta}_\tau, \boldsymbol{a})}\right.\\
&\quad \left. \cdot \left[\overbrace{\overline{\nu}^2_{\tau+1}[w](\theta^1_\tau, a^1, z^1)} + \lambda_{\tau+1} \overrightarrow{\|\sigma^{c,1}_{\tau+1} - \tilde{\sigma}^{c,1}_{\tau+1}[w]\|_1}(\theta^1_\tau, a^1, z^1)\right]\right) \qquad (3.39)\\
&= \max_{a_1} \sum_w \psi^2_\tau(w) \sum_{\theta^2_\tau} \sigma^{c,1}_\tau(\theta^2_\tau|\theta^1_\tau) \sum_{a^2} \beta^2_\tau[w](a^2|\theta^2_\tau) \cdot \left(\overbrace{\sum_s b(s|\boldsymbol{\theta}_\tau) r(s, \boldsymbol{a})}\right.\\
&\quad \left. + \gamma \sum_{z^1} \left(\overbrace{\sum_s b(s|\boldsymbol{\theta}_\tau) \underbrace{Pr(z^1|s, \boldsymbol{a})}}\right) \cdot \left[\overline{\nu}^2_{\tau+1}[w](\theta^1_\tau, a^1, z^1) + \lambda_{\tau+1} \underbrace{\overrightarrow{\|\sigma^{c,1}_{\tau+1} - \tilde{\sigma}^{c,1}_{\tau+1}[w]\|_1}(\theta^1_\tau, a^1, z^1)}\right]\right)\\
&= \max_{a_1} \sum_w \psi^2_\tau(w) \sum_{\theta^2_\tau} \sigma^{c,1}_\tau(\theta^2_\tau|\theta^1_\tau) \sum_{a^2} \beta^2_\tau[w](a^2|\theta^2_\tau) \quad \cdot \left(\sum_s b(s|\boldsymbol{\theta}_\tau) r(s, \boldsymbol{a}) + \gamma \sum_{z^1} \left(\overbrace{\sum_{s,s',z^2} b(s|\boldsymbol{\theta}_\tau) P^{\boldsymbol{z}}_{\boldsymbol{a}}(s'|s)}\right)\right.
\end{aligned}$$

$$\cdot \left[\overline{\nu}^2_{\tau+1}[w](\theta^1_\tau, a^1, z^1) + \lambda_{\tau+1} \overrightarrow{\|T^1_c(\sigma^{c,1}_\tau, \beta^2_\tau[w]) - T^1_c(\tilde{\sigma}^{c,1}_\tau[w], \beta^2_\tau[w])\|_1}(\theta^1_\tau, a^1, z^1)\right]\Bigg) \tag{3.40}$$

(recognizing Equation (3.33))

$$= \frac{1}{\sigma^1_{\tau,m}(\theta^1_\tau)} \max_{a^1 \in \mathcal{A}^1} M^{\sigma_\tau}_{((\theta^1_\tau, a^1),.)} \cdot \psi^2_\tau.$$

□

Corollary 3.1.22 (update)**.** *Let us assume that*

- *a transition $\sigma_{\tau-1} \to \sigma_\tau$ has been performed through playing $\langle \beta^1_{\tau-1}, \beta^2_{\tau-1} \rangle$, and*
- *solving* DLP$\overline{W}_\tau(\sigma_\tau)$ *provides both*
  - *a tree strategy ψ^2_τ (as the main solution of the DLP), and*
  - *a vector $\overline{\nu}^2_\tau = \frac{1}{\sigma^{m,1}_\tau} M^{\sigma_\tau}_{((\theta^1_\tau, a^1),.)} \cdot \psi^2_\tau$ (as a by-product).*

Then,

1. *$\overline{\mathcal{I}}_{\tau-1} \leftarrow \overline{\mathcal{I}}_{\tau-1} \cup \{\langle \sigma^{c,1}_{\tau-1}, \beta^2_{\tau-1}, \langle \overline{\nu}^2_\tau, \psi^2_\tau \rangle \rangle\}$ is a valid update operator in the sense that it preserves $\overline{W}_\tau$'s upper-bounding property, and*
2. *similarly, $\overline{\mathcal{J}}_\tau \leftarrow \overline{\mathcal{J}}_\tau \cup \{\langle \sigma^{c,1}_\tau, \langle \overline{\nu}^2_\tau, \psi^2_\tau \rangle \rangle\}$ is a valid update operator for $\overline{V}_\tau$.*

3.1.2.3 Initialization

To initialize the bounds $\overline{W}_\tau$ and $\overline{V}_\tau$ for any time step, we begin by generating a trajectory in a forward phase. At each time step, a uniform decision rule is picked for both players to derive a sequence of occupancy states $\sigma_0, \dots, \sigma_{H-1}$. Then, during a backward phase, for each time step $\tau = H-1, \dots, 1$, we create a tuple $w_{\tau-1,init} = \langle \sigma^{c,1}_{\tau-1}, \beta^2_{\tau-1}, \langle \overline{\nu}^2_\tau, \psi^2_\tau \rangle \rangle$, where

- $\sigma^{c,1}_{\tau-1}$ is the conditional term associated to $\sigma_{\tau-1}$;
- $\beta^2_{\tau-1}$ is a uniform decision rule;
- ψ^2_τ is
  - a degenerate distribution over the only next tuple $w_{\tau+1}$ if $\tau < H-1$ (which induces a concatenation of uniform decision rules for all future time steps), and
  - undefined if $\tau = H-1$;

  and
- $\overline{\nu}^2_\tau(\theta^1_\tau) = r_{max} \cdot (H - \tau)$ for any history θ^1_τ that player 1 could face.

Tuples $w_{\tau-1,init}$ are added to sets $\overline{\mathcal{I}}_{\tau-1}$. For any time step $\tau \geqslant 0$, we similarly create tuples $\langle \sigma^{c,1}_\tau, \langle \overline{\nu}^2_\tau, \psi^2_{\tau:} \rangle \rangle$ and add them to sets $\overline{\mathcal{J}}_\tau$. The lower bounds are initialized symmetrically.

We now show that occupancy states can also be prescriptive, allowing one to retrieve an ϵ-NES for the subgame at occupancy state σ_τ once the bounds are within ϵ from each other, in particular at $\tau = 0$.

3.1.2.4 Retrieving a NES

As already mentioned, vectors $\overline{\nu}_0^2$ upper bound the value of their associated tree strategies. This allows determining when and how to extract an ϵ-optimal solution strategy for any of the players, as detailed now.

Theorem 3.1.23 (Retrieving a NES for the zs-POSG)**.** *If sets $\overline{\mathcal{J}}_0$ and $\underline{\mathcal{J}}_0$ are such that $\overline{V}_0(\sigma_0) - \underline{V}_0(\sigma_0) \leqslant \epsilon$, then $\arg\max_{\underline{w} \in \underline{\mathcal{J}}_0} \underline{\nu}_0^2$ and $\arg\min_{\overline{w} \in \overline{\mathcal{J}}_0} \overline{\nu}_0^1$ respectively provide strategies ψ_0^1 and ψ_0^2 that form an ϵ-NES of the zs-POSG.*

Proof. First, let us notice that, at $\tau = 0$, the occupancy-state space is reduced to a singleton, $\{\sigma_0 = \langle 1 \rangle\}$, because of the single (empty) joint AOH. The value vectors ν are thus one-dimensional, and here considered as scalar numbers.

Let us assume that sets $\overline{\mathcal{J}}_0$ and $\underline{\mathcal{J}}_0$ are such that

$$\overline{V}_0(\sigma_0) - \underline{V}_0(\sigma_0) \leqslant \epsilon,$$

and let $\underline{w}^* = \langle \sigma_0^{c,1}, \langle \underline{\nu}_0^*, \psi_0^{1,*} \rangle \rangle$ and $\overline{w}^* = \langle \sigma_0^{c,1} \langle \overline{\nu}_0^*, \psi_0^{2,*} \rangle \rangle$ be the tuples returned by $\arg\max_{\underline{w} \in \underline{\mathcal{J}}_0} \underline{\nu}_0^2$ and $\arg\min_{\overline{w} \in \overline{\mathcal{J}}_0} \overline{\nu}_0^1$. Then, noting that $\sigma_0 = \langle 1 \rangle$,

$$\begin{aligned} \nu^1_{[\sigma_0^{c,2}, \psi_0^{1,*}]} - \nu^2_{[\sigma_0^{c,1}, \psi_0^{2,*}]} &\leqslant \overline{\nu}_0^* - \underline{\nu}_0^* \\ &= \max_{\underline{w} \in \underline{\mathcal{J}}_0} \underline{\nu}_0^2 - \min_{\overline{w} \in \overline{\mathcal{J}}_0} \overline{\nu}_0^1 \\ &= \overline{V}_0(\sigma_0) - \underline{V}_0(\sigma_0) \\ &\leqslant \epsilon. \end{aligned}$$

Thus, ψ_0^1 and ψ_0^2 are two strategies whose security levels (values against best-responding opponents) are ϵ-close, and thus form an ϵ-NES of the zs-POSG. □

Note: This result can be generalized to any σ_τ at later time steps, but this generalization is not used in practice.

Distributions ψ_0^2 are stored and can be executed as is. Appendix A.2 still presents a conversion process to retrieve a behavioral strategy $\beta^2_{0:H-1}$ from a distribution ψ_0^2 over tuples $w \in \overline{\mathcal{I}}_0$. Next, we see how to design a practical HSVI-based algorithm that provably returns sets $\overline{\mathcal{J}}_0$ and $\underline{\mathcal{J}}_0$ satisfying Theorem 3.1.23 after finitely many iterations.

3.1.3 HSVI for zs-POSGs

This section details our adaptation of the general HSVI scheme for ϵ-optimally solving zs-POSGs, and presents a theoretical finite-time convergence property.

3.1.3.1 Algorithm

HSVI for zs-POSGs is described in Algorithm 3.1. As vanilla HSVI, it relies on

1. generating trajectories while acting optimistically (lines 10+11), *i.e.,* player 1 (resp. 2) acting "greedily" w.r.t. $\overline{W}_\tau$ (resp. $\underline{W}_\tau$), and
2. locally updating the upper and lower bounds (lines 17+18).

Both phases rely on solving the same games described by LP (3.34). At $\tau = H-1$, line 14 selects DRs by solving an exact game. line 15.

A key difference with Smith et al.'s (2005) HSVI algorithm lies in the criterion for stopping trajectories. The branching factor for zs-oMGs being infinite, we make use of V^*'s Lipschitz-continuity to implement the same adaptations as Horák et al. (2017) used for zs-OS-POSGs. The Lipschitz-continuity allows controlling the variations of the value function within small balls of radius ρ around a previously visited occupancy state. A finite number of such balls is sufficient to cover the whole space. Then, Theorem 3.1.29 (below) ensures ϵ-optimality in finite time if stopping trajectories when $\overline{V}_\tau(\sigma_\tau) - \underline{V}_\tau(\sigma_\tau) \leqslant thr(\tau)$, with the threshold function $thr(\tau) \stackrel{\text{def}}{=} \gamma^{-\tau}\epsilon - \sum_{i=1}^{\tau} 2\rho\lambda_{\tau-i}\gamma^{-i}$ if $\gamma < 1$, and $0 < \rho < \frac{\epsilon}{(r_{\max} - r_{\min})(H+1)H}$ if $\gamma = 1$.

Algorithm 3.1: `HSVI`$(b_0, [\epsilon, \rho])$ [here returning a tuple w_0 containing a solution strategy for player 1]

Input : σ_0 a state
Fct `Solve`$(b_0 \simeq \sigma_0)$
  foreach $\tau \in 0\,..\,H-1$ **do**
    Initialize $\overline{V}_\tau$, $\underline{V}_\tau$, $\overline{W}_\tau$, & $\underline{W}_\tau$
  while $\left[\overline{V}_0(\sigma_0) - \underline{V}_0(\sigma_0) > thr(0)\right]$ **do**
    `Explore`$(\sigma_0, 0, -, -)$
  return $\arg\max_{w_0 \in \underline{\mathcal{J}}_0} \underline{\nu}_0^1$

Fct `Explore`$(\sigma_\tau, \tau, \sigma_{\tau-1}, \boldsymbol{\beta}_{\tau-1})$
  if $\left[\max_{\beta_\tau^1} \overline{W}_\tau(\sigma_\tau, \beta_\tau^1) - \min_{\beta_\tau^2} \underline{W}_\tau(\sigma_\tau, \beta_\tau^2) > thr(\tau)\right]$ **then**
    if $\tau < H-1$ **then**
      $\overline{\beta}_\tau^1 \leftarrow \mathrm{LP}\overline{W}_\tau(\sigma)$
      $\underline{\beta}_\tau^2 \leftarrow \mathrm{LP}\underline{W}_\tau(\sigma)$
      `Explore`$(T(\sigma_\tau, \overline{\beta}_\tau^1, \underline{\beta}_\tau^2), \tau+1, \sigma_\tau, \langle \overline{\beta}_\tau^1, \underline{\beta}_\tau^2 \rangle)$
    else $(\tau = H-1)$
      $(\overline{\beta}_\tau^1, \underline{\beta}_\tau^2) \leftarrow$ `NES` $\left(r(\sigma, \beta_\tau^1, \beta_\tau^2)\right)$
      $\overline{\mathcal{I}}_\tau^1 \leftarrow \overline{\mathcal{I}}_\tau^1 \cup \{\langle \sigma_\tau^{c,1}, \underline{\beta}_\tau^2, - \rangle\}$
      $\underline{\mathcal{I}}_\tau^2 \leftarrow \underline{\mathcal{I}}_\tau^2 \cup \{\langle \sigma_\tau^{c,2}, \overline{\beta}_\tau^1, - \rangle\}$
    $\overline{\texttt{Update}}(\overline{W}_{\tau-1}, \langle \sigma_\tau, \sigma_{\tau-1}^{c,1}, \underline{\beta}_{\tau-1}^2 \rangle)$
    $\underline{\texttt{Update}}(\underline{W}_{\tau-1}, \langle \sigma_\tau, \sigma_{\tau-1}^{c,2}, \overline{\beta}_{\tau-1}^1 \rangle)$

Fct $\overline{\texttt{Update}}(\overline{W}_{\tau-1}, \langle \sigma_\tau, \sigma_{\tau-1}^{c,1}, \overline{\beta}_{\tau-1}^2 \rangle)$
  $\langle \overline{\nu}_\tau^2, \psi_\tau^2 \rangle \leftarrow \mathrm{DLP}\overline{W}_\tau(\sigma_\tau,)$
  $\overline{\mathcal{I}}_{\tau-1} \leftarrow \overline{\mathcal{I}}_{\tau-1} \cup \{\langle \sigma_{\tau-1}^{c,1}, \underline{\beta}_{\tau-1}^2, \langle \overline{\nu}_\tau^2, \psi_\tau^2 \rangle \rangle\}$
  $\overline{\mathcal{J}}_\tau \leftarrow \overline{\mathcal{J}}_\tau \cup \{\langle \sigma_\tau^{c,1}, \langle \overline{\nu}_\tau^2, \psi_\tau^2 \rangle \rangle\}$

Setting ρ As can be observed, this threshold function should always return positive values, which requires a small enough (but > 0, as ρ will later correspond to radius of balls for $\|\cdot\|_1$). For a given problem, the maximum possible value $\rho_{\max}$ depends on the Lipschitz constants at each time step, which themselves depend on the initial upper and lower bounds of the optimal value function.

Proposition 3.1.24. *Bounding λ_τ by $\lambda^\infty = \frac{1}{2}\frac{1}{1-\gamma}\left[r_{\max} - r_{\min}\right]$ when $\gamma < 1$, and noting that*

$$thr(\tau) \geqslant \gamma^{-\tau}\epsilon - 2\rho\lambda^\infty \frac{\gamma^{-\tau}-1}{1-\gamma} \quad \text{if } \gamma < 1 \tag{3.41}$$
$$\left(\geqslant \epsilon - \frac{1}{2}\rho(r_{\max} - r_{\min})(2H+1-\tau)\tau \quad \text{if } \gamma = 1\right),$$

one can ensure positivity (and inferiority to $\gamma^{-\tau}\epsilon$) of the threshold at any $\tau \in 1\,..\,H-1$ by enforcing $0 < \rho < \frac{1-\gamma}{2\lambda^\infty}\epsilon$ (or $0 < \rho < \frac{2\epsilon}{(r_{\max}-r_{\min})(H+1)H}$ if $\gamma = 1$).

Proof. Let us first consider the case $\gamma < 1$.
We have (for $\tau \in \{1\,..\,H-1\}$):

$$\begin{aligned} thr(\tau) &\geqslant \gamma^{-\tau}\epsilon - \sum_{i=1}^{\tau} 2\rho\lambda^\infty\gamma^{-i} \\ &= \gamma^{-\tau}\epsilon - 2\rho\lambda^\infty \sum_{i=1}^{\tau}\gamma^{-i} \end{aligned}$$

$$
\begin{aligned}
&= \gamma^{-\tau}\epsilon - 2\rho\lambda^{\infty}\left(\gamma^{-1} + \gamma^{-2} + \cdots + \gamma^{-\tau}\right) \\
&= \gamma^{-\tau}\epsilon - 2\rho\lambda^{\infty}\gamma^{-1}\left(\gamma^{0} + \gamma^{-1} + \cdots + \gamma^{-(\tau-1)}\right) \\
&= \gamma^{-\tau}\epsilon - 2\rho\lambda^{\infty}\gamma^{-1}\frac{\gamma^{-\tau}-1}{\gamma^{-1}-1} \\
&= \gamma^{-\tau}\epsilon - 2\rho\lambda^{\infty}\frac{\gamma^{-\tau}-1}{1-\gamma}.
\end{aligned}
$$

Then, let us derive the following implications:

$$
\begin{aligned}
0 < thr(\tau) &\impliedby 2\rho\lambda^{\infty}\frac{\gamma^{-\tau}-1}{1-\gamma} < \gamma^{-\tau}\epsilon \\
&\Leftrightarrow \rho < \frac{1}{2\lambda^{\infty}}\frac{1-\gamma}{\gamma^{-\tau}-1}\gamma^{-\tau}\epsilon \\
&\Leftrightarrow \rho < \frac{1}{2\lambda^{\infty}}\frac{1-\gamma}{1-\gamma^{\tau}}\epsilon.
\end{aligned}
$$

To ensure positivity of the threshold for any $\tau \geqslant 1$, one thus just needs to set ρ as a positive value smaller than $\frac{1-\gamma}{2\lambda^{\infty}}\epsilon$.

Let us now consider the case $\gamma = 1$.
We have (for $\tau \in \{1, \ldots, H-1\}$):

$$
\begin{aligned}
thr(\tau) &\overset{\text{def}}{=} \epsilon - \sum_{i=1}^{\tau} 2\rho\lambda_{\tau-i} \\
&\geqslant \epsilon - \sum_{i=1}^{\tau} \rho(H-(\tau-i)) \cdot (r_{\max} - r_{\min}) \\
&= \epsilon - \rho(r_{\max} - r_{\min})\left[\tau(H-\tau) + \sum_{i=1}^{\tau} i\right] \\
&= \epsilon - \rho(r_{\max} - r_{\min})\left[\tau H - \tau^2 + \frac{1}{2}\tau(\tau+1)\right] \\
&= \epsilon - \rho(r_{\max} - r_{\min})\left[(H+\frac{1}{2})\tau - \frac{1}{2}\tau^2\right] \\
&= \epsilon - \frac{1}{2}\rho(r_{\max} - r_{\min})\left[(2H+1)\tau - \tau^2\right] \\
&= \epsilon - \frac{1}{2}\rho(r_{\max} - r_{\min})\left[(2H+1-\tau)\tau\right].
\end{aligned}
$$

Then, let us derive the following equivalent inequalities:

$$
0 < thr(\tau) \impliedby \rho(r_{\max} - r_{\min})(2H+1-\tau)\tau < 2\epsilon
$$

(holds when $\tau = 0$ and $\tau = H+1$)

$$
\Leftrightarrow \rho < \frac{2\epsilon}{(r_{\max} - r_{\min})(2H+1-\tau)\tau}
$$

(when $\tau \in \{0\,..\,H+1\}$).

The function $f : \tau \mapsto \frac{2\epsilon}{(r_{\max}-r_{\min})(2H+1-\tau)\tau}$ reaches its minimum (for $\tau \in (0, H+1)$) when $\tau = H + \frac{1}{2}$. To ensure positivity of the threshold for any $\tau \in \{1\,..\,H-1\}$, one thus just needs to set ρ as a positive value smaller than $\frac{2\epsilon}{(r_{\max}-r_{\min})(H+1)H}$. □

Setting $\rho \in (0, \rho_{\max})$ means making a trade-off between generating many trajectories (small ρ) and long ones (large ρ).

3.1.3.2 Finite-Time Convergence

Proving the finite-time convergence of HSVI to an error-bounded solution requires some preliminary lemmas.

Lemma 3.1.25. *Let $(\sigma_0, \dots, \sigma_{\tau+1})$ be a full trajectory generated by HSVI and $\boldsymbol{\beta}_\tau$ the behavioral* DR *profile that induced the last transition,* i.e., $\sigma_{\tau+1} = T(\sigma_\tau, \boldsymbol{\beta}_\tau)$. *Then, after updating $\overline{W}_\tau$ and $\underline{W}_\tau$, we have that $\overline{W}_\tau(\sigma_\tau, \beta^1_\tau) - \underline{W}_\tau(\sigma_\tau, \beta^2_\tau) \leqslant \gamma thr(\tau+1)$.*

Proof. By definition,

$$\overline{W}_\tau(\sigma_\tau, \beta^1_\tau) = \min_{\substack{\langle \tilde{\sigma}^{c,1}_\tau, \tilde{\beta}^2_\tau, \langle \overline{\nu}^2_{\tau+1}, \psi^2_{\tau+1:} \rangle \rangle \\ \in \overline{\mathcal{I}}^1_\tau}} \beta^1_\tau \cdot \Big(r(\sigma_\tau, \cdot, \tilde{\beta}^2_\tau) + \gamma T^1_m(\sigma_\tau, \cdot, \tilde{\beta}^2_\tau) \cdot \Big[\overline{\nu}^2_{\tau+1} + \lambda_{\tau+1} \overrightarrow{\|T^1_c(\sigma^{c,1}_\tau, \beta^2_\tau) - T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau)\|_1} \Big] \Big).$$

Therefore, after the update (β^2_τ and β^1_τ being added to their respective bags ($\overline{\mathcal{I}}^1_\tau$ and $\underline{\mathcal{I}}^2_\tau$) along with vectors $\overline{\nu}^2_{\tau+1}$ and $\underline{\nu}^1_{\tau+1}$),

$$\begin{aligned} \overline{W}_\tau(\sigma_\tau, \beta^1_\tau) &\leqslant \beta^1_\tau \cdot \left[r(\sigma_\tau, \cdot, \beta^2_\tau) + \gamma T^1_m(\sigma_\tau, \cdot, \beta^2_\tau) \cdot \overline{\nu}^2_{\tau+1} \right], \text{ and} \\ \underline{W}_\tau(\sigma_\tau, \beta^2_\tau) &\geqslant \beta^2_\tau \cdot \left[r(\sigma_\tau, \beta^1_\tau, \cdot) + \gamma T^2_m(\sigma_\tau, \beta^1_\tau, \cdot) \cdot \underline{\nu}^1_{\tau+1} \right]. \end{aligned}$$

Then,

$$\begin{aligned} \overline{W}_\tau(\sigma_\tau, \beta^1_\tau) - \underline{W}_\tau(\sigma_\tau, \beta^2_\tau) \leqslant & \left[\cancel{r(\sigma_\tau, \beta^1_\tau, \beta^2_\tau)} + \gamma T^1_m(\sigma_\tau, \boldsymbol{\beta}_\tau) \cdot \overline{\nu}^2_{\tau+1} \right] \\ & - \left[\cancel{r(\sigma_\tau, \beta^1_\tau, \beta^2_\tau)} + \gamma T^2_m(\sigma_\tau, \boldsymbol{\beta}_\tau) \cdot \underline{\nu}^1_{\tau+1} \right] \end{aligned}$$

↓ from Proposition 3.1.21

$$\begin{aligned} &= \gamma \left[\max_{\beta^1_\tau} \overline{W}_{\tau+1}(T(\sigma_\tau, \boldsymbol{\beta}_\tau)) - \min_{\beta^2_\tau} \underline{W}_{\tau+1}(T(\sigma_\tau, \boldsymbol{\beta}_\tau)) \right] \\ &\leqslant \gamma thr(\tau+1) \qquad \text{(Holds at the end of any trajectory.)} \end{aligned}$$

□

Lemma 3.1.26 (Monotonic evolution of upper and lower approximations W)**.** *Let $K\overline{W}_\tau$ and $K\underline{W}_\tau$ be the approximations after an update at σ_τ with behavioral* DR *$\langle \overline{\beta}^1_\tau, \underline{\beta}^2_\tau \rangle$ (respectively associated to vectors $\overline{\nu}^2_{\tau+1}$ and $\underline{\nu}^1_{\tau+1}$). Let also $K^{(n+1)}\overline{W}_\tau$ and $K^{(n+1)}\underline{W}_\tau$ be the same approximations after n other updates (in various OSs). Then,*

$$\begin{aligned} &\max_{\beta^1_\tau} K^{(n+1)}\overline{W}_\tau(\sigma_\tau, \beta^1_\tau) \leqslant \max_{\beta^1_\tau} K\overline{W}_\tau(\sigma_\tau, \beta^1_\tau) \leqslant \overline{W}_\tau(\sigma_\tau, \overline{\beta}^1_\tau) \quad \textit{and} \\ &\min_{\beta^2_\tau} K^{(n+1)}\underline{W}_\tau(\sigma_\tau, \beta^2_\tau) \geqslant \min_{\beta^2_\tau} K\underline{W}_\tau(\sigma_\tau, \beta^2_\tau) \geqslant \underline{W}_\tau(\sigma_\tau, \underline{\beta}^2_\tau). \end{aligned}$$

Proof. Starting from the definition,

$$\begin{aligned} \max_{\beta^1_\tau} K\overline{W}_\tau(\sigma_\tau, \beta^1_\tau) = \max_{\beta^1_\tau} \min_{\substack{\langle \tilde{\sigma}^{c,1}_\tau, \beta^2_\tau, \langle \overline{\nu}^2_{\tau+1}, \psi^2_{\tau+1:} \rangle \rangle \in \\ \overline{\mathcal{I}}^1_\tau \cup \{\langle \sigma^{c,1}_\tau, \underline{\beta}^2_\tau, \langle \overline{\nu}^2_{\tau+1}, \psi^2_{\tau+1:} \rangle \rangle\}}} & \beta^1_\tau \cdot \Big[r(\sigma_\tau, \cdot, \beta^2_\tau) \\ & + \gamma T^1_m(\sigma_\tau, \cdot, \beta^2_\tau) \cdot \Big(\overline{\nu}^2_{\tau+1} + \lambda_{\tau+1} \overrightarrow{\|T^1_c(\sigma^{c,1}_\tau, \beta^2_\tau) - T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau)\|_1} \Big) \Big] \\ \leqslant \max_{\beta^1_\tau} \min_{\langle \tilde{\sigma}^{c,1}_\tau, \beta^2_\tau, \langle \overline{\nu}^2_{\tau+1}, \psi^2_{\tau+1:} \rangle \rangle \in \overline{\mathcal{I}}^1_\tau} & \beta^1_\tau \cdot \Big[r(\sigma_\tau, \cdot, \beta^2_\tau) \\ & + \gamma T^1_m(\sigma_\tau, \cdot, \beta^2_\tau) \cdot \Big(\overline{\nu}^2_{\tau+1} + \lambda_{\tau+1} \overrightarrow{\|T^1_c(\sigma^{c,1}_\tau, \beta^2_\tau) - T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau)\|_1} \Big) \Big] \end{aligned}$$

$$= \max_{\beta_\tau^1} \overline{W}_\tau(\sigma_\tau, \beta_\tau^1)$$
$$= \overline{W}_\tau(\sigma_\tau, \overline{\beta}_\tau^1).$$

Then, this upper bound approximation can only be refined, so that, for any $n \in \mathbb{N}$,

$$\forall \beta_\tau^1, \quad K^{(n+1)}\overline{W}_\tau(\sigma_\tau, \beta_\tau^1) \leqslant K\overline{W}_\tau(\sigma_\tau, \beta_\tau^1),$$
$$\text{thus, } \max_{\beta_\tau^1} K^{(n+1)}\overline{W}_\tau(\sigma_\tau, \beta_\tau^1) \leqslant \max_{\beta_\tau^1} K\overline{W}_\tau(\sigma_\tau, \beta_\tau^1).$$

The expected result thus holds for $\overline{W}_\tau$, and symmetrically for $\underline{W}_\tau$. □

Lemma 3.1.27. *After updating, in order, $\overline{W}_\tau$ and $\overline{V}_\tau$, we have*

$$K\overline{V}_\tau(\sigma_\tau) \leqslant \max_{\beta_\tau^1} K\overline{W}_\tau(\sigma_\tau, \beta_\tau^1).$$

After updating, in order, $\underline{W}_\tau$ and $\underline{V}_\tau$, we have

$$K\underline{V}_\tau(\sigma_\tau) \geqslant \min_{\beta_\tau^2} K\underline{W}_\tau(\sigma_\tau, \beta_\tau^2).$$

Proof. After updating $\overline{\mathcal{I}}_\tau^1$, the algorithm computes (Algorithm 3.1, line 20) a new solution $\overline{\psi}_\tau^2$ of the dual LP (at σ_τ^1) and the associated vector $\overline{\nu}_\tau^2$, so that

$$\max_{\beta_\tau^1} K\overline{W}_\tau(\sigma_\tau^1, \beta_\tau^1) = \sigma_\tau^{m,1} \cdot \overline{\nu}_\tau^2.$$

This vector will feed $\overline{J}_\tau$ along with σ_τ^1, so that

$$K\overline{V}_\tau(\sigma_\tau) \leqslant \sigma_\tau^{m,1} \cdot \overline{\nu}_\tau^2.$$

As a consequence,

$$K\overline{V}_\tau(\sigma_\tau) \leqslant \max_{\beta_\tau^1} K\overline{W}_\tau(\sigma_\tau, \beta_\tau^1).$$

The symmetric property holds for $K\underline{V}_\tau$ and $K\underline{W}_\tau$, which concludes the proof. □

Lemma 3.1.28. *The function $\sigma_\tau \mapsto \max_{\beta_\tau^1} \overline{W}(\sigma_\tau, \beta_\tau^1)$ is $(3+\lambda)\eta$-Lipschitz-continuous*

Proof. Let $\beta_\tau^{1,*} \in \arg\max_{\beta_\tau^1} \overline{W}(\sigma_\tau, \beta_\tau^1)$ and let us denote by w an element of $\arg\min_w \beta_\tau^{1,*} \cdot M_{\cdot,w}^{\sigma_\tau}$. By definition,

$$\max_{\beta_\tau^1} \overline{W}_\tau^1(\sigma_\tau', \beta_\tau^1) \tag{3.42}$$
$$\leqslant \max_{\beta_\tau^1} r(\sigma_\tau', \beta_\tau^1, \beta_\tau^2[w]) + \gamma T_m^1(\sigma_\tau', \beta_\tau^1, \beta_\tau^2[w]) \cdot \overline{\nu}_{\tau+1}^2[w] \tag{3.43}$$
$$+ \gamma\lambda_{\tau+1} \cdot \| T(\sigma_\tau', \beta_\tau^1, \beta_\tau^2[w]) - T_m^1(\sigma_\tau', \beta_\tau^1, \beta_\tau^2[w]) \underbrace{T_c^1(\tilde{\sigma}_\tau^{c,1}, \beta_\tau^2[w])}_{\perp\!\!\!\perp \sigma_\tau'} \|_1 \Big] \tag{3.44}$$

(r is $\lambda \stackrel{\text{def}}{=} (r_{max} - r_{min})/2$-Lipschitz, and T_m^1 is 1-Lipschitz)

$$\leqslant \max_{\beta_\tau^1} r(\sigma_\tau, \beta_\tau^1, \beta_\tau^2[w]) + \gamma T_m^1(\sigma_\tau, \beta_\tau^1, \beta_\tau^2[w^*]) \cdot \overline{\nu}_{\tau+1}^2[w^*] + (1+\lambda)\eta \tag{3.45}$$
$$+ \gamma\lambda_{\tau+1} \cdot \| T(\sigma_\tau', \beta_\tau^1, \beta_\tau^2[w]) - T_m^1(\sigma_\tau', \beta_\tau^1, \beta_\tau^2[w]) \underbrace{T_c^1(\tilde{\sigma}_\tau^{c,1}, \beta_\tau^2[w])}_{\perp\!\!\!\perp \sigma_\tau'} \|_1 \Big] \tag{3.46}$$

(adding and subtracting the same quantity)

$$\leqslant \max_{\beta_\tau^1} r(\sigma_\tau, \beta_\tau^1, \beta_\tau^2[w]) + \gamma T_m^1(\sigma_\tau, \beta_\tau^1, \beta_\tau^2[w^*]) \cdot \overline{\nu}_{\tau+1}^2[w^*] + (1+\lambda)\eta \tag{3.47}$$

$$+ \gamma\lambda_{\tau+1} \cdot \big[\|T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) - T^1_m(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])\|_1 \tag{3.48}$$

$$-\|T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) - T^1_m(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])\|_1\big] \tag{3.49}$$

$$+ \gamma\lambda_{\tau+1} \cdot \|T(\sigma'_\tau, \beta^1_\tau, \beta^2_\tau[w]) - T^1_m(\sigma'_\tau, \beta^1_\tau, \beta^2_\tau[w]) \underbrace{T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])}_{\perp\!\!\!\perp \sigma'_\tau} \|_1\Big] \tag{3.50}$$

(reordering)

$$= \max_{\beta^1_\tau} r(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) + \gamma T^1_m(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w^*]) \cdot \overline{\nu}^2_{\tau+1}[w^*] \tag{3.51}$$

$$+ \gamma\lambda_{\tau+1} \cdot \big[\|T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) - T^1_m(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])\|_1\big] \tag{3.52}$$

$$+ (1+\lambda)\eta \tag{3.53}$$

$$+ \gamma\lambda_{\tau+1} \cdot \Bigg[\|T(\sigma'_\tau, \beta^1_\tau, \beta^2_\tau[w]) - T^1_m(\sigma'_\tau, \beta^1_\tau, \beta^2_\tau[w]) \underbrace{T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])}_{\perp\!\!\!\perp \sigma'_\tau} \|_1 \tag{3.54}$$

$$- \|T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) - T^1_m(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])\|_1\Big] \tag{3.55}$$

(recognizing $\overline{W}_\tau(\sigma_\tau, \beta^1_\tau)$ and upper-bounding it by $\max_{\tilde{\beta}^1_\tau} \overline{W}_\tau(\sigma_\tau, \tilde{\beta}^1_\tau)$)

$$\leqslant \max_{\beta^1_\tau} \left[\max_{\tilde{\beta}^1_\tau} \overline{W}_\tau(\sigma_\tau, \tilde{\beta}^1_\tau) \right] + (1+\lambda)\eta \tag{3.56}$$

$$\gamma\lambda_{\tau+1} \cdot \Big[\|T(\sigma'_\tau, \beta^1_\tau, \beta^2_\tau[w]) - T^1_m(\sigma'_\tau, \beta^1_\tau, \beta^2_\tau[w]) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])\|_1 \tag{3.57}$$

$$-\|T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) - T^1_m(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])\|_1\Big] \tag{3.58}$$

$$\tag{3.59}$$

But then,

$$|\Big(\|T(\sigma'_\tau, \beta^1_\tau, \beta^2_\tau[w]) - T^1_m(\sigma'_\tau, \beta^1_\tau, \beta^2_\tau[w]) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])\|_1 \tag{3.60}$$

$$-\|T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) - T^1_m(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])\|_1\Big)| \tag{3.61}$$

$$\leqslant \|T(\sigma'_\tau, \beta^1_\tau, \beta^2_\tau[w]) - T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) \tag{3.62}$$

$$- T^1_m(\sigma'_\tau, \beta^1_\tau, \beta^2_\tau[w]) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w]) + T^1_m(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])\|_1 \tag{3.63}$$

$$\leqslant \|T(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) - T(\sigma'_\tau, \beta^1_\tau, \beta^2_\tau[w])\|_1 \tag{3.64}$$

$$+ \|T^1_m(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w]) - T^1_m(\sigma'_\tau, \beta^1_\tau, \beta^2_\tau[w]) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])\|_1 \tag{3.65}$$

But T and $T^{m,1}$ are 1-Lipschitz with respect to σ_τ, meaning that

$$\leqslant \eta + \|T^1_m(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w]) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w]) - T^1_m(\sigma'_\tau, \beta^1_\tau, \beta^2_\tau[w]) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])\|_1 \tag{3.66}$$

$$= \eta + \sum_{\theta^1_{\tau+1}} \sum_{\theta^2_{\tau+1}} |T^1_m(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w])(\theta^1_{\tau+1}) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])(\theta^2_{\tau+1} \mid \theta^1_{\tau+1}) \tag{3.67}$$

$$- T^1_m(\sigma'_\tau, \beta^1_\tau, \beta^2_\tau[w])(\theta^1_{\tau+1}) T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])(\theta^2_{\tau+1} \mid \theta^1_{\tau+1})| \tag{3.68}$$

$$= \eta + \sum_{\theta^1_{\tau+1}} \sum_{\theta^2_{\tau+1}} |\Big[T^1_m(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w])(\theta^1_{\tau+1}) - T^1_m(\sigma'_\tau, \beta^1_\tau, \beta^2_\tau[w])(\theta^1_{\tau+1})\Big] T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])(\theta^2_{\tau+1} \mid \theta^1_{\tau+1})| \tag{3.69}$$

$$= \eta + \sum_{\theta^1_{\tau+1}} \sum_{\theta^2_{\tau+1}} |\Big[T^1_m(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w])(\theta^1_{\tau+1}) - T^1_m(\sigma'_\tau, \beta^1_\tau, \beta^2_\tau[w])(\theta^1_{\tau+1})\Big]| \cdot T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])(\theta^2_{\tau+1} \mid \theta^1_{\tau+1}) \tag{3.70}$$

$$= \eta + \sum_{\theta^1_{\tau+1}} |\Big[T^1_m(\sigma_\tau, \beta^1_\tau, \beta^2_\tau[w])(\theta^1_{\tau+1}) - T^1_m(\sigma'_\tau, \beta^1_\tau, \beta^2_\tau[w])(\theta^1_{\tau+1})\Big]| \underbrace{\sum_{\theta^2_{\tau+1}} T^1_c(\tilde{\sigma}^{c,1}_\tau, \beta^2_\tau[w])(\theta^2_{\tau+1} \mid \theta^1_{\tau+1})}_{=1} \tag{3.71}$$

$$\leqslant 2\eta \tag{3.72}$$

As a consequence, $\max_{\beta^1_\tau} \overline{W}^1_\tau(\sigma'_\tau, \beta^1_\tau) \leqslant \max_{\beta^1_\tau} \overline{W}^1_\tau(\sigma_\tau, \beta^1_\tau) + \eta(3+\lambda)$ □

Theorem 3.1.29 (Finite-time convergence)**.** *HSVI (Algorithm 3.1) terminates in finite time with an ϵ-approximation of $V_0^*(\sigma_0)$ that statisfies Theorem 3.1.23.*

Proof. We will prove by induction from $\tau = H$ to 0, that the algorithm stops expanding OSs at depth τ after finitely many iterations (/trajectories).

First, by definition of horizon H, no OS σ_H is ever expanded. The property thus holds at $\tau = H$.

Let us now assume that the property holds at depth $\tau+1$ after $N_{\tau+1}$ iterations. By contradiction, let us assume that the algorithm generates infinitely many trajectories of length $\tau+1$. As a consequence, because $\mathcal{O}^\sigma_\tau \times \mathcal{B}_\tau$ is compact, after some time the algorithm will have visited $\langle \sigma_\tau, \boldsymbol{\beta}_\tau \rangle$, then, some iterations later, $\langle \sigma'_\tau, \boldsymbol{\beta}'_\tau \rangle$, such that $\|\sigma_\tau - \sigma'_\tau\|_1 \leqslant \rho/(3+\lambda)$. Let us also denote the corresponding terminal OSs (because trajectories beyond iteration $N_{\tau+1}$ do not go further) $\sigma_{\tau+1} = T(\sigma_\tau, \beta_\tau)$ and $\sigma'_{\tau+1} = T(\sigma'_\tau, \boldsymbol{\beta}'_\tau)$. Now, we show that the second trajectory should not have happened, *i.e.,* $\max_{\beta^1_\tau} \overline{W}(\sigma'_\tau, \beta^1_\tau) - \min_{\beta^2_\tau} \underline{W}(\sigma'_\tau, \beta^2_\tau) \leqslant thr(\tau)$.

Combining the previous lemmas,

$$\begin{aligned}
\max_{\beta^1_\tau} \overline{W}_\tau(\sigma'_\tau, \beta^1_\tau) &\leqslant \max_{\beta^1_\tau} \overline{W}_\tau(\sigma_\tau, \beta^1_\tau) + \lambda_\tau \rho && \text{(Lemma 3.1.28)} \\
&= \overline{W}_\tau(\sigma_\tau, \beta^1_\tau) + \lambda_\tau \rho.
\end{aligned}$$

Symmetrically, we also have

$$\min_{\beta^2_\tau} \underline{W}_\tau(\sigma'_\tau, \beta^2_\tau) \geqslant \underline{W}_\tau(\sigma_\tau, \beta^2_\tau) - \lambda_\tau \rho.$$

Hence,

$$\begin{aligned}
\max_{\beta^1_\tau} \overline{W}(\sigma'_\tau, \beta^1_\tau) - \min_{\beta^2_\tau} \underline{W}_\tau(\sigma'_\tau, \beta^2_\tau) &\leqslant \big(\overline{W}_\tau(\sigma_\tau, \beta^1_\tau) + \lambda_\tau\rho\big) - \big(\underline{W}_\tau(\sigma_\tau, \beta^2_\tau) - \lambda_\tau\rho\big) \\
&= \big(\overline{W}_\tau(\sigma_\tau, \beta^1_\tau) - \underline{W}_\tau(\sigma_\tau, \beta^2_\tau)\big) + 2\lambda_\tau\rho \\
&\leqslant \gamma thr(\tau+1) + 2\lambda_\tau\rho && \text{(Lemma 3.1.25)} \\
&= \gamma\left(\gamma^{-(\tau+1)}\epsilon - \sum_{i=1}^{\tau+1} 2\rho\lambda_{\tau+1-i}\gamma^{-i}\right) + 2\lambda_\tau\rho \\
&= \gamma^{-\tau}\epsilon - \sum_{i=1}^{\tau+1} 2\rho\lambda_{\tau+1-i}\gamma^{-i+1} + 2\lambda_\tau\rho \\
&= \gamma^{-\tau}\epsilon - \sum_{j=0}^{\tau} 2\rho\lambda_{\tau-j}\gamma^{-j} + 2\lambda_\tau\rho \\
&= \gamma^{-\tau}\epsilon - \cancel{2\rho\lambda_{\tau-0}\gamma^{-0}} - \sum_{j=1}^{\tau} 2\rho\lambda_{\tau-j}\gamma^{-j} + \cancel{2\lambda_\tau\rho} = thr(\tau).
\end{aligned}$$

Therefore, σ'_τ should not have been expanded. This shows that the algorithm will generate only a finite number of trajectories of length τ.

Finally, note that whenever $\max_{\beta^1_0} \overline{W}_0(\sigma_0, \beta^1_0) - \min_{\beta^2_0} \underline{W}_0(\sigma_0, \beta^2_0) \leqslant thr(0)$, it also holds after an update that $\overline{V}(\sigma_0) - \underline{V}(\sigma_0) \leqslant thr(0)$, which concludes the proof. □

The finite time complexity suffers from the same combinatorial explosion as for cp-POSGs, and is even worse as we have to handle "infinitely branching" trees of possible futures. More precisely, the bound on the number of iterations depends on the number of balls of radius ρ required to cover occupancy simplexes at each depth.

Also, the following proposition allows solving infinite horizon problems as well (when $\gamma < 1$) by bounding the length of HSVI's trajectories using the boundedness of $\overline{V} - \underline{V}$ and the exponential growth of $thr(\tau)$.

Proposition 3.1.30. *Let ρ be a real number satisfying the assumptions of Proposition 3.1.24. When $\gamma < 1$, the length of trajectories is upper bounded by $T_{\max} \overset{\text{def}}{=} \left\lceil \log_\gamma \frac{\epsilon - \frac{2\rho\lambda^\infty}{1-\gamma}}{W - \frac{2\rho\lambda^\infty}{1-\gamma}} \right\rceil$, where λ^∞ is a depth-independent Lipschitz constant and $W \overset{\text{def}}{=} \|\overline{V}^{(0)} - \underline{V}^{(0)}\|_\infty$ is the maximum width between initializations.*

Proof. Since W is the largest possible width, any trajectory stops in the worst case at depth τ such that

$$
\begin{aligned}
thr(\tau) &< W \\
\gamma^{-\tau}\epsilon - 2\rho\lambda^\infty \frac{\gamma^{-\tau} - 1}{1-\gamma} &< W && \text{(from Eq. (3.41))} \\
\gamma^{-\tau}\epsilon - 2\rho\lambda^\infty \frac{\gamma^{-\tau}}{1-\gamma} - 2\rho\lambda^\infty \frac{-1}{1-\gamma} &< W \\
\gamma^{-\tau} \underbrace{\left(\epsilon - \frac{2\rho\lambda^\infty}{1-\gamma}\right)}_{>0} &< W - \frac{2\rho\lambda^\infty}{1-\gamma} \\
\gamma^{-\tau} &< \frac{W - \frac{2\rho\lambda^\infty}{1-\gamma}}{\epsilon - \frac{2\rho\lambda^\infty}{1-\gamma}} \\
\exp(-\tau \ln(\gamma)) &< \exp\left(\ln\left(\frac{W - \frac{2\rho\lambda^\infty}{1-\gamma}}{\epsilon - \frac{2\rho\lambda^\infty}{1-\gamma}}\right)\right) \\
-\tau \ln(\gamma) &< \ln\left(\frac{W - \frac{2\rho\lambda^\infty}{1-\gamma}}{\epsilon - \frac{2\rho\lambda^\infty}{1-\gamma}}\right) \\
\tau \ln(\gamma) &> \ln\left(\frac{\epsilon - \frac{2\rho\lambda^\infty}{1-\gamma}}{W - \frac{2\rho\lambda^\infty}{1-\gamma}}\right) \\
\tau &< \log_\gamma\left(\frac{\epsilon - \frac{2\rho\lambda^\infty}{1-\gamma}}{W - \frac{2\rho\lambda^\infty}{1-\gamma}}\right).
\end{aligned}
$$

□

Before moving on to the experiments, let us first briefly summarize this section's key takeaways. First, we showed the relevance of our notion of "subgames" in `zs-oMG`s as both (an extension of) the minimax theorem and Bellman's optimality principle apply (Theorems 3.1.5 and 3.1.6). Next, we proved continuity properties of transition functions and value functions that allowed designing bounding functions $\overline{W}_\tau$ and $\underline{W}_\tau$ (Proposition 3.1.16). Finally, we showed that, using the latter approximations, selection and backup operators can be written as linear programs (Corollary 3.1.18). Altogether, those ingredients enabled applying of the `HSVI` scheme.

3.2 Experiments

Two main bottlenecks inhibit `HSVI`'s scalability. Firstly, each trajectory generates new tuples w which are added to the bags defining approximators $\overline{W}_\tau$ and $\underline{W}_\tau$. Each trajectory made consequently adds one constraint to each of the `LP`s, which become intractable as the number of trajectories grows. Secondly, contrarily to `cp-oMG`s, the occupancy states are not sparse, because decision rule profiles in `zs-POSG`s are stochastic in general. The dimension of occupancy states quickly grows w.r.t. the horizon and computing them becomes prohibitive, not to mention that it also implies that decision rule profiles live in high-dimensional spaces. Sections 3.4.1 and 3.4.2 discuss complementary levers that could reduce the burden of both bottlenecks.

Experiments presented in this section aim at validating the proposed approach and comparing its behavior to the behavior of some reference algorithms.

3.2.1 Setup

Benchmark Problems

Five benchmark problems were used. *Adversarial Tiger* and *Competitive Tiger* were introduced by Wiggers (2015). *Multi-Agent Broadcast Channel (MABC)* and *Recycling Robots* are well-known 2-player `cp-POSG` benchmark problems (*cf.* `http://masplan.org`) and were adapted to our competitive setting by making player 2 minimize (rather than maximize) the objective function. The fifth benchmark is the adaptation of the well-known Matching Pennies game detailed in Example 2.1.26, with a small difference in that $r(s_h, \cdot, a_h) = +2$ instead of $+1$; this change breaks the symmetry in the optimal strategy, so that `HSVI` can not find the `NES` by "chance" by trying uniform strategies.

We only consider finite horizons H and $\gamma = 1$. Table 3.1 gives the cardinal of the state, action and observation sets for each of these problems.

Table 3.1: Number of states/actions/observations for each benchmark problem

	$\mathcal{S}$	$\mathcal{A}^1$	$\mathcal{A}^2$	$\mathcal{O}^1$	$\mathcal{O}^2$
Competitive Tiger	2	4	4	3	3
Adversarial Tiger	2	3	2	2	2
Recycling Robots	4	3	3	2	2
Mabc	4	2	2	2	2
Matching Pennies	3	2	2	1	1

Algorithms

For conciseness, Algorithm 3.1 is here denoted `HSVI`, and compared against

- `Random` search and `Informed` search (Wiggers 2015) (both using Wiggers's implementation (unlicensed and unreleased)),
- `SFLP` (Koller et al. 1996), and
- `CFR+` (Tammelin 2014)

(the last two using open_spiel (Lanctot et al. 2019) (Apache license)).

All algorithms (but `SFLP`, which is exact) used a target error $\epsilon = 1\%$ of the initial gap $H \cdot (r_{\max} - r_{\min})$. `HSVI` ran with $\lambda_\tau = (H - \tau) \cdot (r_{\max} - r_{\min})$, and ρ the middle of its feasible interval. `HSVI`'s criterion to stop trajectories at a time step $\tau \in \{1, \dots, H-2\}$ (Line 8 of Algorithm 3.1) was changed to stop trajectories whenever $\overline{V}_\tau(\sigma_\tau) - \underline{V}_\tau(\sigma_\tau) \leqslant thr(\tau)$ to reduce the number of `LP`s being solved. We also use `FB-HSVI`'s `LPE` lossless compression of probabilistically equivalent action-observation histories in occupancy states, so as to reduce their dimensionality (Dibangoye et al. 2016). Experiments ran on an Ubuntu machine with i7-10810U 1.10 GHz Intel processor and 16 GB available RAM, and the code is available under MIT license at `https://gitlab.com/aureliendelage1/hsviforzsposgs`.

`Random` and `Informed` only ran once, providing fairly representative results.

3.2.2 Results

Performance Measures A common performance measure in 2-player zero-sum games is the *exploitability* of a strategy $\beta^i_{0:}$, *i.e.,* the difference between the strategy's *security level* (the value of $\neg i$'s best response to $\beta^i_{0:}$) and the Nash equilibrium value $V_0^*(\sigma_0)$:

$$\begin{aligned}\text{exploitability}(\beta^i_{0:}) &= |V^*(\sigma_0) - \sigma_0^{m,1} \cdot \nu^i_{[\sigma_0^{c,\neg i}, \beta^i_{0:}]}| \\ &= |V^*(\sigma_0) - \nu^i_{[\sigma_0^{c,\neg i}, \beta^i_{0:}]}|,\end{aligned}$$

noting that σ_0 is a degenerate distribution over a single element, the pair of empty action-observation histories. In our setting, it will be convenient to look at the (average) *exploitability of a strategy profile* $\langle \beta^1_{0:}, \beta^2_{0:} \rangle$:

$$\begin{aligned}\text{exploitability}(\beta^1_{0:}, \beta^2_{0:}) &= \frac{(V^*(\sigma_0) - \nu^1_{[\sigma_0^{c,2}, \beta^1_{0:}]}) + (\nu^2_{[\sigma_0^{c,1}, \beta^2_{0:}]} - V^*(\sigma_0))}{2} \\ &= \frac{\nu^2_{[\sigma_0^{c,1}, \beta^2_{0:}]} - \nu^1_{[\sigma_0^{c,2}, \beta^1_{0:}]}}{2}.\end{aligned}$$

This quantity is a more concise statistic than both individual exploitabilities, and can be obtained by solving two POMDPs (fixing one player's strategy or the other) without requiring to know the actual NEV, $V^*(\sigma_0)$.

This exploitability can also be defined as half of the *gap between security levels* (SL-gap). To analyze the convergence of algorithms with respect to the initial gap, we will look at the *SL-gap percentage*, *i.e.*,

$$\begin{aligned}\text{SL-gap percentage}(\beta^1_{0:}, \beta^2_{0:}) &= \frac{\nu^2_{[\sigma_0^{c,1}, \beta^2_{0:}]} - \nu^1_{[\sigma_0^{c,2}, \beta^1_{0:}]}}{H \cdot (R_{\max} - R_{\min})} \\ &= \frac{2 \cdot \text{exploitability}(\beta^1_{0:}, \beta^2_{0:})}{H \cdot (R_{\max} - R_{\min})}.\end{aligned}$$

3.2.2.1 Comparison with the state of the art

Table 3.2 gives the convergence time of Wiggers's two heuristic algorithms, CFR, CFR+, SFLP, and HSVI on the benchmark problems with various horizons, or the SL-gap percentage when reaching a 1 h time limit. Executions not returning any result (*i.e.,* for Random, Informed, CFR and CFR+, not performing a single iteration) are noted out-of-time [OOT].

This table first shows that HSVI always outperforms the heuristic baseline provided by Wiggers's algorithms, thus proving the interest of an HSVI scheme. However, HSVI is outperformed by SFLP, CFR and CFR+, unless they run out of time. As can be noted, HSVI is able to keep improving even when the horizon grows thanks to the LPE compression, taking advantage of underlying structure in some games (*e.g.,* Recycling Robots, a problem with transition+observation independence (TOI), when scaling to larger horizons).

We now study the dynamic behavior of the algorithms at hand by providing and analyzing the bounds and exploitability graphs for the same benchmarks.

3.2.2.2 Bounding Graphs

Left-side graphs in Figures 3.2 to 3.7 (pages 70 to 75) show how the computed upper- and lower-bounding values $\overline{V}_0(\sigma_0)$ and $\underline{V}_0(\sigma_0)$ (respectively the *dotted* dark and light green curves) evolve as a function of computation time (always given in seconds). The *solid* dark and light green curves show the security levels $\nu^1_{[\sigma_0, \psi^2_0]}$ and $\nu^2_{[\sigma_0, \psi^1_0]}$ of the current returned strategies ψ^2_0 and ψ^1_0.

Note that, when best-response computations to obtain security levels are expensive (*e.g.,* for the competitive tiger problem, with $H = 4$), they are performed either periodically (*e.g.,* every 10 iterations) or only once, at the end. In the captions, we indicate the (arbitrary) frequency of the POMDP evaluations. For example, $(1, 1, \textit{once})$ means that, for the first two horizons, the POMDP evaluations were done after each iteration, and, for the last one, only once (at the end).

Overall, we observe consistent curves with (i) security levels in-between bounds and around the NEV, and (ii) bounds converging monotonically. Note that HSVI stops when the gap between bounds is small enough, while the gap between SLs (a more relevant criterion used by Informed, Random and CFR, but whose computation can be time-consuming) can be much smaller. As a matter of fact, one can notice that strategies ψ^i_0 returned at each iteration by HSVI are often better (in terms of security level) than their pessimistic lower- or upper-bounding guarantees $\underline{\nu}^1_{[\sigma_0, \psi^1_0]}$ and $\overline{\nu}^2_{[\sigma_0, \psi^2_0]}$.

Table 3.2: Comparison of different solvers on various benchmark problems. Reported values are the running times until the algorithm's error gap (based on bounds for HSVI) is lower than 1 %, or, if the timeout limit is reached, the security-level gap percentages (100 % if gap $= H \cdot (R_{\max} - R_{\min})$). Notes: (1) Horizons with a star exponent (H^*) are those for which the security-level computations ran out of time so that, for HSVI, we give the gap between the pessimistic bounds. (2) Even though Random and Informed contain randomness, we ran them only once, getting fairly representative results.

Domain	H	Wiggers		HSVI	SFLP	CFR	CFR+
		Random	Informed				
Competitive Tiger	2	2.6 %	8.3 %	6 s	**1** s	18 s	2 s
	3	7.0 %	6.1 %	3.8 %	**48** s	57 m	14 m
	4	12.1 %	7.7 %	4.8 %	**14** m	[oot]	[oot]
	5*	[oot]	[oot]	53.3 %	[oot]	[oot]	[oot]
Recycling Robot	2	3.4 %	5.1 %	5 s	**1** s	10 s	**1** s
	3	9.2 %	15.2 %	4 m	**1** s	10 m	1 m
	4	14.1 %	19.6 %	4.9 %	**13** s	2.5 %	24 m
	5	[oot]	[oot]	10.7 %	[oot]	[oot]	[oot]
	6*	[oot]	[oot]	45.5 %	[oot]	[oot]	[oot]
Adversarial Tiger	2	1 s	3.7 %	**1** s	**1** s	**1** s	**1** s
	3	1.5 %	4.4 %	2 m	**1** s	13 s	8 s
	4	2.9 %	5.6 %	2.6 %	**8** s	15 m	4 m
MABC	2	45 s	18.8 %	8 s	3 s	2 m	**1** s
	3	4.2 %	9.2 %	27 s	**1** s	15 m	10 s
	4	18.1 %	36.3 %	4.4 %	**3** s	1.9 %	4 m
Matching Pennies	4	2 m	46.7 %	5 s	**1** s	1 m	**1** s
	5	9 m	45.8 %	1 m	**1** s	7 m	5 s
	6	2.2 %	44.6 %	8 m	**2** s	35 m	17 s

3.2.2.3 Exploitability Graphs

Right-side graphs in Figures 3.2 to 3.7 show the exploitability of the returned strategy profile as a function of computation time for HSVI, Random, Informed, CFR and CFR+ for the different benchmarks considered. A limit precision of 10^{-7} (chosen empirically, according to the LP solver's precision) was applied to HSVI's exploitability.

As can be observed, Random and Informed tend to produce reasonable strategies quickly, but struggle to improve them so as to converge towards an ϵ-NES with $\epsilon \simeq 0$. In contrast, our algorithm keeps improving as computation time increases. The exploitation graphs support the observed behavior in Table 3.2 that HSVI converges in reasonable time compared to Wiggers's algorithms. However, the graphs also show that CFR and CFR+ essentially outperform HSVI when the problems are difficult enough (*i.e.,* when the temporal horizon grows) but the traversal of the whole tree still remains tractable (thus allowing CFR and CFR+ to perform iterations). An interesting observation is that, on small enough problems, HSVI achieves very low exploitabilities earlier than CFR and CFR+.

Finally, HSVI's exploitability graph shares strong similarities with those of Bošanský et al.'s double-oracle algorithms (Bošanský et al. 2014, Fig. 8 and 11). This can be understood as HSVI iteratively building two sets of strategies, one per player, until they are sufficient to support NES profiles, so that the average exploitability is almost zero. But note that Bošanský et al. construct LPs using pure strategies (deterministic best responses), while HSVI's strategies are stochastic.

Having empirically studied the behavior of HSVI compared to other basic offline solvers, we now provide insight about the connections between HSVI and continual (thus online) resolving methods.

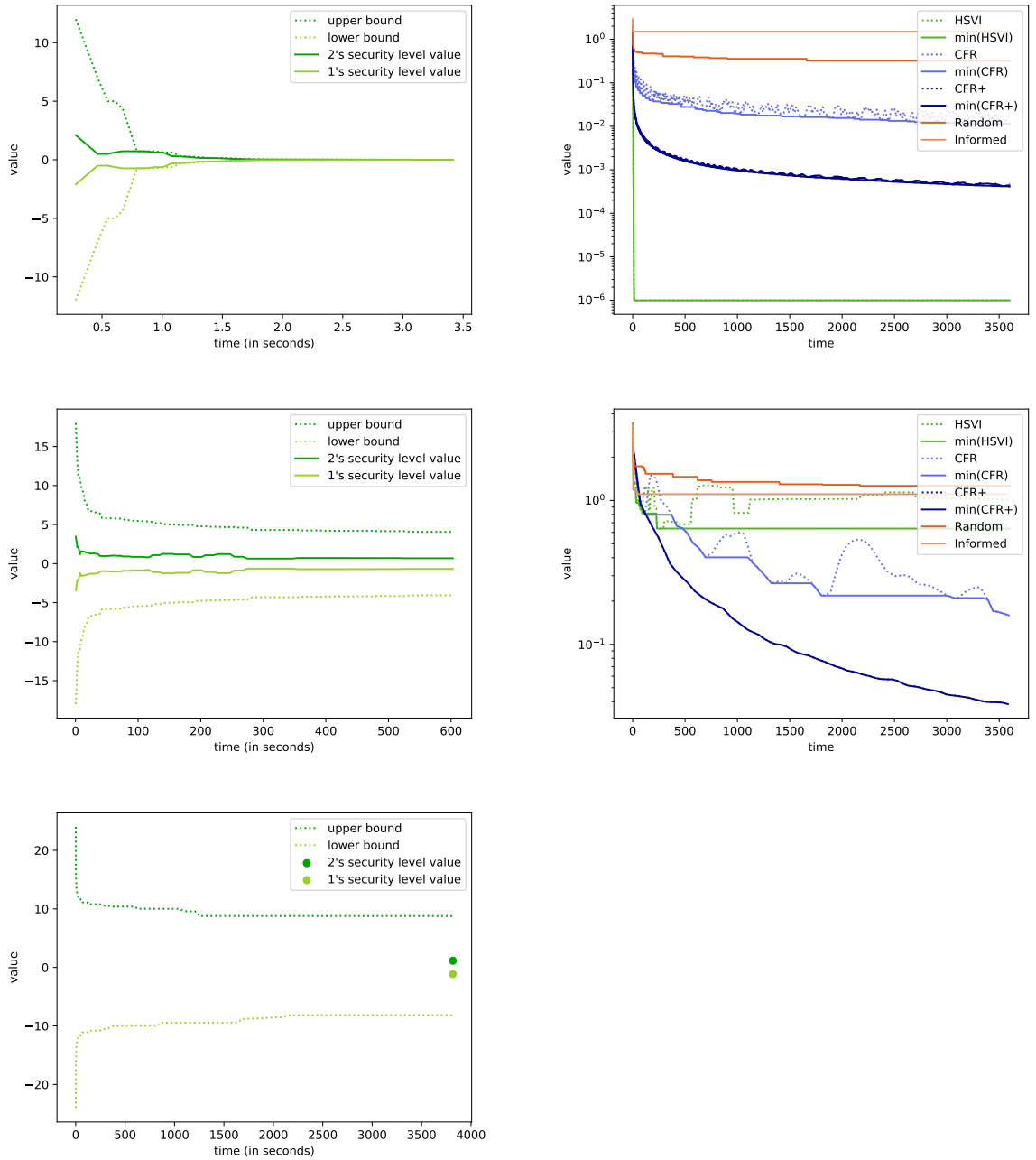


Figure 3.2: **Competitive Tiger ($H = 2, 3$) (1,1): (left)** Evolution of (in dotted lines) the upper- and lower-bound values, and (in solid lines) the security levels of the returned strategies for `HSVI` as a function of time (s). **(right)** Exploitability ($= \frac{\texttt{SL-gap}}{2}$) as a function of time (s) for `Random`, `Informed`, `CFR`, `CFR+`, and `HSVI`.

3.3 Related work

As already pointed out in the introduction and used in our experiments, one can solve a `zs-POSG` by turning it into a zero-sum `EFG` and then using a solving algorithm such as `SFLP` (Koller et al. 1996; Stengel 1996) or `CFR` (Zinkevich et al. 2007). In the following, we present and discuss in more details related work that seem closer to our contribution. We start with Wiggers et al. (2016)'s work. Then, we move on to `HSVI`-like schemes for `zs-OS-POSG`s. Finally, approaches based on continual resolving (Burch et al. 2014) are presented.

3.3.1 Wiggers et al.'s Work on Exploiting the Convex-Concavity of the Optimal Value Function

In this subsection, we come back to Wiggers et al. (2016)'s work, which we built on, to design an `HSVI`-like scheme for solving `zs-POSG`s.

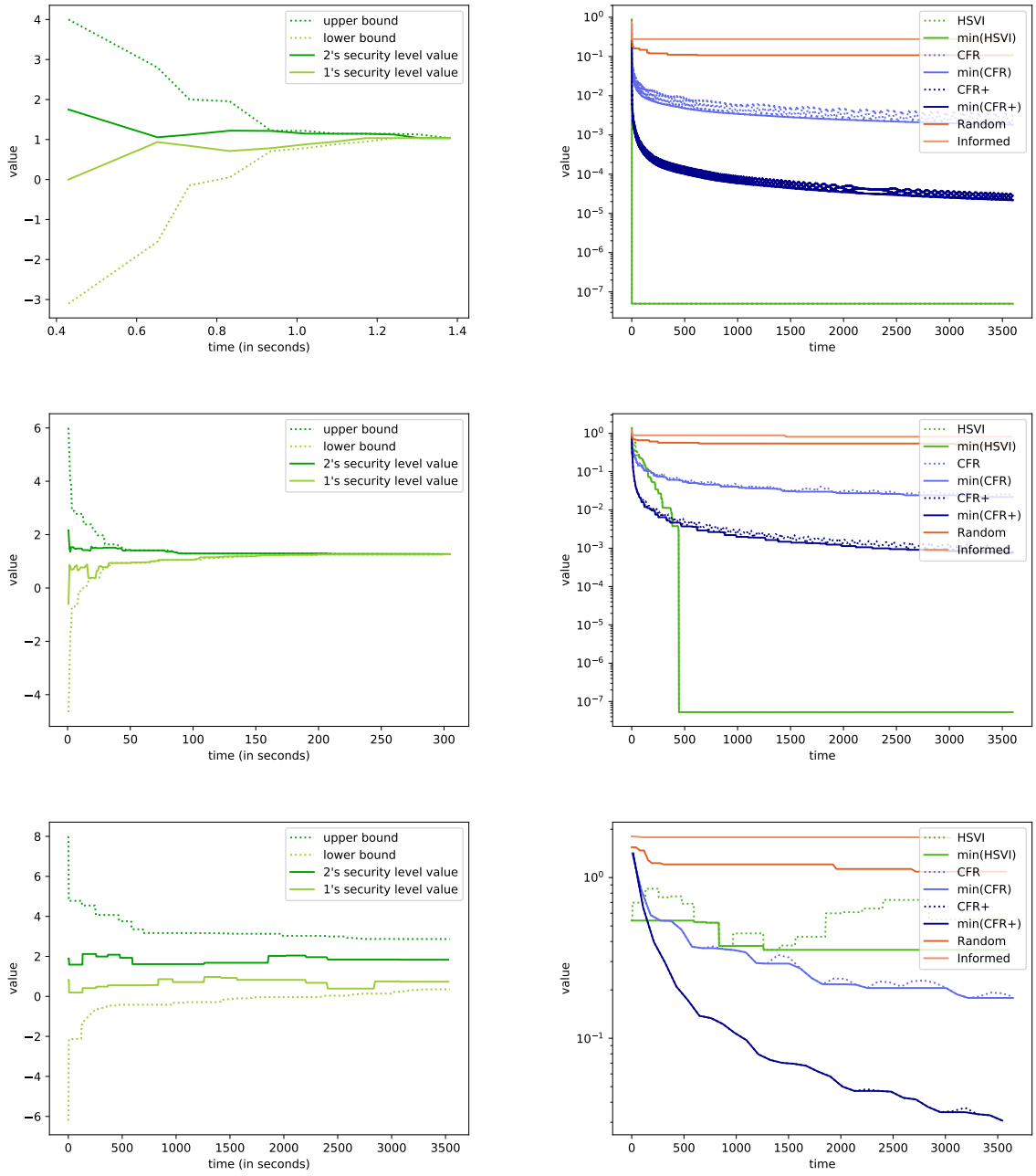


Figure 3.3: **Recycling Robots ($H = 2, 3, 4$) (1,1,10): (left)** Evolution of (in dotted lines) the upper- and lower-bound values, and (in solid lines) the security levels of the returned strategies for `HSVI` as a function of time (s). **(right)** Exploitability ($= \frac{\texttt{SL-gap}}{2}$) as a function of time (s) for `Random`, `Informed`, `CFR`, `CFR+`, and `HSVI`.

3.3.1.1 Deriving zs-SOSGs from zs-POSGs

Our approach for solving `zs-POSG`s by deriving `cp-oMG`s is inspired by Dibangoye et al. (2016)'s introduction of `cp-oMG`s. On the contrary, Wiggers et al. (2016) follow Oliehoek (2013)'s approach that define non-observable Markov decision processes from `cp-oMG`s. Wiggers et al. (2016) thus introduced *plan-time zero-sum non-observable stochastic games (`zs-NOSG`s)* derived from `zs-POSG`s, defined as a tuple $\langle I, \dot{S}, \mathcal{A}^1, \mathcal{A}^2, \dot{\mathcal{O}}, \dot{T}, \dot{O}, \dot{R}, \dot{b}_0 \rangle$, where:

- $I = \{1, 2\}$ is the set of players;
- $\dot{S}$ is the set of augmented states $\dot{s}_t$, each corresponding to a joint AOH θ_t;
- $\mathcal{A}^1$ and $\mathcal{A}^2$ are the decision rules sets as in `zs-oMG`s;

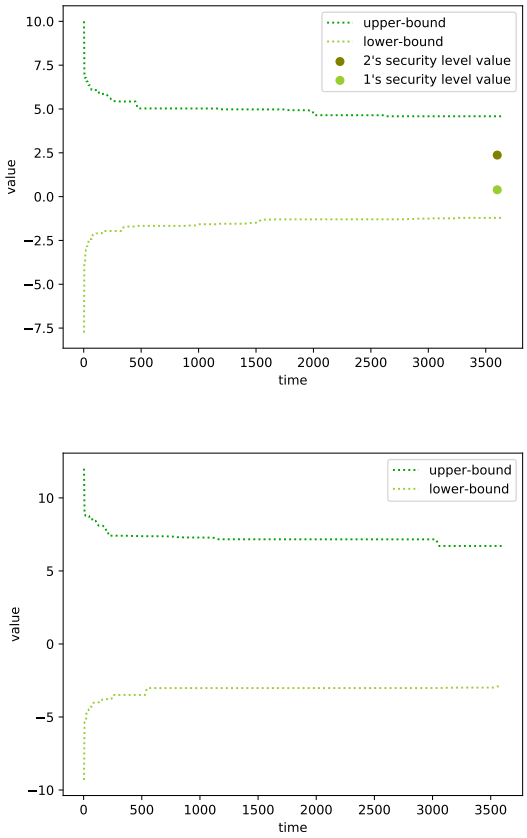


Figure 3.4: **Recycling Robots ($H = 5, 6$) (once,none): (left)** Evolution of (in dotted lines) the upper- and lower-bound values, and (in solid lines) the security levels of the returned strategies for `HSVI` as a function of time (s).

- $\dot{\mathcal{O}}$ is the set of public observations (set to $\{NULL\}$);
- $\dot{T}$ is the transition function between states $\dot{s}_t$ and $\dot{s}_{t+1}$;
- $\dot{O}$ is the observation function that specifies that observation $NULL$ is received with probability 1;
- $\dot{R}$ the reward function : $\dot{R} : \dot{S} \times \mathcal{A}^1 \times \mathcal{A}^2 \to \mathbb{R}$; and
- $\dot{b}_0$ is the initial belief ($\dot{b}_0 \in \Delta(\dot{S})$).

`zs-oMG`s having no observations is equivalent to plan-time `NOSG`s only working with a single trivial $NULL$ observation. The main difference between plan-time `NOSG`s and `zs-oMG`s lies in the state space. The "augmented states" of plan-time `NOSG`s, *i.e.,* tuples of joint histories (s, θ_τ) are very different from occupancy states.[29]

This game is then viewed as a zero-sum *shared observation stochastic game* (`zs-SOSG`) with public actions, Wiggers et al. arguing in their Lemma 4 that, assuming that players are rational, their decision rules can be considered as public. In a thorough and pedagogical discussion regarding public actions, described as the "non-correspondance problem", Sokota et al. show the flaws of such a reasoning (Sokota et al. 2023).[30]

3.3.1.2 Random and Informed (Search)

Building on top of the convex-concave properties, Wiggers introduced Random Search, which alternates between two depth-first searches (one for each player). Each search performs a trajectory through a tree whose vertices correspond to decision rules β^i_τ and nodes to conditional

[29] Note that the same distinction between two possible transformations exists for `cp-POSG`s that are translated either in occupancy `MDP`s or in non-observable `MDP`s (Dibangoye et al., 2016).

[30] In the `cp-POSG` setting, assuming that decision rules are public is valid because agents collaborate.

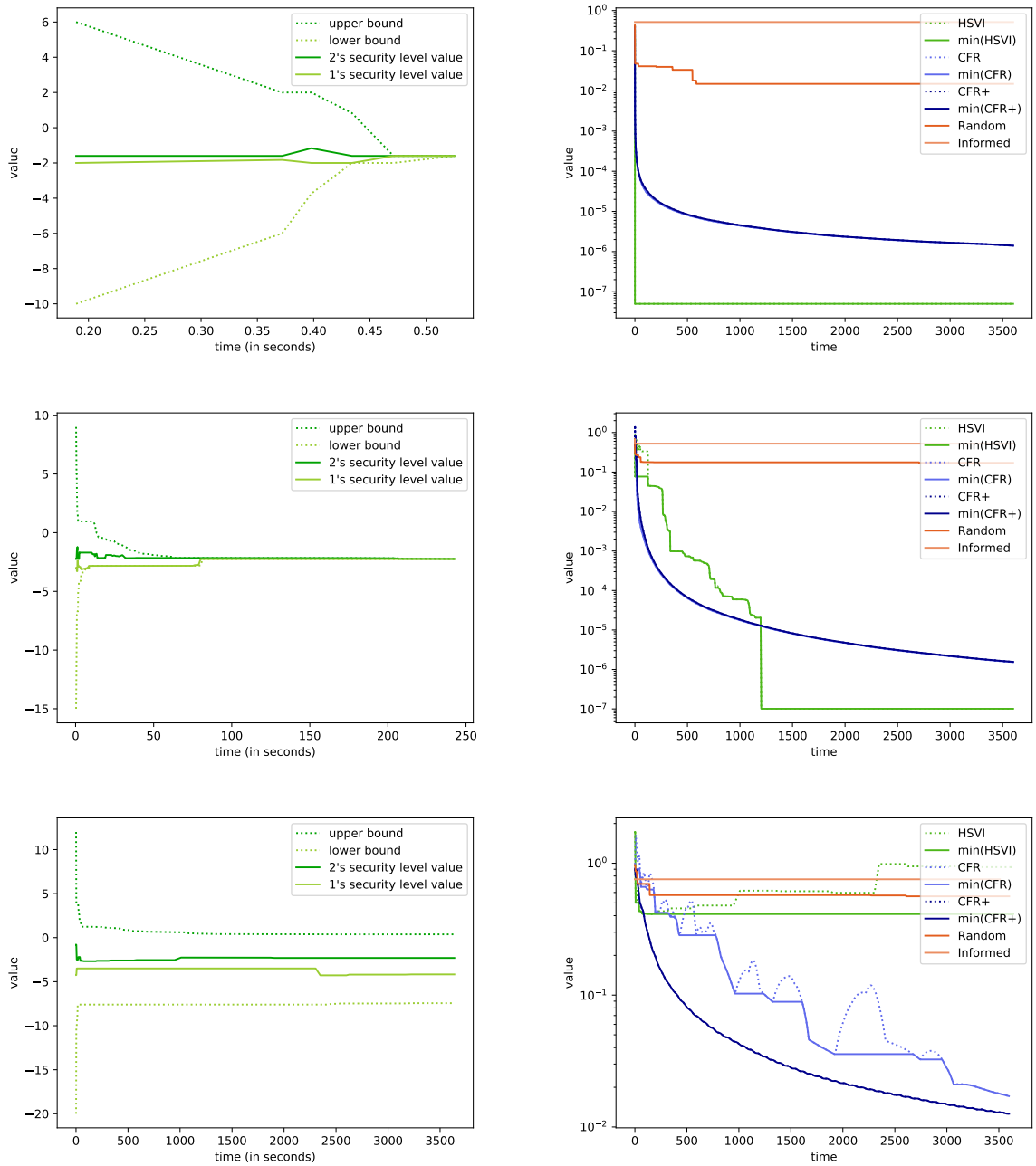


Figure 3.5: **Adversarial Tiger ($H = 2, 3, 4$) (1,1,10): (left)** Evolution of (in dotted lines) the upper- and lower-bound values, and (in solid lines) the security levels of the returned strategies for `HSVI` as a function of time (s). **(right)** Exploitability ($= \frac{\texttt{SL-gap}}{2}$) as a function of time (s) for `Random`, `Informed`, `CFR`, `CFR+`, and `HSVI`.

occupancy states $\sigma_\tau^{c,i}$. The algorithm essentially performs exploration by randomly generating strategies for player $\neg i$ which are evaluated by (costly) computations of security-level vectors $\nu^{\neg i}_{[\sigma_\tau^{c,i}, \beta^{\neg i}_{\tau:}]}$ as solutions of `POMDPs`. Random Search stops whenever its time budget is over or the gap $\nu^2_{[\sigma_0^{c,1}, \beta^2_{0:}]} - \nu^1_{[\sigma_0^{c,2}, \beta^1_{0:}]}$ (measuring the distance to the `NEV`) at the root node is lower than a fixed error ϵ (thus returning the ϵ-`NES`) $\langle \beta^1_{0:}, \beta^2_{0:} \rangle$.

Informed differs from Random in its decision rule generation procedure, by introducing a heuristic to guide trajectories. Instead of using random sampling, the algorithm selects the most promising decision rule for player i against the decision rules stored in $\neg i$'s exploration tree.

In comparison, our `HSVI`-like algorithmic scheme leverages upper- and lower-bounding approximators, from which it generates trajectories induced by decision rule profiles that are optimistic for both players. As a by-product, solution strategies are constructed. Overall, even though high-level similarities exist between `HSVI` and Random (and Informed) Search, major differences

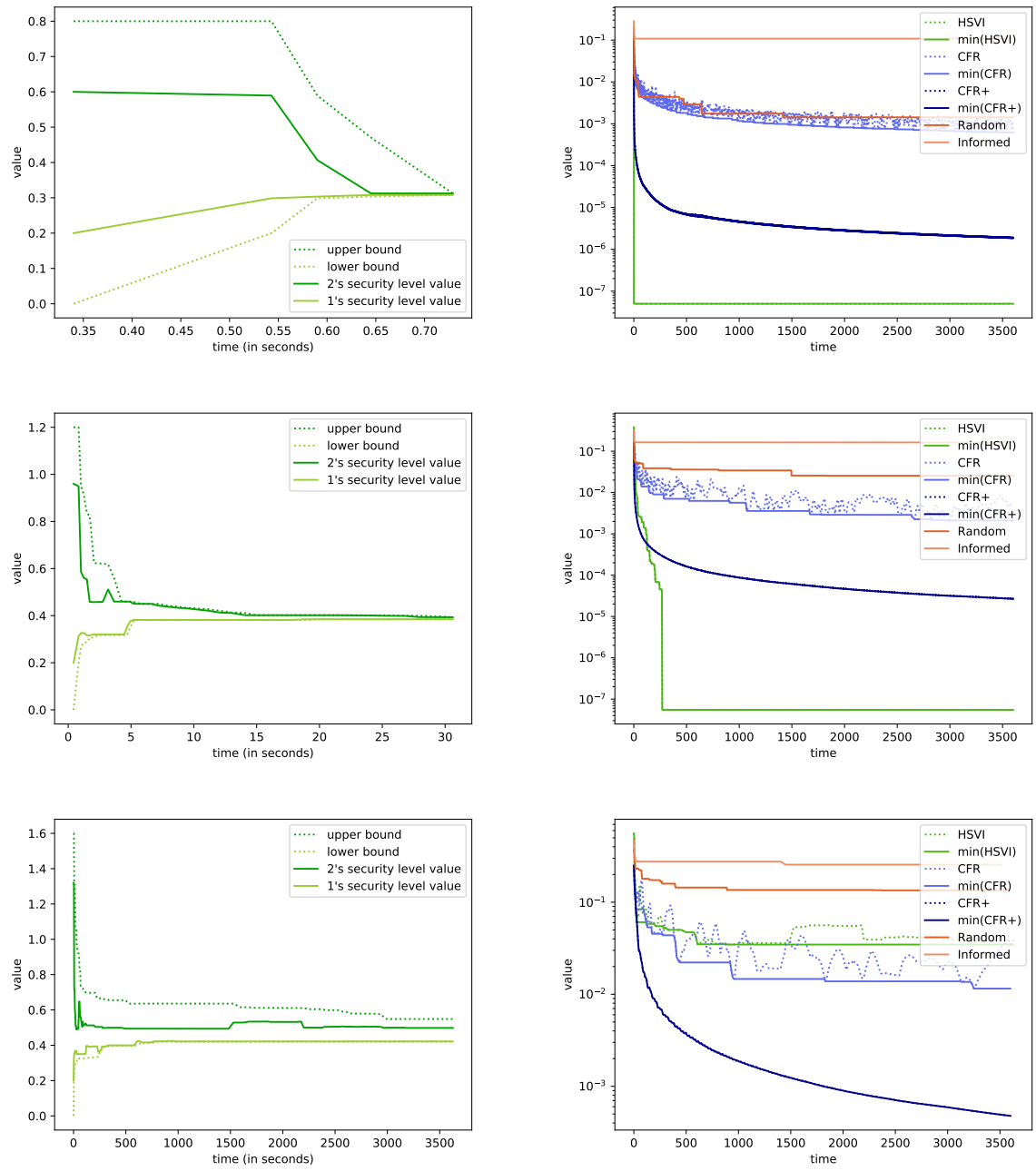


Figure 3.6: **Mabc ($H = 2, 3, 4$) (1,1,10): (left)** Evolution of (in dotted lines) the upper- and lower-bound values, and (in solid lines) the security levels of the returned strategies for `HSVI` as a function of time (s). **(right)** Exploitability ($= \frac{\texttt{SL-gap}}{2}$) as a function of time (s) for `Random`, `Informed`, `CFR`, `CFR+`, and `HSVI`.

distinguish both algorithmic schemes.

3.3.2 Solving zero-sum One-Sided Partially Observable Stochastic Games

For their part, Horák et al. (2017) addressed `zs-OS-POSG`s (a subclass of `zs-POSG`s in which player 2 sees her opponent's actions and observations, as well as the state of the game (Section 2.1.3.6)), succeeding in designing an algorithm that provably converges to an ϵ-`NES` of any `zs-OS-POSG` in finite time. Similarly to us, their work is inspired by techniques developed to solve `POMDP`s (Smith et al., 2005) and later `cp-POSG`s (Dibangoye et al., 2016), relying on a statistic (a belief-state) that induces a Markov game. The resulting value function (V^*) exhibits continuity properties that allow deriving bounding approximations. Then, greedy selection and update operators are defined and included in an `HSVI`-like scheme.

We work in a similar direction for general `zs-POSG`s. As the state space, the strategy space

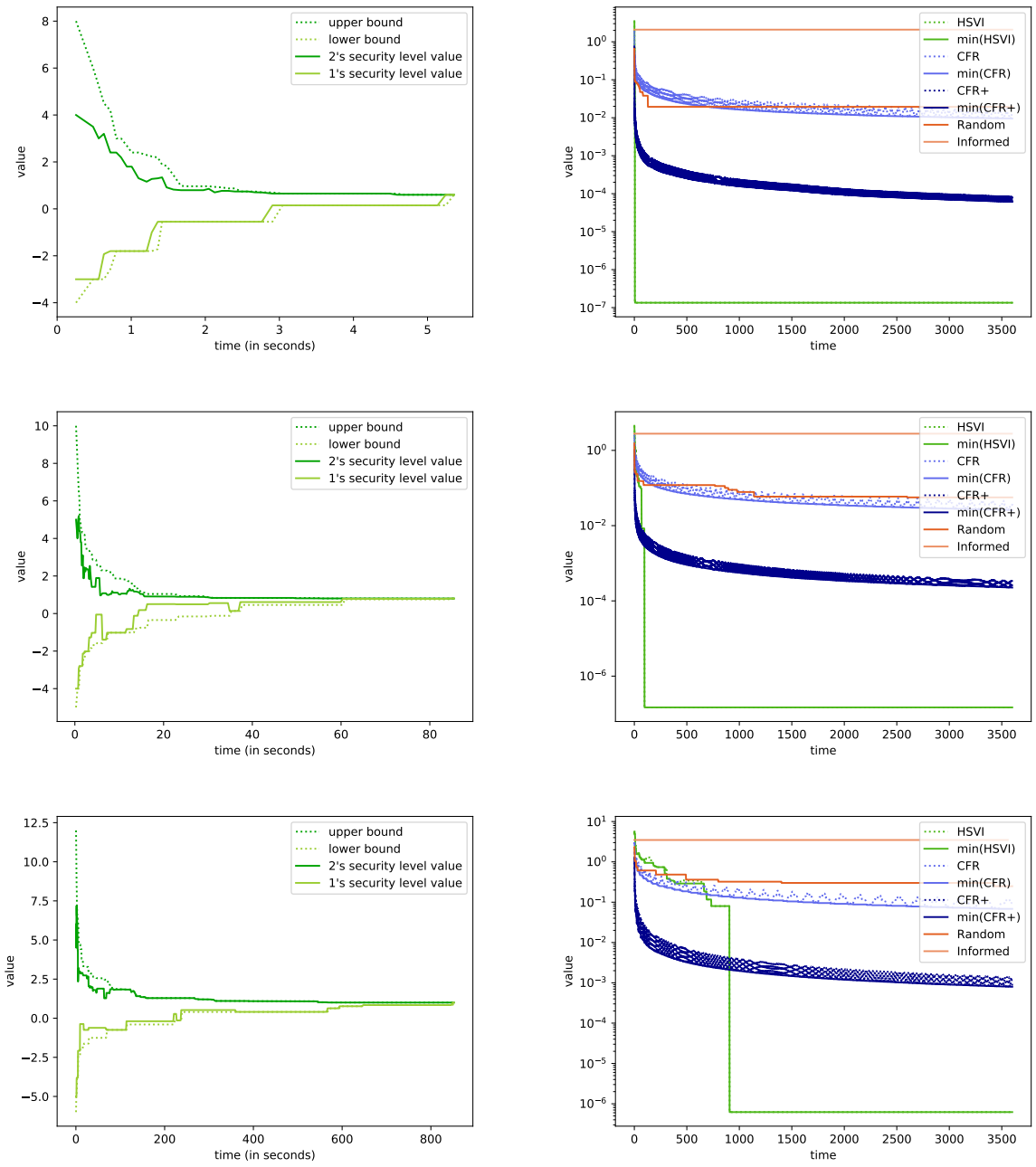


Figure 3.7: **Matching Pennies ($H = 4, 5, 6$) (1,1,1): (left)** Evolution of (in dotted lines) the upper- and lower-bound values, and (in solid lines) the security levels of the returned strategies for `HSVI` as a function of time (s). **(right)** Exploitability ($= \frac{\texttt{SL-gap}}{2}$) as a function of time (s) for `Random`, `Informed`, `CFR`, `CFR+`, and `HSVI`.

and the observability assumptions (*e.g.*, one player having perfect information about the current state of the game, including its opponent's past actions and observations) are different from their counterparts in `zs-OS-POSG`s, building bounding approximators for `zs-POSG`s relies on several technical novelties. A parallel can be made with `cp-POSG`s with one-sided information sharing (Xie et al. 2020), whose observability hypotheses for one agent resulted in significant differences in the `HSVI`-like solving scheme w.r.t. `HSVI` for general `cp-POSG`s.

Furthermore, an important distinction to make with Horák et al.'s work (2023) is about extracting a strategy to execute. The safety issues mentioned in background of this manuscript (Section 2.2.3.5) prevent from trivially computing a Nash equilibrium strategy profile using the knowledge of optimal value functions in `zs-OS-POSG`s. While we showed that guaranteed strategies are intrinsically built (and can then be extracted at the end of the planning phase) when updating bounding approximations in our approach, Horák et al. rely on a continual re-solving method (Moravčík et al., 2017) (see also Section 3.3.3) that is run online after ϵ-convergence of

HSVI bounds

3.3.3 Comparison with Limited-Lookahead Continual Resolving

First of all, let us recall that HSVI, as SFLP and CFR are offline algorithms. On the contrary, continual resolving and limited-lookahead continual resolving are *online* search algorithms, *i.e.,* are meant to make good decisions at each time step, based on the current knowledge about the state of the game.

Limited-lookahead continual resolving (LLCR) schemes, such as *DeepStack* (Moravčík et al. 2017), *Libratus* (Brown et al. 2018), ReBeL (Brown et al. 2020) and *Player of Games* (Schmid et al. 2021), introduce limited-lookahead techniques to the continual resolving scheme (Burch et al. 2014), itself built on top of CFR. LLCR approaches perform well by exploiting a temporal decomposition in subgames, which are specified through knowledge about both players' past strategies. Our approach thus shares similarities with these works.

Yet, as we will see in the next sections, a closer look at continual resolving and LLCR demonstrates how fundamentally different they are.

3.3.3.1 Continual Resolving

(Continual) Resolving techniques have been the first ingredient to adapt CFR to online settings. They assume that, at some time step τ of an actual trajectory, a Nash equilibrium strategy profile is being followed by both players (which induces the subgame at hand), and the objective is to incrementally complete it while preserving global consistency (aka safety) of the whole strategy, *i.e.,* not making choices that could encourage the opponent to deviate *in the past*, before τ. This is achieved by introducing, in a preliminary stage of the subgame, constraints, called *gadgets*, that represent possible deviations and their values, but increase the size of the game tree so that it is practically intractable (Moravčík et al. 2017). For its part, HSVI solves similar subgames (offline), but at prefix strategy profiles that are not necessarily part of a Nash equilibrium, and ensuring only local consistency, *i.e.,* only considering the subgame. HSVI, due to its backpropagation process, ensures the local consistency of each subgame, in particular at $\tau = 0$, which induces a global consistency.

As a comparison with classical (single-agent, deterministic) search, Continual Resolving can be seen as calling Anytime A* (Hansen et al. 2007) during an online resolution process, but with constraints to avoid safety issues, while HSVI can be seen as using LRTA* (Korf 1990), which also works by optimistically generating trajectories until convergence.

Note that neither Resolving nor HSVI require knowing the opponent's actual strategy (which is not public). For Resolving, any opponent prefix strategy of a NES is appropriate to verify that the opponent has no incentive to deviate from a Nash equilibrium strategy. For HSVI, we have shown that reasoning on subgames assuming that the opponent's strategy is known still allows solving a zs-POSG. In practice, both algorithms reason not on actual prefix strategy profiles, but on different statistics. In Resolving, these statistics need to represent *complete* prefix strategy profiles (given the current public information), so that decisions are anticipated for player i even in AOHs (infostates) not reachable given player $\neg i$'s strategy. This leads to using *ranges* (Vojtěch Kovařík et al. 2019), *i.e.,* vectors that give, for each player, her contribution to the probability of any history she could face at time step τ. In contrast, HSVI's occupancy state, which is not necessarily related to a Nash equilibrium strategy in any manner, leads to ignoring unreachable AOHs, which helps to reduce the size of the decision-making (sub)problem.

3.3.3.2 Limited Lookahead

Continual Resolving alone solves complete subgames, thus larger problems at early stages of the game than at the end, which is not appropriate in an online setting. To address this issue through limiting the lookahead of subgames, one needs to estimate the value of the leaves of any truncated subgame. This is achieved through learning offline, for each player i, deep networks that, given the current public belief state, map each AOH to its value under some Nash equilibrium strategy profile. Note that the target function is not unique (Vojtěch Kovařík et al. 2019, Proposition A.1),

since each NES profile maps to different value vectors. Still, according to Vojtěch Kovařík et al. (2019), this does not seem to cause problems in practice.

In contrast, the individual value functions HSVI considers (the "ν" functions) are uniquely defined since they correspond to the best responses to given (not necessarily Nash equilibrium) strategies of the opponent.

3.3.3.3 Limited Lookahead Continual Resolving as a General Scheme?

The previous subsections highlight to what extent HSVI and LLCR are fundamentally different, in particular because they are not on the same algorithmic level. LLCR should be seen as a general scheme in which the subgame solver used, namely CFR, could be replaced by other "basic" offline algorithms such as HSVI or SFLP. But we leave further investigation on this topic for future work.

3.4 Work in Progress

Next, we provide work in progress that might offer new levers to improve the scalability of update and backup operators through (i) pruning unnecessary tuples w that create redondant constraints in our LPs (Corollary 3.1.18, corollary 3.1.18) (Section 3.4.1) and (ii) compressing occupancy states by exhibiting block-diagonal structure (Section 3.4.2).

3.4.1 Pruning $\overline{V}_\tau$

Algorithms for POMDPs and cp-POSGs that rely on approximations of V^* tend to generate an important amount of elements (*e.g.,* vectors, points) that define the approximations. It is often beneficial to prune those that are no longer relevant. This section consequently discusses pruning techniques for cp-oMGs. We start with the following key theorem that allows reusing usual POMDP max-planes pruning techniques in our setting (reverting them to handle min-planes upper bound approximations) to prune approximations of V^*. Next, we present the difficulties encountered when trying to prune approximations of W^*.

Pruning approximations of V^* built using upper envelopes of hyperplanes (*e.g.,* when solving POMDPs) typically involves identifying hyperplanes of the envelope that do not contribute to the approximation, *i.e.,* that are dominated (*e.g.,* by the envelope, or by another hyperplane (Smith 2007)) in any point of the state space.

Theorem 3.4.1 (Proof in Theorem 3.4.1)**.** *Let P be a* min*-planes pruning operator (inverse of* max*-planes pruning for POMDPs), and $\overline{\nu}^2_{[\sigma_\tau^{c,1},\cdot]}$. If P correctly identifies $\overline{\nu}^2_{[\sigma_\tau^{c,1},\cdot]}$ as non-dominated (or resp. dominated) under fixed $\sigma_\tau^{c,1}$, then $\overline{\nu}^2_{[\sigma_\tau^{c,1},\cdot]}$ is non-dominated (or resp. dominated) in $\mathcal{O}^\sigma_\tau$.*

Proof. We will demonstrate that:

- if P shows that a vector ν^2_τ (associated to σ_τ) is dominated *under fixed* $\sigma_\tau^{c,1}$ by a min-planes upper bound relying only on other vectors $\tilde{\nu}^2_\tau$, then this vector is dominated in the whole space $\mathcal{O}^\sigma$;
- else, the vector ν^2_τ is useful at least around $\xi_\tau = (\xi_\tau^{m,1}, \sigma_\tau^{c,1})$, where $\xi_\tau^{m,1}$ is the domination point returned by P.

Note: The following is simply showing that, if the linear part (of the approximator) is dominated by a min-planes approximation for a given conditional term $\sigma_\tau^{c,1}$, then the Lipschitz generalization in the space of conditional terms is also dominated since λ is constant.

Given a matrix $M = (m_{i,j})$, let $\overrightarrow{\|M\|_1}$ denote the column vector whose ith component is $\|m_{i,\cdot}\|_1$. Here, such matrices will correspond to conditional terms, $\overrightarrow{\|\sigma_\tau^{c,1} - \tilde{\sigma}_\tau^{c,1}\|_1}$ denoting the vector whose component for AOH θ^1_τ is $\|\sigma_\tau^{c,1}(\cdot|\theta^1_\tau) - \tilde{\sigma}_\tau^{c,1}(\cdot|\theta^1_\tau)\|_1$ (where $\sigma_\tau^{c,1}(\cdot|\theta^1_\tau)$ may also be denoted $\sigma_\tau^{c,1}(\theta^1_\tau)$ for brevity).

Let us assume that the vector ν_τ^2 (associated to $\sigma_\tau^{c,1}$) is dominated under $\sigma_\tau^{c,1}$, *i.e.*, $\forall \xi_\tau^{m,1}$,

$$(\xi_\tau^{m,1})^\top \cdot (\nu_\tau^2 + \lambda_\tau \overbrace{\|\sigma_\tau^{c,1} - \sigma_\tau^{c,1}\|_1}^{0}) \geqslant \min_{\tilde{\nu}_\tau^2, \tilde{\sigma}_\tau^{c,1}} \left[(\xi_\tau^{m,1})^\top \cdot (\tilde{\nu}_\tau^2 + \lambda_\tau \overrightarrow{\|\sigma_\tau^{c,1} - \tilde{\sigma}_\tau^{c,1}\|_1}) \right].$$

We will show that, $\forall \xi_\tau = (\xi_\tau^{m,1}, \xi_\tau^{c,1})$,

$$(\xi_\tau^{m,1})^\top \cdot (\nu_\tau^2 + \lambda_\tau \overrightarrow{\|\xi_\tau^{c,1} - \sigma_\tau^{c,1}\|_1}) \geqslant \min_{\tilde{\nu}_\tau^2, \tilde{\sigma}_\tau^{c,1}} \left[(\xi_\tau^{m,1})^\top \cdot (\tilde{\nu}_\tau^2 + \lambda_\tau \overrightarrow{\|\xi_\tau^{c,1} - \tilde{\sigma}_\tau^{c,1}\|_1}) \right].$$

Let ξ_τ be an occupancy state. First, remark that $\exists \langle \tilde{\nu}_\tau^2, \tilde{\sigma}_\tau^{c,1} \rangle$ such that

$$(\xi_\tau^{m,1})^\top \cdot (\nu_\tau^2 + \lambda_\tau \overbrace{\|\sigma_\tau^{c,1} - \sigma_\tau^{c,1}\|_1}^{0}) \geqslant (\xi_\tau^{m,1})^\top \cdot (\tilde{\nu}_\tau^2 + \lambda_\tau \overrightarrow{\|\sigma_\tau^{c,1} - \tilde{\sigma}_\tau^{c,1}\|_1}).$$

For the sake of clarity, let us introduce the following functions (where x, y, and z will denote conditional terms for player 1):

$$\begin{aligned} g(x) &\stackrel{\text{def}}{=} \sum_{\theta_\tau^1} \xi_\tau^{m,1}(\theta_\tau^1) \cdot (\nu_y^2(\theta_\tau^1) + \lambda_\tau \|y(\theta_\tau^1) - x(\theta_\tau^1)\|_1) \\ &= g(y) + \lambda_\tau (\xi_\tau^{m,1})^\top \cdot \overrightarrow{\|y - x\|_1}, \end{aligned} \tag{3.73}$$

and

$$\begin{aligned} h(x) &\stackrel{\text{def}}{=} \sum_{\theta_\tau^1} \xi_\tau^{m,1}(\theta_\tau^1) \cdot (\nu_z^2(\theta_\tau^1) + \lambda_\tau \|z(\theta_\tau^1) - x(\theta_\tau^1)\|_1) \\ &= h(z) + \lambda_\tau (\xi_\tau^{m,1})^\top \cdot \overrightarrow{\|z - x\|_1}. \end{aligned}$$

Let us assume that $g(y) \geqslant h(y)$, and show that $g \geqslant h$. First,

$$\begin{aligned} g(x) &= g(y) + \lambda_\tau (\xi_\tau^{m,1})^\top \cdot \overrightarrow{\|x - y\|_1} \\ &\geqslant h(y) + \lambda_\tau (\xi_\tau^{m,1})^\top \cdot \overrightarrow{\|x - y\|_1} \\ &= h(z) + \lambda_\tau (\xi_\tau^{m,1})^\top \cdot (\overrightarrow{\|y - z\|_1} + \overrightarrow{\|x - y\|_1}) \\ &\geqslant h(z) + \lambda_\tau (\xi_\tau^{m,1})^\top \cdot \left(\overrightarrow{\|y - z\|_1} + |(\overrightarrow{\|x - z\|_1} - \overrightarrow{\|z - y\|_1})| \right). \end{aligned} \tag{3.74}$$

Now, $\forall \theta_\tau^1$, if $\|x(\theta_\tau^1) - z(\theta_\tau^1)\|_1 - \|z(\theta_\tau^1) - y(\theta_\tau^1)\|_1 \geqslant 0$, then

$$\begin{aligned} &\|y(\theta_\tau^1) - z(\theta_\tau^1)\|_1 + \left|(\|x(\theta_\tau^1) - z(\theta_\tau^1)\|_1 - \|z(\theta_\tau^1) - y(\theta_\tau^1)\|_1)\right| \\ &= \cancel{\|y(\theta_\tau^1) - z(\theta_\tau^1)\|_1} + \|x(\theta_\tau^1) - z(\theta_\tau^1)\|_1 - \cancel{\|z(\theta_\tau^1) - y(\theta_\tau^1)\|_1} \\ &= \|x(\theta_\tau^1) - z(\theta_\tau^1)\|_1, \end{aligned} \tag{3.75}$$

else,

$$\begin{aligned} &\|y(\theta_\tau^1) - z(\theta_\tau^1)\|_1 + \left|(\|x(\theta_\tau^1) - z(\theta_\tau^1)\|_1 - \|z(\theta_\tau^1) - y(\theta_\tau^1)\|_1)\right| && (3.76) \\ &= 2\|y(\theta_\tau^1) - z(\theta_\tau^1)\|_1 - \|x(\theta_\tau^1) - z(\theta_\tau^1)\|_1 \\ &\geqslant \|x(\theta_\tau^1) - z(\theta_\tau^1)\|_1. && (3.77) \end{aligned}$$

Finally, coming back to (Equation (3.74)):

$$\begin{aligned} g(x) &\geqslant h(z) + \lambda_\tau (\xi_\tau^{m,1})^\top \cdot \left(\overrightarrow{\|y - z\|_1} + |(\overrightarrow{\|x - z\|_1} - \overrightarrow{\|z - y\|_1})| \right) \\ &\geqslant h(z) + \lambda_\tau (\xi_\tau^{m,1})^\top \cdot \|x(\theta_\tau^1) - z(\theta_\tau^1)\|_1 \qquad \text{(from (Equation (3.75)+Equation (3.77)))} \\ &\geqslant h(x). \end{aligned}$$

With $x = \xi_\tau^{c,1}$, $y = \sigma_\tau^{c,1}$ and $z = \tilde{\sigma}_\tau^{c,1}$, this gives:

$$g(\xi_\tau^{c,1}) = \sum_{\theta_\tau^1} \xi_\tau^{m,1}(\theta_\tau^1)(\nu_\tau^2(\theta_\tau^1) + \lambda_\tau \|\sigma_\tau^{c,1}(\theta_\tau^1) - \xi_\tau^{c,1}(\theta_\tau^1)\|_1) \tag{3.78}$$

$$\geqslant h(\xi_\tau^{c,1}) = \sum_{\theta_\tau^1} \xi_\tau^{m,1}(\theta_\tau^1)(\tilde{\nu}_\tau^2(\theta_\tau^1) + \lambda_\tau \|\tilde{\sigma}_\tau^{c,1}(\theta_\tau^1) - \xi_\tau^{c,1}(\theta_\tau^1)\|_1). \tag{3.79}$$

This shows that ν_τ^2 is dominated for every $(\xi_\tau^{m,1}, \xi_\tau^{c,1})$, where both $\xi_\tau^{m,1}$ and $\xi_\tau^{c,1}$ are arbitrary. Therefore, one can prune a vector ν_τ^2 using P applied in the space where $\sigma_\tau^{c,1}$ is fixed.

As a consequence, some properties of P are preserved in its extension to zsPOSGs:

- If P correctly identifies ν_τ^2 as non-dominated at $\sigma_\tau^{c,1}$, then $\langle \nu_\tau^2, \sigma_\tau^{c,1} \rangle$ is non-dominated in $\mathcal{O}_\tau^\sigma$.
  That is, if P does not induce false negatives, neither does its extension to zsPOSGs.
- If P correctly identifies ν_τ^2 as dominated at $\sigma_\tau^{c,1}$, then $\langle \nu_\tau^2, \sigma_\tau^{c,1} \rangle$ is dominated in $\mathcal{O}_\tau^\sigma$.
  That is, if P does not induce false positives, neither does its extension to zsPOSGs.

□

The last theorem permits to prune upper- and lower-bound approximations of V^*. Still, it is not clear how approximators $\overline{W}_\tau$ and $\underline{W}_\tau$ can be pruned as they involve multiplicative terms between the occupancy state σ_τ and decision rules β_τ^1. Therefore, knowing whether tuples w are dominated throughout the whole space $\mathcal{O}_\tau \times \mathcal{B}_\tau^1$ is not an easy task. However, this question could be investigated further by considering relaxed pruning criteria (*i.e.,* that may prune elements that are not dominated everywhere; see for example Smith's PhD thesis for detailed discussion regarding pruning approaches) that can empirically be efficient.

3.4.2 Occupancy-state Decomposition

Note: The following is written for `cp-oMG`*s involving only two players for simplicity (*e.g. *decision rules are deterministic, . . .) but also applies for* `zs-POSG`*s. The extension to more players is left for future work. As discussed later (Section 3.4.2.5), we do not expect major differences with the proposed approach.*

Solving `POSG`s through `oMG`s suffers from the main bottleneck of Bellman's selection and update operators' time and memory complexity. Subclasses characterized by additional observability assumptions were identified as they exhibit structure allowing to drop the previously mentioned complexity (*e.g.,* delayed information-sharing (Nayyar et al. 2010), one-sidedness (Horák et al. 2017; Horák et al. 2019b; Hadfield-Menell et al. 2016; Malik et al. 2018; Xie et al. 2020)). We thus investigate to what extent occupancy states might reveal common knowledge (*i.e.,* something that player 1 knows, player 2 knows that player 1 knows, player 1 knows that player 2 knows that she knows, and so on), and how one can take advantage of it to reduce the complexity of greedily selecting decision rules.

The `POSG` formalism does not explicitly distinguish between private observations that contain public information and those that do not, but adding a public observation variable should be possible. Players' observations would be pairs (z_{pub}, z_{priv}), where z_{pub} is common to all. As a consequence, occupancy states at time step t would naturally be decomposable in blocks, one for each possible sequence of public observations received before t. An analysis of the `POSG`'s dynamics suffices to exhibit such kind of *structural* public knowledge and define public observation variables. In poker for instance, the cards drawn during the flop are given to players as private observations in its classical `POSG` formulation, while being public knowledge. Such situations are of great computational help as they allow branching over public information and drop the complexity of greedy-selection operators. Conversely, a decomposable occupancy state implies that each player knows which block everyone is in, which corresponds to common knowledge.

It is noteworthy that there exist cases for which a preliminary analysis of the games' dynamics would not necessarily reveal some common knowledge. This is partly due to the possibility of common knowledge being generated by specific sequences of players' actions (*i.e.,* such a

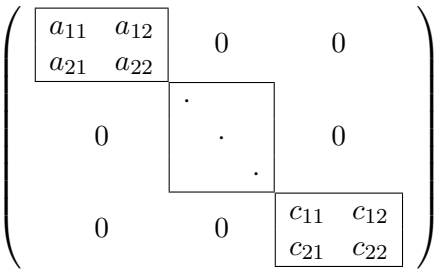


Figure 3.8: An example of a bloc-diagonal matrix

phenomenon would arise along specific trajectories in the game tree). Revealing this *contextual* structure in an occupancy state reduces the complexity of finding the most optimistic decision rule for this occupancy state as decision rules can be computed independently for each common-knowledge block.

Let us consider that, for some reason, an occupancy state σ_t is diagonal, or block diagonal. Assume that player 1 observes individual history $\theta_\tau^{1,*}$ while

- the only individual history for player 2 for which $\sigma_\tau(\theta_\tau^{1,*}, \theta_\tau^2) > 0$ is $\theta_\tau^{2,*}$ and
- the only individual history for player 1 for which $\sigma_\tau(\theta_\tau^1, \theta_\tau^{2,*}) > 0$ is $\theta_\tau^{1,*}$.

It follows that 1 knows that 2 must have observed $\theta_\tau^{2,*}$, player 2 knows that 1 observed $\theta_\tau^{1,*}$ and that 1 knows that she knows, and so on. Still, the latter example only involves one individual history for each player, while the concept of common knowledge is more general.

Definition 3.4.2. *Let σ_t be an occupancy state. A pair of subsets $(\Theta^1, \Theta^2) \subset \Theta^1 \times \Theta^2$ is said to be common knowledge if and only if:*

$$\forall \theta^1 \in \Theta^1, \forall \theta^2 \in \Theta^2, \sigma_t(\boldsymbol{\theta}) > 0 \implies \theta^2 \in \Theta^2, \qquad \textit{and} \tag{3.80}$$

$$\forall \theta^2 \in \Theta^2, \forall \theta^1 \in \Theta^1, \sigma_t(\boldsymbol{\theta}) > 0 \implies \theta^1 \in \Theta^1. \tag{3.81}$$

For example, games with only public observations (and public actions) lead to occupancy states with common knowledge, where sets Θ^i are singletons. Figure 3.8 illustrates an occupancy state for a two-player `cp-BG` such that there exist partitions $(\Theta_1^1, \dots, \Theta_k^1)$ and $(\Theta_1^2, \dots, \Theta_k^2)$ with the property that, for any i in $\{1, \dots, k\}$, (Θ_i^1, Θ_i^2) is common knowledge.

Remark 3.4.3 (Block Decomposition for Bayesian Games)**.** *The current presentation takes the viewpoint of `zs-oMGs` and consequently focuses on occupancy-state decomposition to improve the scalability of planning in `zs-oMGs`. Still, the theory developed here is relevant for Bayesian games, as types correspond to private histories, and probability distributions p to occupancy states.*

In the following, we start by formally describing block-diagonal occupancy states. Then, Section 3.4.2.2 provides an algorithm to decompose, if possible, an occupancy state and its time complexity is studied. Section 3.4.2.3 tackles the interesting question of approximating an occupancy state with one that can be decomposed, while guaranteeing that the loss in Bellman's equation solutions is no more than a given ϵ. Finally, Section 3.4.2.4 investigates the impact of block-diagonal occupancy states on Bellman's optimality equations.

3.4.2.1 Block Decomposition of Occupancy States

We start with a lemma proving the sufficiency of a collection of independent statistics to summarize a block-diagonal occupancy state.

Lemma 3.4.4 (Reduction of Occupancy States)**.** *Let σ_t be an occupancy state. Then, without loss of information, σ_t can be represented by a matrix $M = (\mathbb{P}(\theta_t^1, \theta_t^2 \mid \sigma_\tau))_{\theta_t^1, \theta_t^2 \in \Theta_t^1 \times \Theta_t^2}$. If this matrix is block-diagonal,* i.e., *$M = Diag(B_1, \dots, B_q)$ with $q \leqslant \min\{|\Theta_t^1|, |\Theta_t^2|\}$, then σ_t carries the same information as a mixture of occupancy states with lower dimensionalities.*

Proof. Let us define the function $B_k : \Theta^1_\tau \times \Theta^2_\tau \to [0,1]$ such that $B_k(\theta^1_\tau, \theta^2_\tau) = \mathbb{P}(\theta^1_\tau, \theta^2_\tau \mid \sigma_\tau)$ if (θ^1_t, θ^2_t) belongs to block B_k in M, and 0 otherwise. Besides, we denote, $\forall k$, $\texttt{supp}(B_k)$ the set of pairs (θ^1_t, θ^2_t) belonging to the block B_k within σ_t.

Reusing the notations of the lemma,$\forall (\theta^1_t, \theta^2_t) \in \Theta^1_t \times \Theta^2_t$,

$$\mathbb{P}(\theta^1_t, \theta^2_t \mid \sigma_t) = \sum_{k=1}^{q} B_k(\theta^1_t, \theta^2_t), \tag{3.82}$$

$$= \sum_{k=1}^{q} \left[\sum_{(\tilde{\theta}^1_t, \tilde{\theta}^2_t) \in \texttt{supp}(\eta_q)} B_k(\tilde{\theta}^1_t, \tilde{\theta}^2_t) \right] \frac{B_k(\theta^1_t, \theta^2_t)}{\left[\sum_{(\tilde{\theta}^1_t, \tilde{\theta}^2_t) \in \texttt{supp}(\eta_q)} B_k(\tilde{\theta}^1_t, \tilde{\theta}^2_t) \right]}, \tag{3.83}$$

noting $\eta_k \overset{\text{def}}{=} \sum_{(\tilde{\theta}^1_t, \tilde{\theta}^2_t) \in \texttt{supp}(B_k)} B_k(\tilde{\theta}^1_t, \tilde{\theta}^2_t)$,

$$= \sum_{k=1}^{q} \eta_k \cdot \frac{B_k(\theta^1_t, \theta^2_t)}{\eta_k} \tag{3.84}$$

$$\overset{\text{def}}{=} \sum_{k=1}^{q} \eta_k \cdot \sigma_{t,k}(\theta^1_t, \theta^2_t). \tag{3.85}$$

Therefore, $(\eta_k, \sigma_{t,k})_k$, which describes a distribution over q occupancy states, carries the same information as σ_t. □

Lemma 3.4.4 would also hold if we "decompose" the matrix row by row (or column by column), retrieving usual marginal and conditional terms. What is of particular interest with block-diagonal matrices is that each decision-rule profile can be decomposed in q independent "partial decision-rule profiles". Besides, the dimensionality of spaces of interest (occupancy states $\sigma_{t,k}$ and partial decision rules) are highly reduced.

Lemma 3.4.5. *Let σ_t be an occupancy state at time step t. For any k in $\{1, \dots, q\}$, let us define $\beta_{t,k} : (\theta^1_t, \theta^2_t) \in \texttt{supp}(B_k) \mapsto \beta_{t,k}(\theta^1_t, \theta^2_t)$ a "partial decision rule" (*i.e., *prescribing actions to players only for the histories in block B_k). Then, $\sigma_{t+1} = T(\sigma_t, \langle \beta_{t,1}, \dots, \beta_{t,q} \rangle)$ carries the same information as $(\eta_{t,k}, T^k(\sigma_{t,k}, \beta_{t,k}))_k$, where T^k is a transition function.*

Proof. The proof below uses the fact that decision rules in a `cp-POSG` can be considered deterministic (Oliehoek et al., 2008). The extension of the proof to stochastic ones would follow similar derivations.

Let $\tilde{\theta}^1_\tau$ and $\tilde{\theta}^2_\tau$ be histories for player 1 and 2 at time step t. For any $\tilde{\theta}^1_{t+1}, \tilde{\theta}^2_{t+1}$ (noting $\tilde{\boldsymbol{\theta}}_{t+1} = \tilde{\boldsymbol{\theta}}_t \oplus \{\tilde{\boldsymbol{a}}_t, \tilde{\boldsymbol{z}}_{t+1}\}$), it holds that

$$\mathbb{P}(\tilde{\boldsymbol{\theta}}_{t+1} \mid \sigma_{t+1}) = \sum_{\boldsymbol{\theta}_t} \mathbb{1}_{\{\boldsymbol{\theta}_t = \tilde{\boldsymbol{\theta}}_t\}} \cdot \sigma_t(\boldsymbol{\theta}_t) \cdot \boldsymbol{\beta}_t(\boldsymbol{\theta}_t, \tilde{\boldsymbol{a}}_t) \cdot \mathbb{P}(\tilde{\boldsymbol{z}}_{t+1} \mid \boldsymbol{\theta}_t, \tilde{\boldsymbol{a}}_t) \tag{3.86}$$

$$= \sum_{k} \sum_{\boldsymbol{\theta}_t \in \texttt{supp}(B_k)} \mathbb{1}_{\{\boldsymbol{\theta}_t = \tilde{\boldsymbol{\theta}}_t\}} \cdot \eta_k \cdot \sigma_{t,k}(\boldsymbol{\theta}_t) \cdot \boldsymbol{\beta}_{t,k}(\boldsymbol{\theta}_t, \tilde{\boldsymbol{a}}_t) \cdot \mathbb{P}(\tilde{\boldsymbol{z}}_{t+1} \mid \boldsymbol{\theta}_t, \tilde{\boldsymbol{a}}_t) \tag{3.87}$$

$$\overset{\text{def}}{=} \sum_{k} \sum_{\boldsymbol{\theta}_t \in \texttt{supp}(B_k)} \mathbb{1}_{\{\boldsymbol{\theta}_t = \tilde{\boldsymbol{\theta}}_t\}} \cdot \eta_{t,k} \cdot T^k(\sigma_{t,k}, \beta_{t,k})(\boldsymbol{\theta}_{t+1}). \tag{3.88}$$

□

Note that, if an occupancy state σ_t is block-diagonal, then the occupancy state $\sigma_{\tau+1} = T(\sigma_\tau, \boldsymbol{\beta}_\tau)$ reached for any decision rule profile $\boldsymbol{\beta}_\tau$ has, at least, as many blocks as σ_t.

3.4.2.2 Finding Blocks

We here study how to find blocks and the computational cost of doing so. This complexity must be less than exponential to hope for an overall gain for the optimistic decision rule selection.

Let σ_t be an occupancy state at time step $t \in \{0, \dots, H-1\}$ and $G_{\sigma_t} = \langle \Theta^1_t \cup \Theta^2_t, E \rangle$ be an undirected graph, where a tuple (θ^1_t, θ^2_t) belongs to the set E of edges if and only if $\sigma_t(\theta^1_t, \theta^2_t) > 0$.

Note that G_{σ_t} is bi-partite as nodes in Θ_t^1 are only connected to nodes in Θ_t^2. Then, finding a block decomposition of an occupancy state σ_t is equivalent to finding all connected subgraphs in the corresponding graph G_{σ_t}.

Algorithm 3.2 (Hopcroft et al., 1973) finds connected subgraphs in a graph G_{σ_t} (for example, by performing a best-first search).

Algorithm 3.2: Find Connected Subgraphs (Hopcroft et al. 1973)

Data: Graph $G_{\sigma_t} = \langle \Theta_t^1 \cup \Theta_t^2, E \rangle$
Result: List of unconnected subgraphs

```
visited ← empty set
subgraphs ← empty list
foreach θ_t ∈ Θ_t^1 ∪ Θ_t^2 do
    if θ_t is not in visited then
        subgraph ← empty set
        DFS(θ_t, subgraph)
        subgraphs.append(subgraph)
return subgraphs
Function DFS(θ_t, subgraph):
    visited.add(θ_t)
    subgraph.add(θ_t)
    foreach θ'_t such that (θ_t, θ'_t) is an edge in G do
        if θ'_t is not in visited then
            DFS(θ'_t, subgraph)
```

The time complexity of Algorithm 3.2 is $\mathcal{O}(|V| + |E|)$, where $|V| = |\Theta_t^1| + |\Theta_t^2|$, and $|E|$ is, at most, $|\Theta_t^1| \cdot |\Theta_t^2|$. In a `cp-POSG`, $|\Theta_t^1| = (|\mathcal{A}^1| \cdot |\mathcal{Z}^1|)^t$, and $\Theta_t^2 = (|\mathcal{A}^2| \cdot |\mathcal{Z}^2|)^t$.

3.4.2.3 ϵ-Close Block Decomposition

Whenever exact block decomposition is impossible, or if it creates a block almost as large as the occupancy state itself, it may be interesting to search for a lossy approximation of the occupancy state by a decomposable one. Tools from graph theory might help us to do so.

Definition 3.4.6 (min k-cut problem)**.** *Given an undirected graph $G = (V, E)$ and a weight function $w : E \to \mathbb{N}$, the minimal k-cut problem is to find a partition $F = (C_1, \dots, C_k)$ of V into disjoint sets $(C_i)_{i \in \{1,\dots,k\}}$ such that the loss $\sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \sum_{v_1 \in C_i v_2 \in C_j} w(e_{v_1 \to v_2})$ is minimal, where $e_{v_1 \to v_2}$ is the edge between vertices v_1 and v_2.*

Gomory-Hu trees (Gomory et al. 1961) represent minimal cuts in G for all pairs of vertices. A k-cut whose value is at most $(2 - 2/k)$ times the minimum value of the min k-cut problem, can be found by iteratively selecting pairs with minimal value in the representation. This procedure should be applicable to our graphs $G = \langle \Theta^1 \cup \Theta^2, p \rangle$. Even though Gomory-Hu trees allow computing a block decomposition that can be at most twice worse than an optimal one, its computation is easier than other methods and runs in polynomial time for a fixed k. The following example illustrates the procedure of ϵ-block decomposition using Gomory Hu trees, for $k = 2$.

Example 3.4.7. *Let us consider an occupancy state σ_t:*

$$\begin{pmatrix} a & 0 & b & 0 \\ 0 & \alpha & \epsilon_1 & \beta \\ c & \epsilon_2 & d & 0 \\ 0 & \gamma & 0 & \delta \end{pmatrix},$$

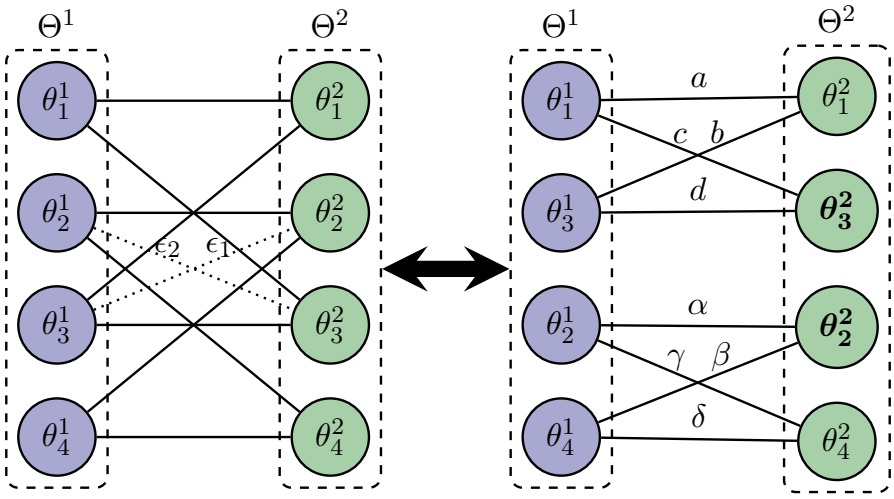


Figure 3.9: Illustration of an occupancy state as a graph. Probabilities on the right are not normalized.

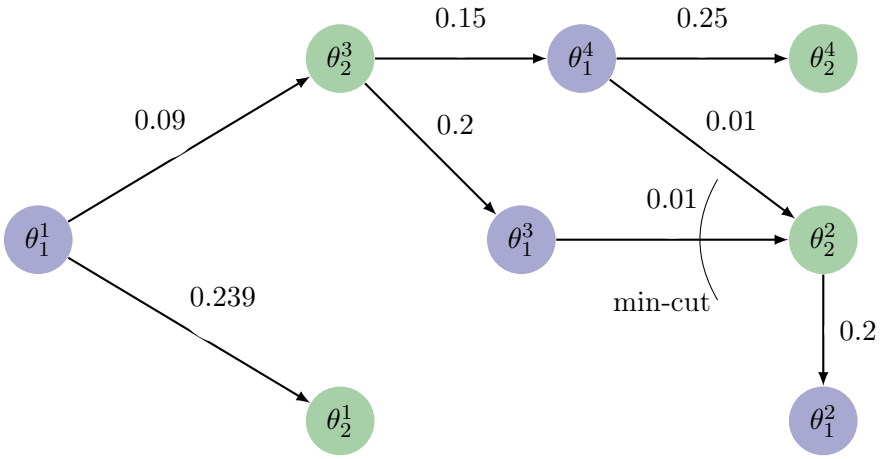


Figure 3.10: Example of Gomory-Hu Tree.

where ϵ_1 *and* ϵ_2 *are small positive numbers. Figure 3.9 (left) illustrates the graph* $G = \langle \Theta^1 \cup \Theta^2, E \rangle$ *derived from the occupancy state.*

Discarding edges with weight ϵ_1 *and* ϵ_2*, normalizing, and re-organizing the order of individual histories (see Figure 3.9 (right)),* σ_t *can be re-written:*

$$\frac{1}{1-(\epsilon_1+\epsilon_2)} \begin{pmatrix} \boxed{\begin{matrix} a & b \\ c & d \end{matrix}} & 0 \\ 0 & \boxed{\begin{matrix} \alpha & \beta \\ \gamma & \delta \end{matrix}} \end{pmatrix}$$

Figure 3.10 shows the Gomory-Hu tree for the following occupancy state (where constants $a, b, c, d, \alpha, \beta, \gamma, \delta$ *are specified):*

$$M = \begin{pmatrix} 0.2 & 0 & 0.05 & 0 \\ 0 & 0.149 & 0.001 & 0.05 \\ 0.039 & 0.001 & 0.159 & 0 \\ 0 & 0.1 & 0 & 0.2 \end{pmatrix}.$$

Performing a $\min k-cut$ *on the pair* (θ_1^3, θ_2^2) *of vertices yields a block diagonal decomposition* $\tilde{M}$ *such that* $\|M - \tilde{M}\|_1 \leqslant 0.02$.

Dibangoye et al. (2014) proposed elegant tools to bound the error made when computing Bellman's equations with respect to an approximation of occupancy states, as a function of the total variation between the approximation and the original distribution. We now investigate the

impact of block decomposition in Bellman's optimality equation and the time (and memory) complexity of greedy selection operators.

3.4.2.4 Bellman's equation

As a reminder, Bellman's equation for a `cp-oMG` is

$$V_t^*(\sigma_t) = \max_{\beta_t} r(\sigma_t, \beta_t) + \gamma V_{t+1}^*(T(\sigma_t, \beta_t)), \tag{3.89}$$

which can be rewritten (Dibangoye et al. 2016) using α-vectors $\alpha \in \mathbb{R}^{|\Theta_t|}$ as:

$$= \max_{\alpha} \langle \sigma_t, \alpha \rangle. \tag{3.90}$$

We recall that, in Bellman's classical equation, every decision rule profile is enumerated, whereas decomposition allows enumerating decision rules independently for each block.

Theorem 3.4.8. *Let σ_t be a block-diagonal occupancy state whose blocks are noted $(B_1, \dots, B_q)$. For any player i, let $\mathit{supp}^i(B_k)$ be the set of individual histories of player i within block B_k. The complexity of computing Bellman's equation is reduced from $\Pi_i |\mathcal{A}^i|^{|\Theta_t^i|}$ to $\sum_k \prod_i |\mathcal{A}^i|^{|\mathit{supp}^i(B_k)|}$.*

Proof. Using Lemma 3.4.5,

$$V_t^*(\sigma_t) = \max_{\beta_t} \left[\sum_k \eta_k \mathbb{E}[R(S_t, A_t) \mid \sigma_{t,k}, \beta_{t,k}] + \gamma \max_{\beta_{t+1:}} V_{t+1}((T^k(\sigma_{t,k}, \beta_{t,k}))_{k=1}^q, \beta_{t+1:}) \right]. \tag{3.91}$$

Note that $\sigma_{t+1,k}$ is an element of $\mathcal{O}_{t+1}$ as it is normalized (similarly to $\sigma_{t,k}$), giving sense to $V_{t+1}(T^k(\sigma_{t,k}, \beta_t), \beta_{t+1:,k})$. Besides, $\beta_{t+1:}$ can be decomposed in $\langle \beta_{t+1:,1}, \dots, \beta_{t+1:,q} \rangle$ as each history at $t+1$ extends an history belonging to only one block of σ_t. Then,

$$V_t^*(\sigma_t) = \max_{\beta_t} \sum_k \eta_k \left[r(\sigma_{t,k}, \beta_t) + \gamma \max_{\beta_{t+1:,k}} V_{t+1}(T^k(\sigma_{t,k}, \beta_t), \beta_{t+1:,k}) \right] \tag{3.92}$$

$$= \max_{\langle \beta_{t,1}, \dots, \beta_{t,q} \rangle} \{ \sum_k \eta_k \left[r(\sigma_{t,k}, \beta_{t,k}) + \gamma V_{t+1}^*(T^k(\sigma_{t,k}, \beta_{t,k})) \right] \} \tag{3.93}$$

$$= \sum_k \eta_k \left[\max_{\beta_{t,k}} \{ r(\sigma_t, k, \beta_{t,k}) + \gamma V_{t+1}^*(T^k(\sigma_{t,k}, \beta_t^k)) \right] \tag{3.94}$$

$$= \langle (\eta_k \sigma_{t,k})_k, (\alpha_k^*)_k \rangle, \tag{3.95}$$

where $\forall k,\ \alpha_k : \mathtt{supp}(B_k) \to \mathbb{R}$ and $\alpha_k^* \in \arg\max_{\alpha_k} \langle \eta_k \sigma_{t,k}, \alpha_k \rangle$. □

Note that, if the matrix is diagonal, the greedy selection's complexity becomes exponential instead of double exponential.

Theorem 3.4.8 also incites algorithms such as `HSVI` to branch over possible blocks. Similarly to `HSVI` for `POMDP`s, only the next occupancy state $T(\sigma_{t,k}, \boldsymbol{\beta}_{t,k})$ contributing the most to the uncertainty on the value of σ_t would be studied next by an `HSVI` for `cp-POSG`s leveraging block decomposition.

3.4.2.5 Block Decomposition for More Than Two Players

In the case where more than two players are involved in the game, our approach for finding block decomposition (ϵ-closely or not) does not apply directly. Indeed, occupancy states are no longer matrices but tensors and the derived graphs would typically be hypergraphs, whose hyperedges gather tuples $(\theta_\tau^1, \dots, \theta_\tau^n)$ for which $\sigma(\theta_\tau^1, \dots, \theta_\tau^n) > 0$. Still, there exist algorithmic approaches to tackle min k-cut problems for hypergraphs (Fox et al. 2023), which shall satisfy our needs.

3.4.2.6 Conclusion

This section presented an algorithmic tool to exhibit block-diagonal structure within occupancy states. The main purpose is to reduce the prohibitive time complexity of (i) Bellman's equation in cp-OMGs and (ii) occupancy states computations in large games. We showed that any occupancy state carries the same information as a mixture over smaller occupancy states and proved that using the latter ones reduces the complexity of Bellman's equations. Besides, block decomposition of occupancy states allows branching over blocks and only studying most relevant ones in trajectories of HSVI-like schemes, which also decreases the difficulty of maintaining estimates of occupancy states. Inspired by tools for approximately compressing occupancy states (Dibangoye et al. 2014a), we also discussed the topic of of ϵ-block decompositions.

Multiple interesting questions are raised by our findings. Finding efficient tools for ϵ-close approximation is certainly one of them, but studying the case with more than two players as well. Relative homogeneity within exhibited blocks would also be appreciated, as the complexity of Bellman's operator depends on the size of the *largest* block(s). Finally, experiments on classical benchmarks or real-life applications shall be made to validate the empirical gain of the approach on some problems.

3.5 Discussion

This chapter addressed the problem of ϵ-optimally solving zs-POSGs. In contrast to SFLP or CFR, we provide the necessary foundational building blocks to apply dynamic programming (in tandem with heuristic search) to solve zs-POSGs. We introduce Bellman optimality equations and uniform-continuity properties of the optimal value function. Next, we exhibit rules for updating value functions while preserving uniform continuity and the ability to extract globally-consistent solutions. Finally, we describe the first effective DP algorithm for zs-POSGs, with finite-time convergence to an ϵ-optimal solution. Experiments support our theoretical findings.

We believe our approach complements existing ones, e.g., SFLP and CFR, in two dimensions. First, it breaks the original zs-POSG into nested subgames. Second, it generalizes values from visited subgames to unvisited ones. Our performances are as good as or better than those from SFLP and CFR for small-dimensional subgames (*e.g.,* with TOI structure). Unfortunately, the advantage of breaking the original problem into subgames and exploiting uniform continuity properties often fails to fully manifest in the overall computational time.

Despite some similarities, our (offline) approach is fundamentally different from (online) continual resolving approaches. The latter could even possibly be adapted to use other offline methods than CFR-based ones, including HSVI.

We hope that this approach will lay the foundation for further work in the area of both exact and approximate DP solutions for zs-POSGs. In the short term, we shall investigate and implement pruning methods (Section 3.4.1) and compression techniques (Section 3.4.2). In the long term, we shall investigate (deep) reinforcement learning for zs-POSGs, similarly to a recent approach for cp-POSGs (Bono et al. 2019). Of particular interest for (deep) reinforcement learning is the trade-off between the update-rule accuracy and the computational efficiency when facing high-dimensional subgames, hence providing competitive solvers.

4

N-Player Common-Payoff Games Under Hierarchical-Information Sharing

Contents

Optimally planning in cp-POSGs is a computationally difficult task in general, due to the intractable game tree, whose size grows double exponentially with respect to the planning horizon. This difficulty also comes from the non interchangeability of Nash equilibria, contrary to zs-POSGs: Players can not unilaterally compute their optimal behavior and must instead coordinate.

Solving schemes split between local and global ones, each coming with strengths and weaknesses. Local methods trade global optima, or ϵ-approximations thereof, for weaker solution concepts, *e.g.,* Nash equilibria that may not be globally optimal (Nair et al. 2003), or any arbitrary feasible solution. While local ones share core ideas with global ones, their primary focus is on solving relaxations of the original problem (*e.g.,* independent planners reason in isolation, policy gradients target first-order solutions of non-convex functions) (Tan 1998; Peshkin et al. 2001; Bono et al. 2018). Of particular attention, local methods using deep neural networks can apply effectively to virtually any non-critical application, *e.g.,* online services, logistics, or board games (Lowe et al. 2017; Foerster et al. 2018; Rashid et al. 2018).

In the contrary, solving cp-POSGs through cp-oMGs and HSVI-like algorithmic schemes (Dibangoye et al. 2016) offers theoretical guarantees which are key for high-stake applications (*e.g.,* search and rescue, security, healthcare). The cost is, however, that players' decision variables are entangled together in decision rule profiles at each time step.

Many real-life applications possess structure which can be exploited by algorithms, including those with theoretical guarantees, to find solutions empirically way faster than generic algorithms.

Multiple structures have been identified as of particular interest—*e.g.,* dynamics independence (Becker et al. 2004; Dibangoye et al. 2012), weak-separability (Nair et al. 2005; Dibangoye et al. 2014b), delayed information-sharing (Nayyar et al. 2010), one-sidedness (Horák et al. 2017; Horák et al. 2019b; Hadfield-Menell et al. 2016; Malik et al. 2018; Xie et al. 2020).

After reminding the reader of necessary background, we take a closer look at one particular structure, assuming that players are organized through some hierarchy. Players' knowledge about the game follow an inclusion relation, *i.e.,* player 1 only knows her individual history, player 2 knows what 1 knows, plus her own individual history, and so on, as detailed next.

4.1 Background

This background section comes back on the use of `cp-oMG`s for optimal planning in `cp-POSG`s. It presents the limitations in terms of scalability that the rest of the chapter will overcome through the exploitation of hierarchical information-sharing.

4.1.1 `cp-POSG`s

Let us recall that `cp-POSG`s are particular cases of `POSG`s but, contrary to `zs-POSG`s, it is sufficient to consider deterministic behavioral strategies (Oliehoek et al. 2008). In `cp-POSG`s, computing the optimal value function

$$V_\tau^* : \sigma_t \mapsto \max_{\boldsymbol{\beta}_{\tau:H-1}} \sum_{\tau=t}^{H-1} \gamma^{\tau-t} \mathbb{E}\left[r(S_t, A_t) \mid \beta_{\tau:H-1}\right] \tag{4.1}$$

is sufficient to retrieve a Nash equilibrium strategy profile with highest value. At each time step τ, the optimal value of an occupancy state σ_τ is also $V_\tau^*(\sigma_\tau) = \max_{\boldsymbol{\beta}_\tau} Q_\tau^*(\sigma_\tau, \boldsymbol{\beta}_\tau)$, where

$$Q_\tau^*(\sigma_\tau, \boldsymbol{\beta}_\tau) \stackrel{\text{def}}{=} \mathbb{E}\left[r(S_t, A_t) \mid \sigma_\tau\right] + \gamma V_{\tau+1}^*(T(\sigma_\tau, \boldsymbol{\beta}_\tau)). \tag{4.2}$$

Computing $V^*(\sigma_\tau)$ is equivalent to solving a common-payoff game $G_{\sigma_\tau} = \langle n, \times_i \mathcal{B}_\tau^i, \sigma_\tau, Q_\tau^*(\sigma_\tau, \cdot)\rangle$. In this game, private histories θ^i correspond to types of player i whose probabilities are derived from the occupancy state σ_τ and where the common-payoff function $Q_\tau^*(\sigma_\tau, \cdot)$ maps decision rule profiles $\boldsymbol{\beta}_\tau$ to values $Q_\tau^*(\sigma_\tau, \boldsymbol{\beta}_\tau)$.

In the following, for any history profile $\boldsymbol{\theta}_t$, hidden state s and any behavioral strategy profile $\boldsymbol{\beta}_{t:H-1}$, we note

$$\alpha_t^{\boldsymbol{\beta}_{t:H-1}} : s, \boldsymbol{\theta}_t \mapsto \mathbb{E}\{\sum_{\tau=t}^{H-1} \gamma^{\tau-t} r(S_\tau, A_\tau) \mid s, \boldsymbol{\theta}_t, \boldsymbol{\beta}_{t:H-1}\}.$$

Interestingly, Q^* exhibits strong continuity properties that allow reducing the complexity of solving G_{σ_τ}.

Lemma 4.1.1 (Dibangoye et al. (2018))**.** *For every time step τ, the optimal value function $Q_\tau^*\colon \mathcal{O}_\tau^\sigma \times \mathcal{B}_\tau \to \mathbb{R}$ is piecewise-linear and convex over occupancy states and decision rule profiles. In other words, there exists a finite collection $\mathcal{Q}_\tau \subseteq \{q_\tau^{\boldsymbol{\beta}_{\tau+1:}} | \boldsymbol{\beta}_{\tau+1:} \in \mathcal{B}_{\tau+1:}\}$ of action-value functions $q_\tau^{\boldsymbol{\beta}_{\tau+1:}}$ under behavioral strategy profile $\boldsymbol{\beta}_{\tau+1:}$, such that: for occupancy state σ_τ and decision rule profile $\boldsymbol{\beta}_\tau$,*

$$\begin{aligned} Q_\tau^*(\sigma_\tau, \boldsymbol{\beta}_\tau) &= \max_{q_\tau \in \mathcal{Q}_\tau} \ \mathbb{E}_{(s,\boldsymbol{\theta}_\tau,\boldsymbol{a})\sim \Pr\{\cdot|\sigma_\tau,\boldsymbol{\beta}_\tau\}}\{q_\tau(s, \boldsymbol{\theta}_\tau, \boldsymbol{a})\}, \\ \text{where } q_\tau^{\boldsymbol{\beta}_{\tau+1:}}(s, \boldsymbol{\theta}_\tau, \boldsymbol{a}) &= r(s, \boldsymbol{a}) + \gamma \mathbb{E}_{(s',\boldsymbol{z})\sim p(\cdot|s,\boldsymbol{a})}\{\alpha_{t+1}^{\boldsymbol{\beta}_{\tau+1:}}(s', (\boldsymbol{\theta}_\tau, \boldsymbol{z}, \boldsymbol{a}))\}, \end{aligned} \tag{4.3}$$

with boundary condition $\alpha_H^{\cdot}(\cdot) = q_H^{\cdot}(\cdot) \stackrel{\text{def}}{=} 0$.

Lemma 4.1.1 allows us to optimally solve the common-payoff Bayesian games G_{σ_τ} by taking the best among solutions of single-stage subgames $G_{\sigma_\tau}^{q_\tau} \stackrel{\text{def}}{=} \langle n, \times_i \mathcal{B}_\tau^i, Q_{q_\tau}(s_\tau, \cdot)\rangle$ induced by action-value function $q_\tau \in \mathcal{Q}_\tau$, where $Q_{\beta_\tau}(\sigma_\tau, \boldsymbol{\beta}_\tau) \stackrel{\text{def}}{=} \mathbb{E}_{(s,\boldsymbol{\theta}_\tau,\boldsymbol{a})\sim \Pr\{\cdot|\sigma_\tau,\boldsymbol{\beta}_\tau\}}\{q_\tau(s, \boldsymbol{\theta}_\tau, \boldsymbol{a})\}$. In particular, the problem of maximizing $\boldsymbol{\beta}_\tau \mapsto Q_{q_\tau}(\sigma_\tau, \boldsymbol{\beta}_\tau)$, which is equivalent to a common-payoff Bayesian game, will play a crucial role in disentangling decision variables.

From now on, we will make the following hierarchical information-sharing assumption.

Assumption 4.1.2. *Planning horizon H is finite.*

Remark 4.1.3. *Regarding infinite-horizon problems, the infinite-horizon solutions are ϵ-close to H-horizon optimal solutions, where $H = \lceil \log_\gamma (1-\gamma)\epsilon/c \rceil$, for discount factor $\gamma \in [0,1)$ and scalar $\epsilon > 0$.*

Assumption 4.1.4. *Every player i has instantaneous and cost-free access to their subordinate actions and observations at every time step τ*—i.e., $(a^i_{\tau-1}, z^i_\tau) \sqsubseteq z^{i+1}_\tau$. *Player 1's actions and observations are public to all other players (*i.e., *is at the bottom of the hierarchy); and player n sees all actions and observations of other players (*i.e., *is at the top of the hierarchy).*

We refer to `cp-POSG`s with hierarchical information-sharing as `his-cp-POSG`s.

4.1.2 Limitations of Single-Player Reformulations

The methodology suggesting to solve an `his-cp-POSG` through the derived `his-cp-oMG` applies, but the curse of dimensionality restricts its scalability in the face of games with many players. To better understand this, notice that the complexity of optimally solving a `his-cp-oMG` using point-based algorithmic schemes (*e.g.,* `HSVI` or `PBVI`) depends on two operators: the point-based backup operator, which optimally solves single-stage games $G^{q_\tau}_{\sigma_\tau}$, and the estimation operator that maintains occupancy states. In either case, the `HIS` assumption is not leveraged. State-of-the-art approaches to solving $G^{q_\tau}_{\sigma_\tau}$ perform either brute-force or implicit enumeration and evaluation of double-exponentially many decision-rule profiles (Oliehoek et al.; Dibangoye et al.; Dibangoye et al.; Dibangoye et al., 2010; 2009; 2013; 2016). This provides an intuitive explanation for the negative complexity results: optimally solving $G^{q_\tau}_{\sigma_\tau}$ is `NP-hard`, and finding ϵ-approximations remains hard (Tsitsiklis, 1984). The estimation operator also suffers from the curse of dimensionality. Indeed, the number of decision variables of all players involved grows exponentially with time and team size. The following sections aim at leveraging the `HIS` assumption, whereas a more concise description of occupancy states (*e.g.,* through lossless compression techniques) is left for future work.

4.2 Hierarchical Information Sharing

This section leverages the `HIS` assumption to reduce the time complexity of solving Bayesian games $G^{q_\tau}_{\sigma_\tau}$.

4.2.1 From Single-Stage to Extensive-Form Games

The application of Bellman's optimality principle to `cp-POSG`s typically introduces a temporal decomposition that allows computing a solution of G_{σ_0} recursively. The recursion at time step τ involves solving games $G^{q_\tau}_{\sigma_\tau}$. Here, we complement this by introducing a player-based decomposition for the computation of each game $G^{q_\tau}_{\sigma_\tau}$. Let us consider a perfect-information game (Shoham et al. 2008) (*cf.* Figure 4.1) starting with player 1 at the bottom of the hierarchy that chooses action a^1_τ according to its total available information $\varsigma^1_\tau = (\sigma_\tau, \theta^1_\tau)$. The game then randomly lands on total available information $\varsigma^2_\tau = (\varsigma^1_\tau, a^1_\tau, \theta^2_\tau)$, for which player 2, the next player in the reversed order of the hierarchy, chooses action a^2_τ. The process continues until the game randomly lands on total available information $\varsigma^n_\tau = (\varsigma^{n-1}_\tau, a^{n-1}_\tau, \theta^n_\tau)$. Player n chooses an action a^n_τ and receives expected rewards $R(\varsigma^n_\tau, a^n_\tau) \overset{\text{def}}{=} \mathbb{E}_{s\sim\Pr\{\cdot|\varsigma^n_\tau, a^n_\tau\}}\{q_\tau(s, \boldsymbol{\theta}_\tau, a_\tau)\}$ upon taking action a^n_τ in information state ς^n_τ. Note that the expectancy is well-defined as θ^n_τ contains histories of players 1 to $n-1$ and is consequently equal to a joint history $\boldsymbol{\theta}_\tau$.

More formally, the game can be described by a perfect-information extensive form game[31] in which nodes for player i correspond to total information ς^i_τ. The probability $T(\varsigma^i_\tau, a^i_\tau, \tilde{\varsigma}^{i+1}_\tau)$ that the game moves a node ς^i_τ to another one $\tilde{\varsigma}^{i+1}_\tau = \langle \tilde{\varsigma}^i_\tau, \tilde{a}^i_\tau, \tilde{\theta}^{i+1}_\tau \rangle$ is $\Pr\{\theta^{i+1}_\tau | \sigma_\tau, \theta^1_\tau, \dots, \theta^i_\tau\} \cdot \delta^{\tilde{\varsigma}^{i+1}_\tau}_{\varsigma^i_\tau, a^i_\tau, \theta^{i+1}_\tau}$.

[31] Information sets are reduced to single nodes.

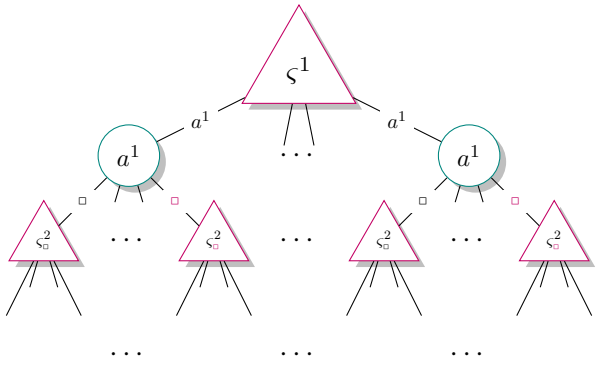


Figure 4.1: The search space for a single-stage game using player-based decomposition, illustrated as an AND/OR tree. OR nodes (triangle) represent alternative ways to solve $\bar{G}^{q_\tau}_{\sigma_\tau}$. AND nodes (circle) represent subproblem alternatives to be solved.

Definition 4.2.1. *The common-payoff perfect-information extensive-form game w.r.t.* $G^{\beta_\tau}_{\sigma_\tau}$ *is a tuple* $\bar{G}^{q_\tau}_{\sigma_\tau} \stackrel{\text{def}}{=} \langle n, \Sigma, \cup_i \mathcal{A}^i, T, R \rangle$ *where:*

- n *is the number of players;*
- $\Sigma = \cup_{i=1}^n \Sigma_i$ *is the set of nodes* ς^i_τ *for all players* i*, induced by* σ_τ*;*
- *for any player* i*,* $\mathcal{A}^i$ *is as in the* `cp-POSG` *and corresponds here to her set of actions*[32]*;*
- *transition function* $T\colon \Sigma \times (\cup_{i=1}^n \mathcal{A}^i) \times \Sigma \to [0,1]$ *specifies the probability of a successor node;*
- *reward function* $R\colon \Sigma^n \times (\cup_{i=1}^n A^i) \to \mathbb{R}$ *specifies the common payoff received upon taking an action in a leaf.*

4.2.2 Optimally Solving $G^{q_\tau}_{\sigma_\tau}$ As $\bar{G}^{q_\tau}_{\sigma_\tau}$

Optimally solving a common-payoff perfect-information extensive-form game aims at finding the action-value functions $\gamma_\tau^{1:n,*}$ mapping nodes and actions to optimal values. Unlike the original single-stage game $G^{q_\tau}_{\sigma_\tau}$, the perfect information extensive form game $\bar{G}^{q_\tau}_{\sigma_\tau}$ makes the `HIS` structure explicit. Every time a player acts, she is perfectly informed about all the histories that have previously occurred—*i.e.,* all histories of its subordinates. Hence, the total information nodes include the actions of subordinate players of the current player, along with the histories of its subordinates. Nonetheless, both games yield the same solution.

Theorem 4.2.2. *Any optimal solution for* $\bar{G}^{q_\tau}_{\sigma_\tau}$ *is also an optimal solution for* $G^{q_\tau}_{\sigma_\tau}$*. Besides, the optimal action-value functions* $\gamma_\tau^{1:n,*}$ *of* $\bar{G}^{q_\tau}_{\sigma_\tau}$ *are the solution of the Bellman optimality equations: for any* i*,* ς^i_τ*, and* a^i_τ*,*

$$\gamma^{i,*}_\tau(\varsigma^i_\tau, a^i_\tau) = \mathbb{E}_{\varsigma^{i+1}_\tau \sim T(\cdot|\varsigma^i_\tau, a^i_\tau)}\{\max_{a^{i+1}_\tau} \gamma^{i+1,*}_\tau(\varsigma^{i+1}_\tau, a^{i+1}_\tau)\},$$

with boundary condition $\gamma^{n,*}_\tau\colon (\varsigma^n_\tau, a^n_\tau) \mapsto R(\varsigma^n_\tau, a^n_\tau)$*. Also, greedy decision rule* $\beta^{i,*}_\tau$ *for any player* i *at* θ^i_τ *is:*

$$\beta^{i,*}_\tau(\theta^i_\tau) \in \arg\max_{a^i_\tau} \gamma^{i,*}_\tau(\varsigma^i_\tau, a^i_\tau),$$

where $\varsigma^i_\tau \stackrel{\text{def}}{=} \langle \sigma_\tau, \boldsymbol{\theta}^{1:i}_\tau, \boldsymbol{\beta}^{1:i-1,*}_\tau(\boldsymbol{\theta}^{1:i-1}_\tau) \rangle$.

Proof. The proof proceeds in two steps. First, we show that the original game $G^{q_\tau}_{\sigma_\tau}$ can alternatively be solved sequentially, by breaking $G^{q_\tau}_{\sigma_\tau}$ down into smaller subgames

$$\langle G^{q_\tau}_{\sigma_\tau,\varnothing}, G^{q_\tau}_{\sigma_\tau,\beta^1_\tau}, \ldots, G^{q_\tau}_{\sigma_\tau,\boldsymbol{\beta}^{1:n-1}_\tau} \rangle,$$

[32] Similarly to `cp-oMGs`, and given that we are trying to solve a `cp-oMG`, we assume that any action $a^i \in \mathcal{A}^i$ is valid for player i, for any node ς^i_τ.

one subgame per player. To this end, recall the goal of optimally solving $G^{q_\tau}_{\sigma_\tau}$, *i.e.,* finding a decision rule profile which yields the highest performance index, $V_{q_\tau}(\sigma_\tau) \overset{\text{def}}{=} \max_{\boldsymbol{\beta}_\tau} \; Q_{q_\tau}(\sigma_\tau, \boldsymbol{\beta}_\tau)$. The expansion of decision rule profile $\boldsymbol{\beta}_\tau$ as an n-tuple of private decision rules $(\beta^1_\tau, \beta^2_\tau, \dots, \beta^n_\tau)$ allows to rewrite the objective of $G^{q_\tau}_{\sigma_\tau}$ as follows, $V_{q_\tau}(\sigma_\tau) = \max_{\beta^1_\tau} \max_{\beta^2_\tau} \dots \max_{\beta^n_\tau} \; Q_{q_\tau}(\sigma_\tau, \beta_\tau)$. Let $Q^i_{q_\tau}(\sigma_\tau, \cdot)\colon \boldsymbol{\beta}^{1:i}_\tau \mapsto \max_{\boldsymbol{\beta}^{i+1:n}_\tau} \; Q_{q_\tau}(\sigma_\tau, \boldsymbol{\beta}_\tau)$ be a sequential action-value function. Then, it follows that

$$\begin{aligned} V_{q_\tau}(\sigma_\tau) &= \max_{\beta^1_\tau} \max_{\beta^2_\tau} \dots \max_{\beta^n_\tau} \; Q_{q_\tau}(\sigma_\tau, \boldsymbol{\beta}_\tau), \\ &= \max_{\beta^1_\tau} \; \left[\max_{\beta^2_\tau} \dots \max_{\beta^n_\tau} \; Q_{q_\tau}(\sigma_\tau, \boldsymbol{\beta}_\tau) \right], \\ &= \max_{\beta^1_\tau} \; Q^1_{q_\tau}(\sigma_\tau, \beta^1_\tau). \end{aligned}$$

Interestingly, for every player $i \in \{1, 2, \dots, n-1\}$, the action-value functions $Q^i_{q_\tau}(\sigma_\tau, \boldsymbol{\beta}^{1:i}_\tau)$ satisfy the following recursion

$$\begin{aligned} Q^i_{q_\tau}(\sigma_\tau, \boldsymbol{\beta}^{1:i}_\tau) &= \max_{\beta^{i+1}_\tau} \max_{\beta^{i+2}_\tau} \dots \max_{\beta^n_\tau} \; Q_{q_\tau}(\sigma_\tau, \boldsymbol{\beta}_\tau), \\ &= \max_{\beta^{i+1}_\tau} \; \left[\max_{\beta^{i+2}_\tau} \dots \max_{\beta^n_\tau} \; Q_{q_\tau}(\sigma_\tau, \boldsymbol{\beta}_\tau) \right], \\ &= \max_{\beta^{i+1}_\tau} \; Q^{i+1}_{q_\tau}(\sigma_\tau, \boldsymbol{\beta}^{1:i+1}_\tau), \end{aligned}$$

with boundary condition $Q^n_{q_\tau}(\sigma_\tau, \boldsymbol{\beta}_\tau) \overset{\text{def}}{=} Q_{q_\tau}(\sigma_\tau, \boldsymbol{\beta}_\tau)$. For any arbitrary player $i \in \{2, 3, \dots, n\}$, define game $G^{q_\tau}_{\sigma_\tau, \boldsymbol{\beta}^{1:i-1}_\tau} \overset{\text{def}}{=} \langle i, \mathcal{A}^i, Q^i_{q_\tau}(\sigma_\tau, \boldsymbol{\beta}^{1:i-1}_\tau, \cdot) \rangle$ to be the subgame upon the selected decision rules $\boldsymbol{\beta}^{1:i-1}_\tau$ starting in game $G^{q_\tau}_{\sigma_\tau}$, with boundary condition $G^{q_\tau}_{\sigma_\tau, \varnothing} \overset{\text{def}}{=} \langle 1, \mathcal{A}^1, Q^1_{q_\tau}(\sigma_\tau, \cdot) \rangle$. Consequently, optimally solving the original game $G^{q_\tau}_{\sigma_\tau}$ can be performed by optimally solving smaller subgames $\langle G^{q_\tau}_{\sigma_\tau, \varnothing}, G^{q_\tau}_{\sigma_\tau, \beta^1_\tau}, \dots, G^{q_\tau}_{\sigma_\tau, \boldsymbol{\beta}^{1:n-1}_\tau} \rangle$, one subgame per player, recursvively.

Next, we shall prove that the best decision rule in any arbitrary sequential-move subgame $G^{q_\tau}_{\sigma_\tau, \boldsymbol{\beta}^{1:i-1}_\tau}$ depends on the current occupancy state σ_τ along with previously selected decision rules $\boldsymbol{\beta}^{1:i-1}_\tau$, only through the corresponding nodes $\varsigma^i_\tau \overset{\text{def}}{=} (\sigma_\tau, \boldsymbol{a}^{1:i-1}_\tau, \boldsymbol{\theta}^{1:i}_\tau)$ of the perfect information extensive form game $\bar{G}^{q_\tau}_{\sigma_\tau}$. In other words, instead of selecting actions for all private histories of player i in sync, one can choose the best action for each private history independently without compromising optimality. The proof of this statement proceeds by induction from player n to player 1. At player n, the greedy decision rule $\hat{\beta}^n_\tau$ satisfies the following:

$$\begin{aligned} \hat{\beta}^n_\tau &\in \arg\max_{\beta^n_\tau} \; Q^n_{q_\tau}(\sigma_\tau, \boldsymbol{\beta}_\tau), \\ &\in \arg\max_{\beta^n_\tau} \; Q_{q_\tau}(\sigma_\tau, \boldsymbol{\beta}_\tau), \\ &\in \arg\max_{\beta^n_\tau} \; \mathbb{E}_{(s, \boldsymbol{\theta}, \boldsymbol{a}) \sim \Pr\{\cdot | \sigma_\tau, \boldsymbol{\beta}_\tau\}} \{ q_\tau(s, \boldsymbol{\theta}, \boldsymbol{a}) \}. \end{aligned}$$

Expanding over private histories of player n, we have that

$$\hat{\beta}^n_\tau(\theta^n_\tau) \in \arg\max_{a^n_\tau} \; \mathbb{E}_{(s, \boldsymbol{\theta}, \boldsymbol{a}) \sim \Pr\{\cdot | \sigma_\tau, \theta^n_\tau, \boldsymbol{\beta}_\tau\}} \{ q_\tau(s, \boldsymbol{\theta}, \boldsymbol{a}) \}.$$

Leveraging information available to player n as provided by the `HIS` assumption, we know that the knowledge of private history θ^n_τ implies the knowledge of histories of all other players $\boldsymbol{\theta}^{1:n-1}_\tau$, hence the joint history $\boldsymbol{\theta}_\tau$, *i.e.,*

$$\hat{\beta}^n_\tau(\theta^n_\tau) \in \arg\max_{a^n_\tau} \; \mathbb{E}_{s \sim \Pr\{\cdot | \sigma_\tau, \boldsymbol{\theta}_\tau, \boldsymbol{\beta}_\tau\}} \{ q_\tau(s, \boldsymbol{\theta}, \boldsymbol{a}) \}.$$

In addition, the knowledge of $\boldsymbol{\theta}^{1:n-1}_\tau$ together with the decision rules $\boldsymbol{\beta}^{1:n-1}_\tau$ of subordinates, makes it possible to access node $\varsigma^n_\tau \overset{\text{def}}{=} \langle \sigma_\tau, \boldsymbol{\theta}^{1:n}_\tau, \boldsymbol{\beta}^{1:n-1}_\tau(\boldsymbol{\theta}^{1:n-1}_\tau) \rangle$ such that:

$$\hat{\beta}^n_\tau(\theta^n_\tau) \in \arg\max_{a^n_\tau} \; \gamma^n_\tau(\varsigma^n_\tau, a^n_\tau),$$

where $\gamma_\tau^n \colon (\varsigma_\tau^n, a_\tau^n) \mapsto \mathbb{E}_{s\sim \Pr\{\cdot|\varsigma_\tau^n, a_\tau^n\}}\{\beta_\tau(s, \boldsymbol{\theta}, \boldsymbol{a})\}$, which proves the statement holds at player n. Define function $\alpha_\tau^n \colon \varsigma_\tau^n \mapsto \max_{a_\tau^n} \gamma_\tau^n(\varsigma_\tau^n, a_\tau^n)$ at player n. Notice that the value of the sequential-move subgame $G^{\beta_\tau}_{\sigma_\tau, \boldsymbol{\beta}_\tau^{1:n-1}}$ can be rewritten as follows:

$$Q_{q_\tau}^{n-1}(\sigma_\tau, \boldsymbol{\beta}_\tau^{1:n-1}) = \max_{\beta_\tau^n} Q_{q_\tau}^n(\sigma_\tau, \boldsymbol{\beta}_\tau) \tag{4.4}$$

$$= \mathbb{E}_{\varsigma_\tau^n \sim \Pr\{\cdot|\sigma_\tau, \boldsymbol{\beta}_\tau^{1:n-1}\}}\{\max_{a_\tau^n} \gamma_\tau^n(\varsigma_\tau^n, a_\tau^n)\} \tag{4.5}$$

$$= \mathbb{E}_{\varsigma_\tau^n \sim \Pr\{\cdot|\sigma_\tau, \boldsymbol{\beta}_\tau^{1:n-1}\}}\{\alpha_\tau^n(\varsigma_\tau^n)\}. \tag{4.6}$$

Suppose the statement holds for any player $i > 1$, with greedy decision rule $\hat{\beta}_\tau^i(\theta_\tau^i) \in \arg\max_{a_\tau^i} \gamma_\tau^i(\varsigma_\tau^i, a_\tau^i)$. Define function $\alpha_\tau^i \colon \varsigma_\tau^i \mapsto \max_{a_\tau^i} \gamma_\tau^i(\varsigma_\tau^i, a_\tau^i)$ at player i. Also, the value of the sequential-move subgame $G^{q_\tau}_{\sigma_\tau, \boldsymbol{\beta}_\tau^{1:i-1}}$ can be rewritten by expanding over the sequential-move nodes $\varsigma_\tau^i \stackrel{\text{def}}{=} \langle \sigma_\tau, \boldsymbol{\theta}_\tau^{1:i}, \boldsymbol{\beta}_\tau^{1:i-1}(\boldsymbol{\theta}_\tau^{1:i-1}) \rangle$, *i.e.*,

$$Q_{q_\tau}^{i-1}(\sigma_\tau, \boldsymbol{\beta}_\tau^{1:i-1}) = \mathbb{E}_{\varsigma_\tau^i \sim \Pr\{\cdot|\sigma_\tau, \boldsymbol{\beta}_\tau^{1:i-1}\}}\{\alpha_\tau^i(\varsigma_\tau^i)\}. \tag{4.7}$$

We are now ready to prove the statement also holds at player $i-1$. For player $i-1$, decision rule $\hat{\beta}_\tau^{i-1}$ satisfies the following expression:

$$\begin{aligned}\hat{\beta}_\tau^{i-1} &\in \arg\max_{\beta_\tau^{i-1}} \; Q_{q_\tau}^{i-1}(\sigma_\tau, \boldsymbol{\beta}_\tau^{1:i-1}),\\ &\in \arg\max_{\beta_\tau^{i-1}} \mathbb{E}_{\varsigma_\tau^i \sim \Pr\{\cdot|\sigma_\tau, \boldsymbol{\beta}_\tau^{1:i-1}\}}\{\alpha_\tau^i(\varsigma_\tau^i)\}.\end{aligned}$$

Similarly to player n, the knowledge of $\boldsymbol{\theta}_\tau^{1:i-1}$ together with the decision rules $\boldsymbol{\beta}_\tau^{1:i-2}$ of subordinates, makes it possible to access node $\varsigma_\tau^{i-1} \stackrel{\text{def}}{=} \langle \sigma_\tau, \boldsymbol{\theta}_\tau^{1:i-1}, \boldsymbol{\beta}_\tau^{1:i-2}(\boldsymbol{\theta}_\tau^{1:i-2}) \rangle$ such that:

$$\hat{\beta}_\tau^{i-1}(\theta_\tau^{i-1}) \in \arg\max_{a_\tau^{i-1}} \; \gamma_\tau^{i-1}(\varsigma_\tau^{i-1}, a_\tau^{i-1}),$$

where $\gamma_\tau^{i-1} \colon (\varsigma_\tau^{i-1}, a_\tau^{i-1}) \mapsto \mathbb{E}_{\varsigma_\tau^i \sim \Pr\{\cdot|\varsigma_\tau^{i-1}, a_\tau^{i-1}\}}\{\alpha_\tau^i(\varsigma_\tau^i)\}$, which proves the statement holds at player $i-1$. Define function $\alpha_\tau^{i-1} \colon \varsigma_\tau^{i-1} \mapsto \max_{a_\tau^{i-1}} \gamma_\tau^{i-1}(\varsigma_\tau^{i-1}, a_\tau^{i-1})$ at player $i-1$. Consequently, the value of the sequential-move subgame $G^{q_\tau}_{\sigma_\tau, \varnothing}$ can be rewritten by expanding over the sequential-move nodes $\varsigma_\tau^1 \stackrel{\text{def}}{=} \langle \sigma_\tau, \theta_\tau^1 \rangle$, *i.e.*,

$$V_{q_\tau}(\sigma_\tau) = \mathbb{E}_{\varsigma_\tau^1 \sim \Pr\{\cdot|\sigma_\tau\}}\{\alpha_\tau^1(\varsigma_\tau^1)\}.$$

The value of a cooperative game being unique, we know the optimal solution for $\bar{G}^{q_\tau}_{\sigma_\tau}$ is also an optimal solution for $G^{q_\tau}_{\sigma_\tau}$. In demonstrating this statement, we also exhibited Bellman's optimality equations, providing the solution of the perfect-information extensive-form game $\bar{G}^{q_\tau}_{\sigma_\tau}$, *i.e.,* at any player i, node ς_τ^i, and action a_τ^i,

$$\gamma_\tau^{i,*}(\varsigma_\tau^i, a_\tau^i) = \mathbb{E}_{\varsigma_\tau^{i+1} \sim T(\cdot|\varsigma_\tau^i, a_\tau^i)}\{\max_{\boldsymbol{a}_\tau^{i+1}} \gamma_\tau^{i+1,*}(\varsigma_\tau^{i+1}, \boldsymbol{a}_\tau^{i+1})\},$$

with boundary condition $\gamma_\tau^{n,*} \colon (\varsigma_\tau^n, a_\tau^n) \mapsto R(\varsigma_\tau^n, a_\tau^n)$. This ends the proof. □

Theorem 4.2.2 introduces Bellman optimality equations that enable us to find a greedy joint decision at single-stage subgame $G^{q_\tau}_{\sigma_\tau}$ by solving the corresponding extensive-form game $\bar{G}^{q_\tau}_{\sigma_\tau}$. It proceeds in two phases. From player n at the top of the hierarchy to player 1 at the bottom, a backward pass computes optimal action-values $\gamma_\tau^{i,*}(\varsigma_\tau^i, a_\tau^i)$ for each player i, each node ς_τ^i, and each action a_τ^i. Then, from player 1 at the bottom of the hierarchy to player n at the top, a forward pass selects a greedy decision rule independently for each player i, and each node ς_τ^i. This backward induction algorithm requires a time complexity linear in the number of players, nodes, and actions $\mathbf{O}(n|\Sigma||\mathcal{A}^*|)$ instead of double exponential $\mathbf{O}(|\Theta^*|^{|\mathcal{A}^*|^n})$ where $\Theta^* \stackrel{\text{def}}{=} \arg\max_{\Theta^i} |\Theta^i|$ with Θ^i being the set of reachable histories of player i in σ_τ and $\mathcal{A}^* \stackrel{\text{def}}{=} \arg\max_{\mathcal{A}^i} |\mathcal{A}^i|$. A careful reader would notice that the linearity of q_τ over occupancy states and joint decision rules is key in demonstrating Theorem 4.2.2.

4.3 Near-Optimally Solving his-cp-POSGs

This section adapts the point-based value-iteration (PBVI) algorithm (Pineau et al., 2003) to compute an ϵ-optimal strategy profile for cp-POSGs under HIS starting at initial state distribution σ_0 for planning horizon H. We chose the PBVI algorithm because it leverages the linear functions q_τ involved in the optimal value function. Besides, it is guaranteed to find near-optimal solutions asymptotically. Notice that some algorithmic schemes such as HSVI might benefit our findings. HSVI's classical lower bound can be adequately modified to increase the efficiency of update computations, but technical challenges need to be addressed regarding the upper bound, which involves non-linearities (either using convex hulls or sawtooth approximations).

This section presents a pseudocode for the point-based value iteration algorithm to solve common-payoff partially observable stochastic games with hierarchical information sharing near-optimally.

Algorithm 4.1: PBVI for cp-oMGs under HIS.

```
function PBVI() Initialize Õ^σ_{0:} and V_{0:}
while V_{0:} has not converged do
    improve(V_{0:}, Õ^σ_{0:})
    Õ^σ_{0:} ← expand(Õ^σ_{0:})
function Improve(Õ^σ_{0:}, Q_{0:})
for τ = ℓ − 1 to 0 do
    for s_τ ∈ Õ^σ_τ do
        V_τ ← V_τ ∪ {backup(s_τ, V_{τ+1})}
```

PBVI, *cf.* Algorithm 4.1, has two main parts for solving a cp-oMG under HIS. First, it bounds the size of the value function at each stage τ of the game by representing the value only at a finite, reachable occupancy subset $\tilde{\mathcal{O}}^\sigma_\tau$. Next, it optimizes the value function represented as a collection $\mathcal{V}_\tau$ at each stage τ using point-based backup, *i.e.,* at any stage τ, $\mathcal{V}_\tau = \{\texttt{backup}(\sigma_\tau, \mathcal{V}_{\tau+1}) \colon \sigma_\tau \in \tilde{\mathcal{O}}^\sigma_\tau\}$, where backups are executed in no particular order, *i.e.,*

$$\texttt{backup}(\sigma_\tau, \mathcal{V}_{\tau+1}) = \underset{\alpha^{\boldsymbol{\beta}_{\tau:}}_\tau \colon \boldsymbol{\beta}_\tau \in \mathcal{B},\, \alpha^{\boldsymbol{\beta}_{\tau+1:}}_\tau \in \mathcal{V}_\tau}{\arg\max} Q_{q^{\boldsymbol{\beta}_{\tau+1:}}_\tau}(\sigma_\tau, \boldsymbol{\beta}_\tau).$$

This representation always lower bounds the optimal value function. Each iteration of PBVI traverses occupancy-state subsets bottom up. This iterative process repeats until convergence or until a budget, *e.g.,* CPU time, memory, or number of iterations, has been exhausted. The algorithm adds supplemental points into occupancy subsets to improve the value functions further. It selects candidate points using a portfolio of exploration strategies, including random explorations and greedy w.r.t. underlying (PO)MDP value functions. For every stage τ, the algorithm adds only candidate points beyond a certain distance from the occupancy subset $\tilde{\mathcal{O}}^\sigma_\tau$ to create a new occupancy-state set $\tilde{\mathcal{O}}^\sigma_{\tau+1}$.

For any arbitrary occupancy-state set $\tilde{\mathcal{O}}^\sigma_{0:}$, PBVI produces a value $v_0(\sigma_0)$. The error between $v_0(\sigma_0)$ and $v^*_0(\sigma_0)$ is bounded. The bound depends on how $\tilde{\mathcal{O}}^\sigma_{0:}$ samples the entire occupancy-state space; with denser sampling, the estimate $v_0(\sigma_0)$ converges to $v^*_0(\sigma_0)$. The remainder of this section states and proves PBVI's approximation error.

Define the density $\delta_{\tilde{\mathcal{O}}^\sigma_{0:}}$ to be the maximum distance from any legal occupancy state to sets $\tilde{\mathcal{O}}^\sigma_{0:}$. More precisely, $\delta_{\tilde{\mathcal{O}}^\sigma_{0:}} \overset{\text{def}}{=} \max_{\tau \in [\![0:H-1]\!]} \max_{\sigma \in \mathcal{O}^\sigma_\tau} \min_{\sigma' \in \tilde{\mathcal{O}}^\sigma_\tau} \|\sigma - \sigma'\|_1$. Define a positive scalar r_{max} such that $\|r(\cdot,\cdot)\|_\infty \leqslant r_{max}$.

Theorem 4.3.1. *For any occupancy subsets $\tilde{\mathcal{O}}^\sigma_{0:}$, the error of the PBVI algorithm is bounded by*

$$v^*_0(\sigma_0) - v_0(\sigma_0) \leqslant 2 r_{max} \delta_{\tilde{\mathcal{O}}^\sigma_{0:}} \frac{1 + H\gamma^{H+1} - (H+1)\gamma^H}{(1-\gamma)^2}.$$

Proof. The proof is a direct adaptation of Pineau et al. (2003)'s one by approximating the error of a finite-horizon problem instead of an infinite-horizon one. □

It is worth noticing that whenever H goes to infinity, our bound meets that from Pineau et al. (2003) for infinite-horizon partially observable Markov decision processes.

4.4 Experiments

This section presents the outcomes of our experiments, which were carried out to juxtapose our findings with the leading-edge theory employed in global methods, encompassing the utilization of the PBVI algorithm as a standard algorithmic scheme. Our analysis involves three variants of the PBVI algorithm, namely PBVI^{enum}, PBVI^{milp}, and hPBVI, each employing distinct methods of performing point-based backups. PBVI^{enum} relies on brute-force enumeration of joint decision rules. At the same time, PBVI^{milp} utilizes mixed-integer linear programs (MILPs) for implicit enumeration, following the state-of-art approach for general cp-POSGs (Dibangoye et al., 2016). We used ILOG CPLEX Optimization Studio to solve the MILPs. Finally, hPBVI incorporates our findings to facilitate point-based backups under hierarchical information sharing. Global methods are not designed to scale up with the number of players. To present a comprehensive view, we have also compared our results against local policy- and value-based methods, *i.e.,* asynchronous actor-critic (A2C) (Konda et al. 1999) and independent Q-learning (IQL) (Tan 1998), respectively. The experiments were executed on an Ubuntu machine with 32 GB of available RAM and a 2.5 GHz processor, utilizing only one core, with a time limit of 30 minutes (except Tiger, for which all algorithms were given 1 hour).

We have comprehensively assessed various algorithms using several two-player benchmarks sourced from academic literature, available at https://masplan.org. These benchmarks encompass Multi-Agent Broadcast Channel (MABC), Recycling Robots (recycling), Meeting in a Grid (Grid3x3), and Decentralized Tiger (tiger). To enable a comparison of multiple players, we have also introduced the multi-player variants of these benchmarks.

Multi-player Tiger The single-player tiger problem was first introduced by Kaelbling et al. (1998) and was later generalized to a two-player version by Nair et al. (2003). This game describes a scenario where players face two closed doors, one of which conceals a treasure while the other hides a dangerous tiger. Neither player knows which door leads to the treasure and which one leads to the tiger, but they can receive partial and noisy information about the tiger's location by listening. At any given time, each player can choose to open either the left or right door, which will either reveal the treasure or the tiger, and reset the game. To gain more information about the tiger's location, players can listen to hear the tiger on the left or right side, but with uncertain accuracy. We have extended this problem to an n-player version by incorporating hierarchical information-sharing and modifying the transition, observation, and reward models, while ensuring that the original two-player problem can still be recovered.

In our n player version of the tiger problem, only the reward function is not straightforwardly adapted. Listening costs -1, as in the original problem. Now, the penalty for opening the wrong door is set to $-100/n_w$, where n_w is the number of players opening the bad door (doing so, we retrieve the original problem for $n = 2$), and the reward for opening the good door is 10 for each player opening the good door.

Multi-player Recycling Robot The recycling robot task was first introduced by Sutton et al. (1998) as a single-player problem. Later on, Amato et al. (2012) generalized it to a two-player version. The multi-player formulation requires robots to work together to recycle soda cans. In this problem, both robots have a battery level, which can be either high or low. They have to choose between collecting small or big cans and recharging their own battery level. Collecting small or big cans can decrease the robot's battery level, with a higher probability when collecting the big waste. When a robot's battery is completely exhausted, it needs to be picked up and

placed onto a recharging spot, which results in a negative reward. The coordination problem arises since robots cannot pick up a big can independently.

In our n-player version of the problem, transition and observation functions are straightforwardly adapted as the classical problem with two players (and consequently, so are the problems we generated for more than two players) is transition- and history-independent. Picking up small cans rewards the same for each agent as in the two-player version. A reward of $+5$ multiplied by the number of players is given if and only if *all* players try to pick up a big can, while a penalty -10 is given if not all agents synchronize to carry the big can.

Multi-player Broadcast Channel Ooi et al. (1996) introduced a scenario in which a unique channel is shared by n players, who aim at transmitting packets. The time is discretized, and only one packet can be transmitted at each time step. If two or more players attempt to send a packet at the same time, the transmission fails due to a collision. Hansen et al. (2004) extended this problem to a partially observable one, focusing on two players (Hansen et al., 2004). We used similar adaptations to define a partially observable version of the original n-player broadcast channel.

Multi-player Grid3x3 This problem was first introduced by Amato et al. (2009). It involves two players who want to meet each other as soon as possible on a two-dimensional grid. Each player has five possible actions: moving north, south, west, east, or staying in place. To simulate an uncertain environment, each player's action has a fixed probability of being successful. Additionally, each player can only sense their own location and has no knowledge of the other player's location. To adapt the problem for multiple players, we placed n players on the grid, each with the same actions and perceptions as described above. The reward has been redefined as the largest number of players minus one present at one of the two meeting points. This way, the original problem can be retrieved for two players.

We conducted three sets of experiments to assess our findings:

1. To assess the exponential drop in time complexity of backups with respect to an increasing number of players, we measure the average time required to perform a single backup, *cf.* Section 4.4.1 – Average Backup Time for Increasingly Many Players.

2. To assess the exponential drop in time complexity of backups with respect to increasing horizons, we measure the average time required to perform a single backup, *cf.* Section 4.4.2 – Average Backup Time for Increasing Horizon.

3. To assess the empirical interest of our findings with respect to the state-of-the-art approach to solve general `cp-POSG`s, *cf.* Section 4.4.3, we compare anytime performances of `hPBVI` against state-of-the-art solvers – Against State-Of-The-Art Solvers.

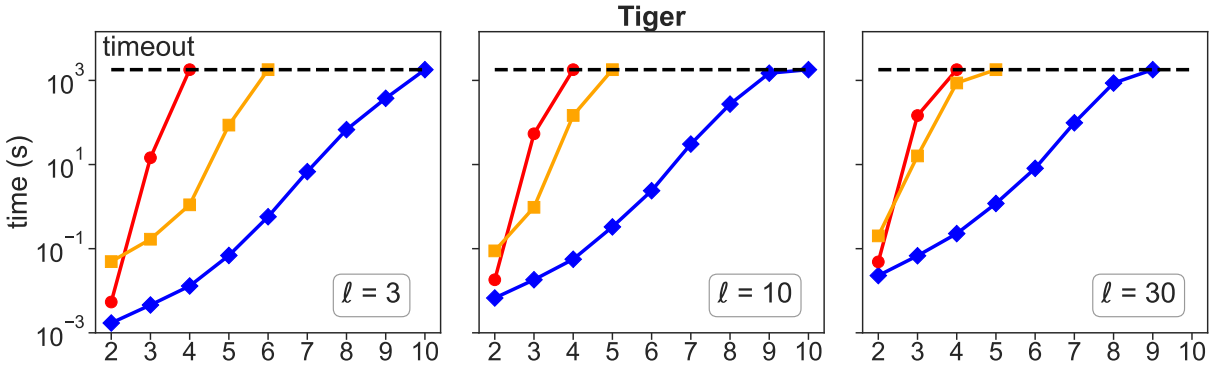
The average time of backups is computed by:

- performing one complete iteration of `PBVI` with only one occupancy state per time step; and
- considering that (especially for this specific iteration) expansion costs are negligible.

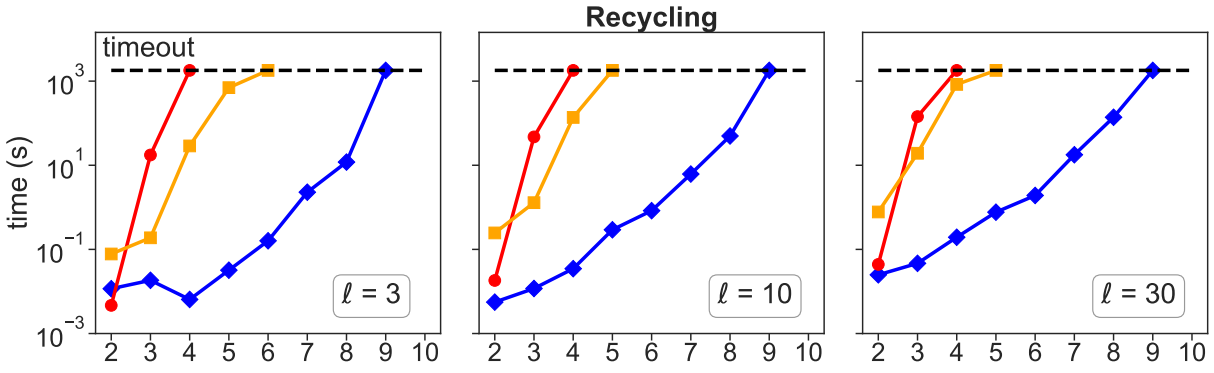
A timeout is set to 1800 seconds for all problems, except Tiger, for which algorithms are given 1 hour.

4.4.1 Average Backup Time for Increasingly Many Players

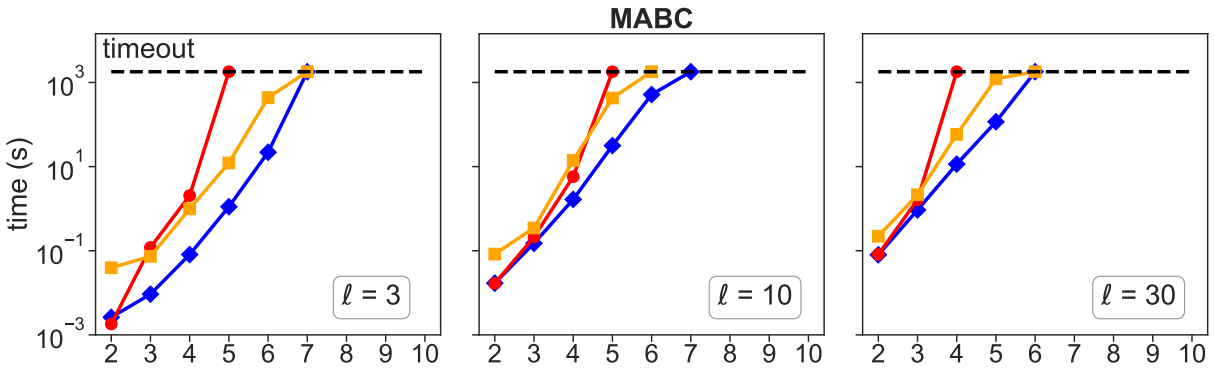
This section investigates the average computation time required to perform a single backup for increasing number of players, *cf.* Figures 4.2a to 4.2d The experiments show that, on all tested benchmarks, `hPBVI` exhibits a reduction in computation time compared to the other variants. Moreover, `hPBVI` can handle a larger number of agents (up to 9 for Tiger, and Recycling) compared to the other variants, which are limited to a maximum of 5 agents. This time-complexity reduction in `hPBVI` is the result of our findings providing the ability to fully exploit the hierarchical information-sharing structure.



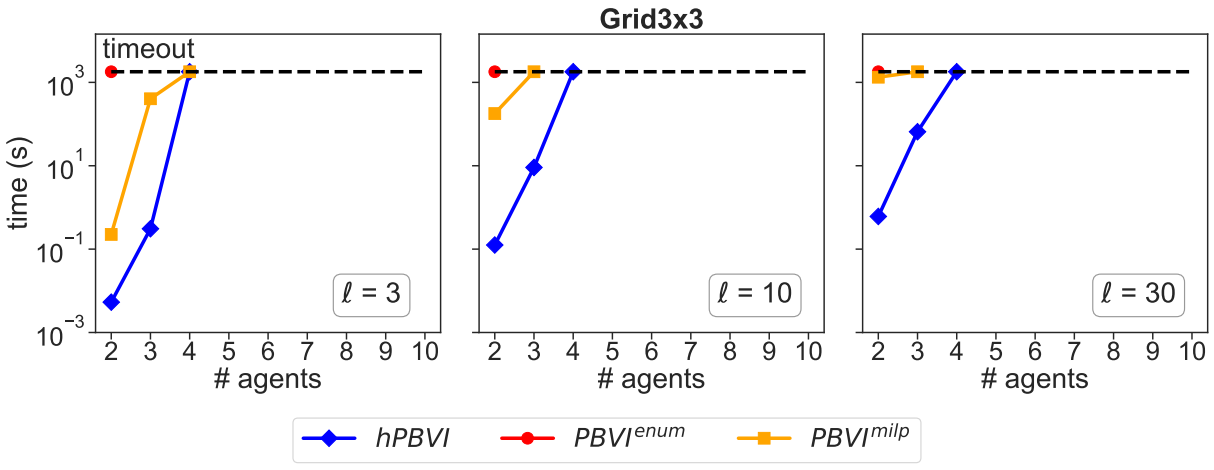
(a) ABT for the tiger problem and different numbers of players.



(b) ABT for the recycling problem and different numbers of players.



(c) ABT for the mabc problem and different numbers of players.



(d) ABT for the grid3x3 problem and different numbers of players.

4.4.2 Average Backup Time for Increasing Horizons

This section studies similar data to the previous one, investigating the average computational time required to perform a single backup for increasing horizons, *cf.* Figures 4.3 to 4.6. The

experiments show once again that, on all tested benchmarks, `hPBVI` exhibits an exponential drop in time complexity compared to the other variants. However, all three variants of the `PBVI` algorithm exhibit an increase in time complexity with respect to the planning horizon. This increase in time complexity is expected since, as time goes, the size of collections $(\mathcal{V}_\tau)_\tau$ also increases.

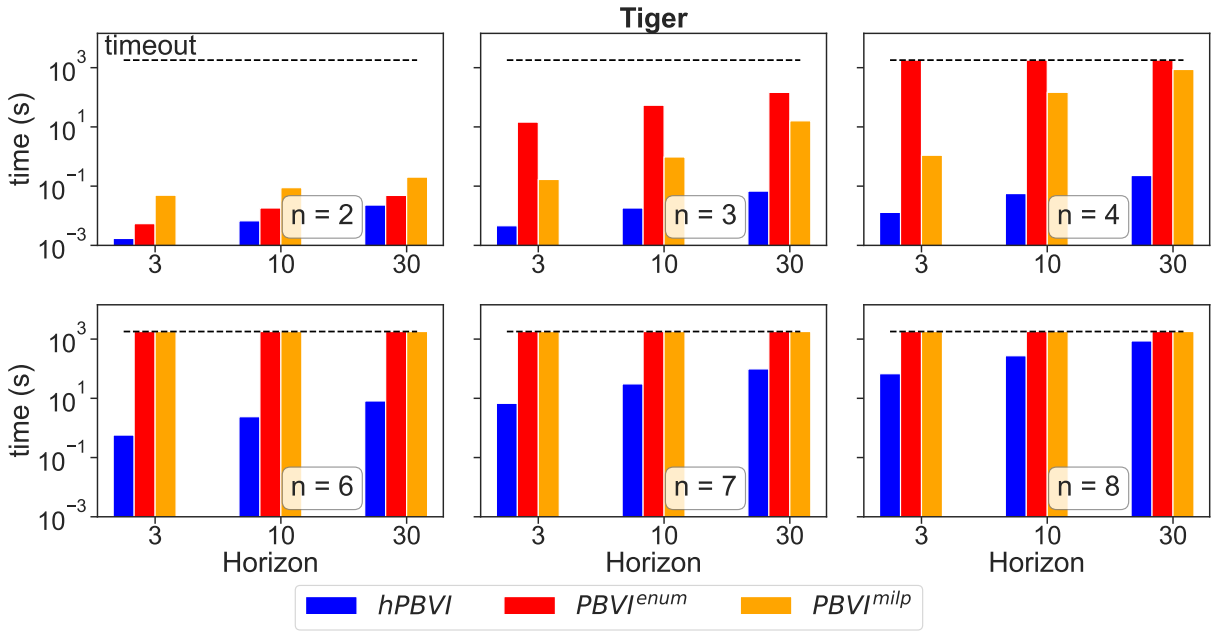


Figure 4.3: Average backup time as a function of planning horizons for Tiger.

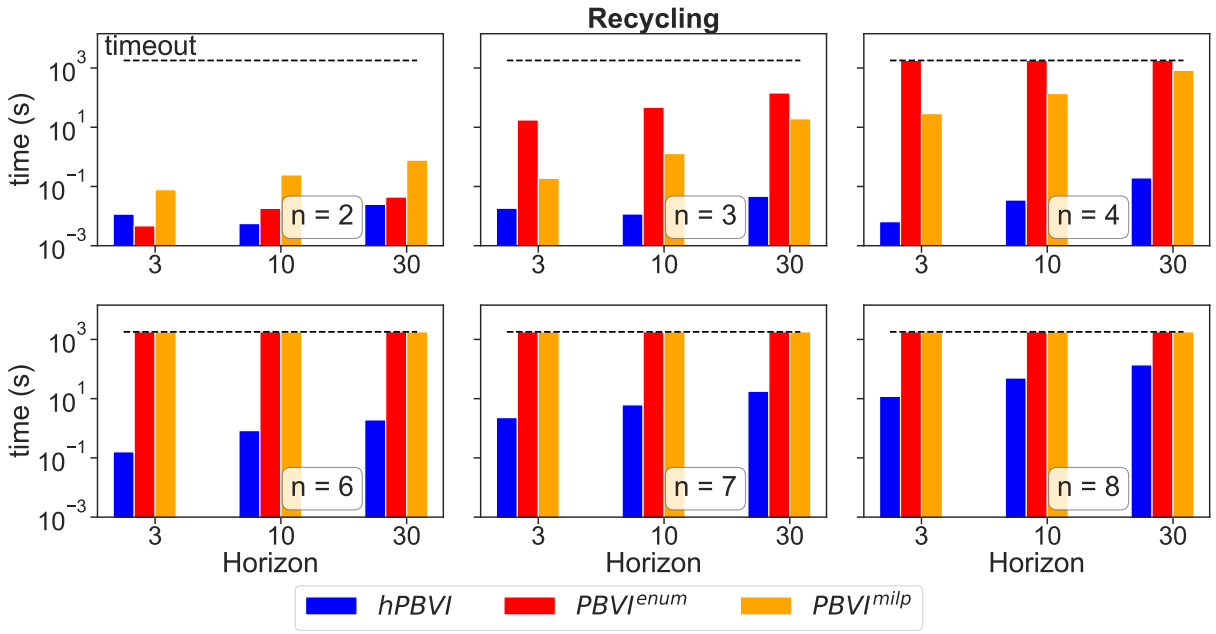


Figure 4.4: Average backup time as a function of planning horizons for Recycling.

4.4.3 Against State-Of-The-Art Solvers

In this section, we compare our `PBVI` algorithm variants with two local algorithms, namely `A2C` and `IQL`, which are state-of-the-art and can handle a large number of players, as shown in Figures 4.7 to 4.10. However, these algorithms prioritize scalability over optimality and may get stuck in local optima. Our experiments demonstrate that `hPBVI` consistently outperforms all competitors in nearly all tested benchmarks in terms of convergence time and the value of the

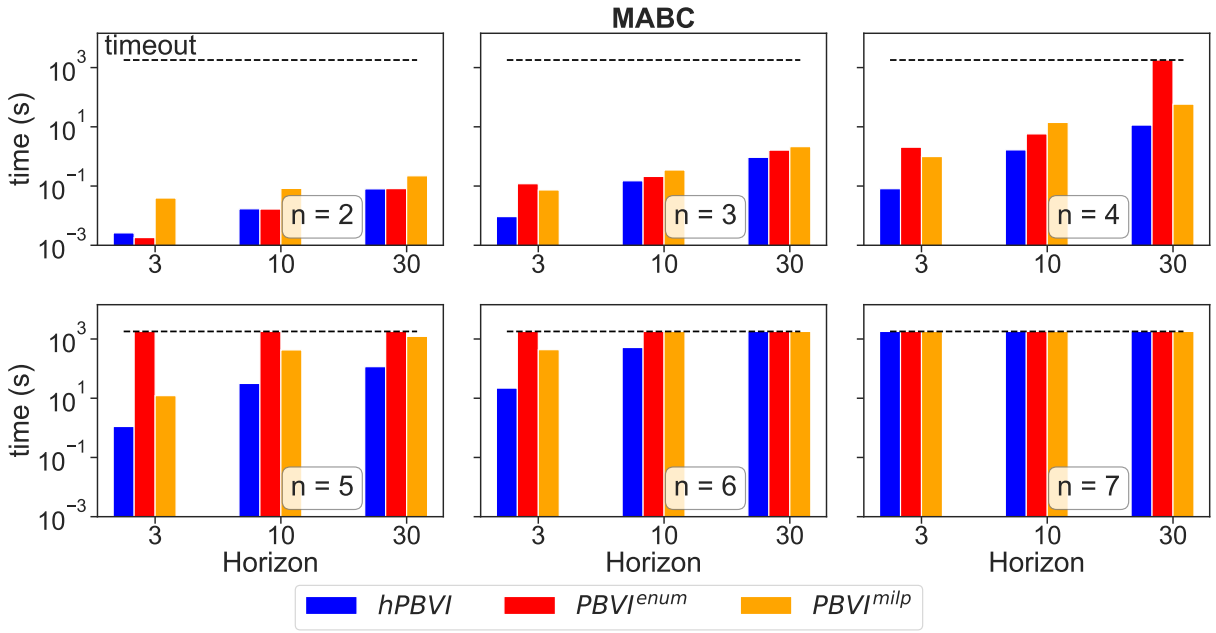


Figure 4.5: Average backup time as a function of planning horizons for MABC.

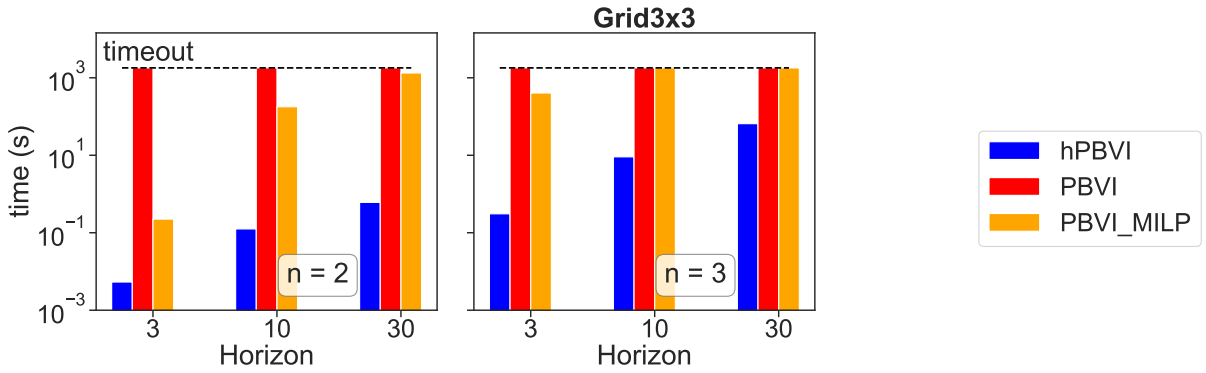


Figure 4.6: Average backup time as a function of planning horizons for Grid3x3.

solution found within an hour. In some weakly-coupled domains, A2C and IQL find near-optimal solutions close to those found by hPBVI.

Our study aimed to assess the reduction in complexity achieved by point-based backups and its effect on solving larger multi-player games. Our findings show that hPBVI performs point-based backups significantly faster than other methods, which enables it to scale up to larger teams, as illustrated in Table 4.1. Specifically, hPBVI was able to perform point-based backups for up to 8 players in about 138.28 seconds in recycling(8) at $H = 30$, while PBVI^{enum} ran out of time for 4 players, and PBVI^{milp} for 6 players. Additionally, hPBVI converges faster than PBVI^{enum} and PBVI^{milp} in 2- to 3-player domains. For example, hPBVI can converge in under 1 second in grid3x3(2) at $H = 30$, while PBVI^{milp} takes about 1329.33 seconds, not to mention PBVI^{enum}. Our results in Table 4.1 demonstrate that hPBVI can scale up to larger teams of players where neither PBVI^{milp} nor PBVI^{enum} can.

Local methods A2C and IQL do scale up to larger teams as expected. Surprisingly, they perform very well on certain domains with weakly coupled players, as shown in mabc(6), grid3x3(3) and tiger(6), *cf.* Table 4.1, for which either A2C or IQL outperforms hPBVI. Figures 4.7 to 4.10 report anytime performances in Section 4.4. Although this observation goes beyond our original goal, it provides encouraging insights when comparing local against global methods over teams of medium sizes. Nonetheless, we caution readers against drawing general conclusions from this observation, as different local methods may yield different local optima and convergence rates.

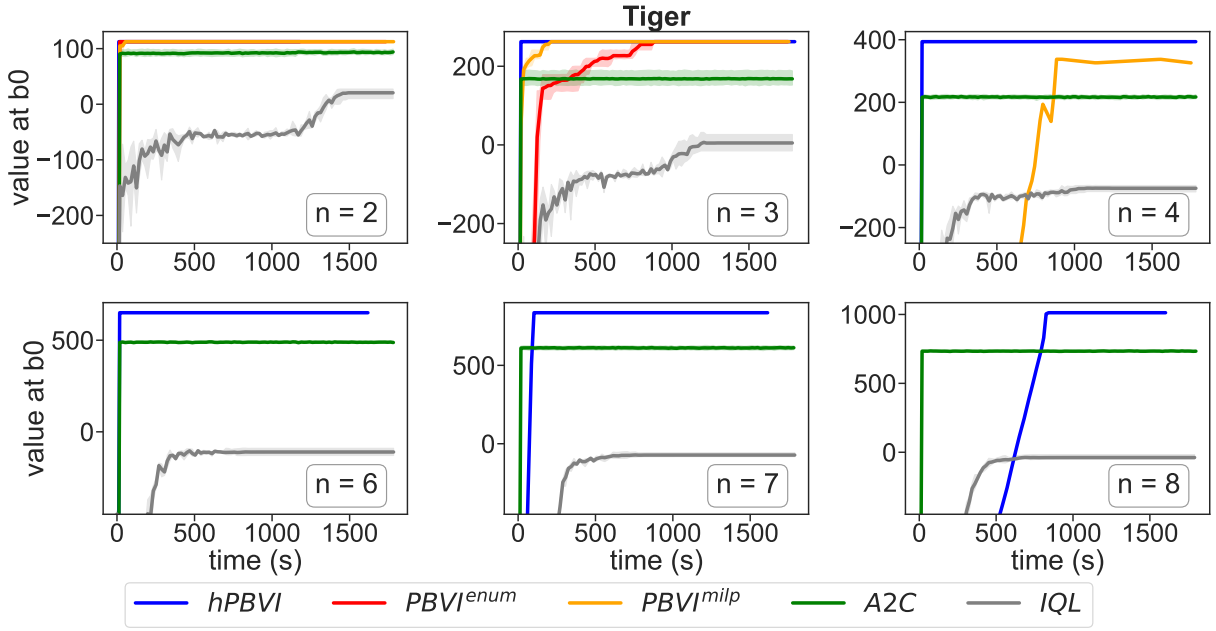


Figure 4.7: Anytime values for Tiger and $H = 30$.

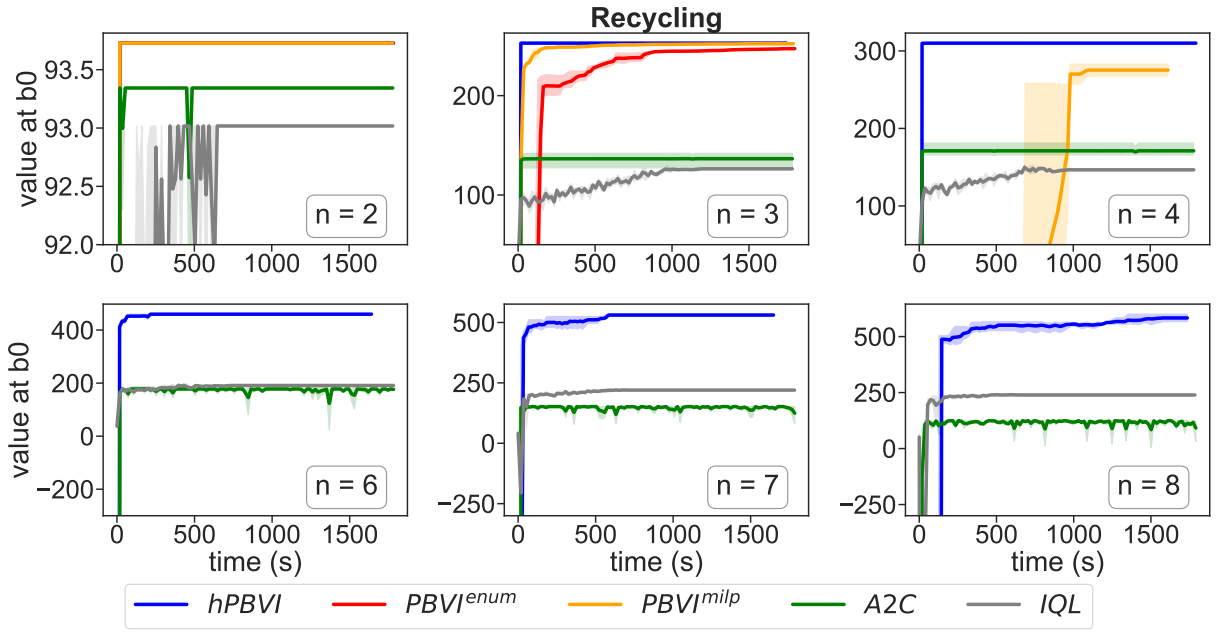


Figure 4.8: Anytime values for Recycling and $H = 30$.

4.5 Conclusion

This chapter presented a point-based value iteration algorithm for near-optimally solving `cp-POSG`s. We exploited a hierarchical information-sharing structure, a dominant management style in our society for corporations, governments, criminal enterprises, armies. Under this assumption, we showed that point-based backup operations can be solved as perfect-information extensive-form games without compromising optimality. Doing so results in an exponential complexity drop, allowing global methods to scale up to larger teams of players. In contrast, the state-of-the-art global approaches quickly ran out of time. Another important empirical finding is that our approach scales to all medium-sized tested domains while providing equal or better performances than state-of-the-art local methods.

Traditionally, global methods have been considered ineffective in games that involve medium to large-sized teams of players. For instance, state-of-the-art `cp-POSG` solvers such as `FB-HSVI` were only designed for two players (Oliehoek et al.; Dibangoye et al.; Dibangoye et al.; Dibangoye

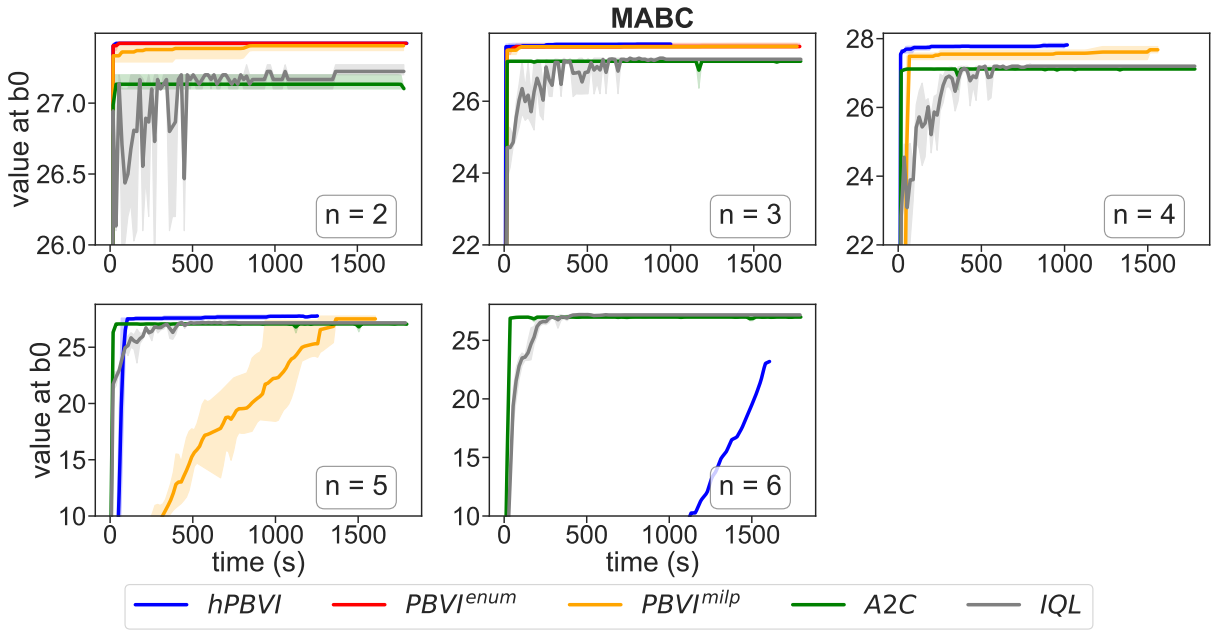


Figure 4.9: Anytime values for Multi-agent broadcast channel and $H = 30$.

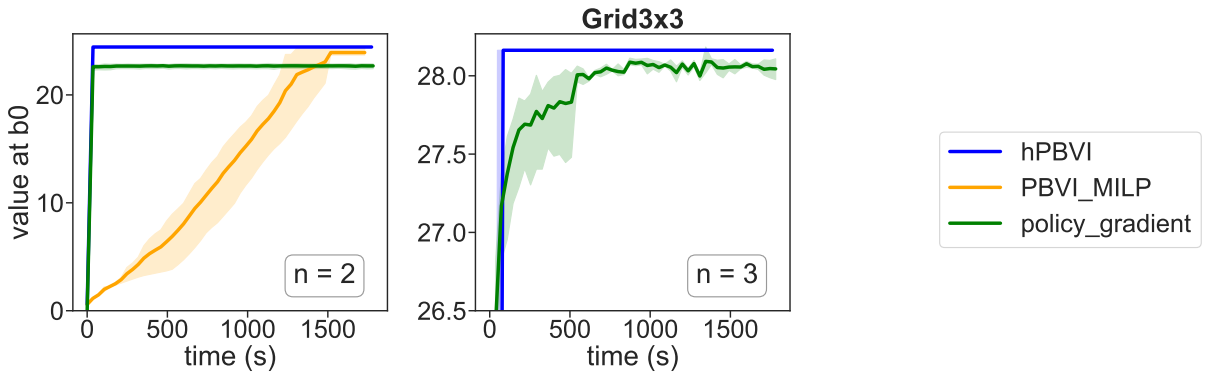


Figure 4.10: Anytime values for Grid3x3 and $H = 30$.

et al., 2010; 2009; 2013; 2016). However, we have presented a contribution that puts forth several propositions for developing global methods that possess the scalability of local methods while maintaining global guarantees. In applications where the stakes are high and critical, such as search and rescue, security, and healthcare, scalable global methods with more reliable solutions than those from local methods are essential.

Next, we discuss two lines of future work, one enabling to take further advantage of specific structure in problems through compression, while the other involves the study of more general hierarchical information-sharing structures compared to the linear one we considered so far.

4.6 Future Work

4.6.1 Compression

An important tool that helps planning algorithms such as `HSVI` or `PBVI` to scale up in general `cp-POSG`s is the ability to compress equivalent individual histories maintained in the support of occupancy states (Dibangoye et al. 2016). Still, players' individual histories in `his-cp-POSG`s carry different information and are actually richer than in `cp-POSG`s. It is not absolutely clear how this affects existing compression techniques (*e.g.,* `LPE` or `TPE` (Dibangoye et al. 2016)). We believe that an analysis of the perfect-information extensive form game introduced for sequential Bellman backup operators would exhibit equivalent nodes, in the sense that computations made in their whole subgames are the same. Even more importantly, since nodes are associated to individual histories, one might be able to compress the occupancy state associated to the game

Table 4.1: Snapshot of empirical results, *cf.* Section 4.4. For each game(n) and algorithm, we report average time (in seconds) per backup and the best value for horizon $H = 30$. oot means time limit of 30 minutes (except for Tiger, for which 1h was given to all algorithms) has been exceeded and '–' is not applicable.

	hPBVI		PBVI^{milp}		PBVI^{enum}		A2C		IQL	
	ABT (s)	$V(b_0)$	ABT (s)	$V(b_0)$	ABT (s)	$V(b_0)$		$V(b_0)$		$V(b_0)$
tiger(2)	**0.18**	**103.70**	1.63	91.80	OOT		–	95.73	–	80.15
tiger(3)	**1.05**	**262.50**	141.72	218.81	OOT		–	167.15	–	255.99
tiger(4)	**6.28**	**393.75**	OOT		OOT		–	207.70	–	218.96
tiger(6)	**912.63**	65.61	OOT		OOT		–	**201.02**	–	-129.51
recycling(2)	**0.02**	**93.73**	0.77	**93.73**	0.05	**93.73**	–	93.34	–	93.01
recycling(3)	**0.047**	**252.83**	19.28	**252.83**	143.59	247.80	–	142.00	–	129.57
recycling(4)	**0.19**	**310.07**	1157.31	283.05	OOT		–	181.25	–	153.03
recycling(6)	**1.91**	**459.78**	OOT		OOT		–	186.11	–	197.93
recycling(8)	**138.28**	**600.00**	OOT		OOT		–	126.19	–	244.02
mabc(2)	**0.08**	**27.42**	0.22	**27.42**	0.08	**27.42**	–	27.20	–	27.27
mabc(3)	**0.93**	**27.61**	2.14	**27.61**	1.63	**27.61**	–	27.12	–	27.20
mabc(4)	**11.44**	**27.87**	58.15	27.77	OOT		–	27.12	–	27.27
mabc(6)	1800	23.19	OOT		OOT		–	27.03	–	**27.21**
grid3x3(2)	**0.61**	**24.44**	1329.33	24.33	OOT		–	22.93	–	24.35
grid3x3(3)	**65.43**	**28.16**	OOT		OOT		–	27.92	–	**28.16**

tree and consequently reduce its dimensionality.

The game tree analysis would start at player n at the top of the hierarchy. For any of her nodes (corresponding to the leaves of the game tree), she knows a probability distribution over the hidden states of the game as she is aware of all her subordinates' individual histories. Then, leaves that induce same probability distribution shall be equivalent, in the sense that any best response of player n to any of her subordinates' actions in one of the node is also a best response for the other equivalent node. Recursively, player i can merge two nodes if (up to a specific ordering over nodes' children) (i) outgoing edges have same probability and (ii) each child of a node is equivalent to a child of the other node, in the sense of $i-1$'s analysis of equivalent nodes.

4.6.2 Hierarchical Organizations

It appears rather intuitive to imagine other hierarchical information-sharing organizations in lieu of the linear hierarchy we exhibited. There could be more than one player at each level of the hierarchy, each knowing what a subset of all players in the subordinate levels of the hierarchy knows. We call *tree-shaped information-sharing* such assumption. Still, it does not appear obvious to determine whether there is anything smarter to do than simply considering a centralized selection of decision rule profiles for each level of the hierarchy, which again entangles players' decision variables for each level.

Let us, therefore, allow ourselves to leverage additional structure in a tree-shaped hierarchical information-sharing cp-POSG. One possible path to follow could be to assume that evaluations of players' behaviors only depend on the behavior of their superiors, the strategies of their subordinates being fixed. In other words, evaluations of behaviors can be independently performed within siblings. Significant structure in the cp-POSG's dynamics must exist to ensure such independence. In fact, the state, observation and reward variables of a player must not be affected by her siblings choice of actions. At high-level, the structure just described resembles the network-distributed one and it might be possible to transfer some of the results from the ND-POMDP literature to our case.

Finally, a very interesting question lies in the possibility to transfer our findings to the zero-sum case. Doing so would generalize Horák et al.'s (2017) work by only assuming that the most-informed player knows her opponent's history, but not the true state of the game. It is, however, an open question to determine whether the complexity of Bellman's operators can be reduced as in the common-payoff setting.

5

min-max Optimization in Non-Linear-Payoff Zero-Sum Games

Contents

This chapter focuses on zero-sum games with only two players, that only involve one time step, and for which von Neumann's minimax theorem typically does not apply.

5.1 Introduction

There is a growing interest in min - max problems from various communities. Generative Adversarial Networks (Sanjabi et al. 2018; Oliehoek et al. 2017), fair statistical inference (Sattigeri et al. 2018; Jagielski et al. 2019), robust decision-making (Chow et al. 2015), and general resource allocation (Du et al. 2017) witness its importance for machine learning. While it has been known for a long time that formulating real-life problems as games is relevant to economics (Friedman 1998), physical phenomena (Brunner et al. 2013) are also interestingly linked to such an optimization problem. Basically, any situation in which two entities are trying to optimize opposite performance criteria by interacting with a common system might be seen through a game-theoretic perspective.

The players' actions having heterogeneous influences on the system, min - max optimization is a problem whose hypotheses and, thus, difficulty substantially vary depending on the context.

As emphasized in the literature (Daskalakis 2022), applications of game theory to deep learning require modeling more and more complex interactions, making the usual convex-concavity or even differentiability assumptions unfortunately too restrictive. This chapter thus provides a first answer to tackle such problems, only requiring mild α-Hölder continuity properties.

Moving on to a formal problem definition, for any given continuous function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ (where $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{Y} \subset \mathbb{R}^p$ are compact), the general min - max problem is to compute at least one pair $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ such that

$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}} \max_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}), \tag{5.1}$$

$$\boldsymbol{y}^* \in \arg\max_{\boldsymbol{y}} f(\boldsymbol{x}^*, \boldsymbol{y}). \tag{5.2}$$

The simplest case, introduced by von Neumann (1928), assumes that (i) $\mathcal{X}$ and $\mathcal{Y}$ are respectively the unit simplices of $\mathbb{R}^n$ and $\mathbb{R}^p$, and (ii) f is bilinear. Then, min - max equals max - min, the problem corresponds to finding a Nash Equilibrium of a 2-player 0-sum normal-form game, and is solved by a simple linear program in polynomial time (Shoham et al. 2008). While min - max and max - min are still equal when replacing (i) and (ii) by (iii) $\mathcal{X}$ and $\mathcal{Y}$ convex and (iv) f convex-concave (Sion 1958), such equality and simplicity of resolution does not hold in the general case.

Relaxing assumptions (iii) and (iv) on f leads to various concerns. If f is either not convex or not concave, Nash Equilibria do not necessarily exist (Daskalakis 2022). Even if they do, the computational cost of finding one might be very high, as deciding whether a Nash Equilibrium exists when the game is not convex-concave is `NP-Hard` in general (Daskalakis et al. 2021).

Relaxing assumptions leads to instabilities and divergent behaviors of gradient-based approaches. To tackle this problem, we build on a global optimizing algorithm for functions with weak continuity assumptions, namely deterministic optimistic optimization (`DOO`) (Munos 2011), to ϵ-optimally solve the $\min - \max$ problem in finite time for functions with only mild α-Hölder continuity properties. This allows to deal with a larger class of games, starting with the ones including entropy measures in the payoff function (Daskalakis 2022, Figure 1-b) (Asarin et al. 2015; Brandsen et al. 2022).

After detailing related work in Section 5.2, and providing some technical background regarding game theory and `DOO` in Section 5.3, Section 5.4 introduces two adaptations of `DOO` in order to provide a finite-time convergent algorithm whenever the variables live in simplices. Section 5.5 then derives an algorithm to ϵ-optimally solve the min-max problem, and this algorithm is experimentally studied in Section 5.6.

5.2 Related work

Assuming only differentiability (but not that a Nash equilibrium necessarily exists), the most common approach to solve the min - max problem is to alternate between gradient ascents and descents to respectively comply with max and min's will. The resulting algorithm is known as *Gradient Descent Ascent (`GDA`)*. However, this can fail (Goktas et al. 2022), even for some simple zero-sum bilinear games (Mescheder et al. 2018).

For this reason, under continuous differentiability assumption, a first-order Nash equilibrium (`FNE`) solution concept was defined through the first-order Taylor approximation of the function, and the existence of at least one `FNE` is guaranteed for twice-differentiable functions (Nouiehed et al. 2019). Recent work introduced modifications to the `GDA` algorithm (Sanjabi et al. 2018; Nouiehed et al. 2019; Goktas et al. 2022) to provide algorithms asymptotically converging towards an `FNE` in a number of gradient evaluations going from $\mathcal{O}(\epsilon^{-2})$ to $\mathcal{O}(\epsilon^{-6})$ (Rafique et al. 2018) through $\mathcal{O}(\epsilon^{-4})$ (Jin et al. 2019; Lin et al. 2020), depending on the assumptions made on f. As a matter of fact, computing ϵ-FNEs has been shown to be in ExpTime with respect to either $1/\epsilon$, the smoothness of the game, or its dimensionality (Daskalakis et al. 2021).

The intensive study of gradient-based approaches is partly due to its utility for deep learning. Thus, min - max problems without, at least, differentiability assumptions were not studied even though they naturally appear (Daskalakis 2022, Figure 1-b) (Asarin et al. 2015; Brandsen et

al. 2022). Since we are interested in games with poor continuity properties, we focus on the min - max solution concept, which is well-defined and exists.

Encouragingly, the min - max problem always admits a solution (by f's continuity) and the optima of a "local" min - max version of the problem (Jin et al. 2019), *i.e.,* where players can only deviate within a ball of radius δ from their strategy, are strongly linked to the stable limit point of `GDA` (Jin et al. 2019). Besides, the $\min - \max$ computation makes sense as it corresponds to finding *security levels* for player min by searching for the most rewarding strategy that she can announce to player max, and the resulting problem is called a *zero-sum Stackelberg competition.*

Buşoniu et al. (2014) consider a slightly more general case in which players perform sequences of actions. They show that, from this setting, one can derive a min - max algorithm to solve the min - max optimization problem for Lipschitz-continuous functions w.r.t. a semi norm l, by iteratively constructing a tree representation of possible sequences of actions for players 1 and 2. Interestingly, min-max optimization problems involving only one time step can be tackled through seeing the game as a sequential one, in which players' decisions (min playing before max) consist in making a dichotomy of their search space. While we share similar ideas, our contribution follows a different line of research, focusing on game-theoretical applications, thus providing the adaptation for simplex strategy spaces, games with dependent feasible sets, and drawing connections with recent questions in the game theory literature. We furthermore expect that, while different, both algorithmic schemes yield complementary levers. For example, we leverage the α-Hölder properties to guide search in a subdivision tree. On the other hand, pruning criteria in α-β-like algorithms might be more powerful than the pruning criterion we evoked in Remark 5.5.2.

5.3 Background

First, we start by providing some necessary background about game theory and insights about the relevance of the α-Hölder condition in games.

5.3.1 Games and solution concepts

Definition 5.3.1 (Two-player Zero-Sum Game)**.** *In this whole chapter, by default, games considered are zero-sum two-player one-shot games $\langle 2, \mathcal{A}^1, \mathcal{A}^2, f, -f\rangle$ (as defined in Definition 2.1.1, page 12), in which f is continuous but not necessarily differentiable. We recall that the game is in normal form*[33] *if (i) $\mathcal{A}^1$ and $\mathcal{A}^2$ are respectively the unit simplex S_{p_1} of $\mathbb{R}^{p_1}$ and the unit simplex S_{p_2} of $\mathbb{R}^{p_2}$, and (ii) $f : x, y \mapsto \boldsymbol{x}^\top \cdot M \cdot \boldsymbol{y}$, is derived from a matrix M belonging to $\mathcal{M}_{p_1,p_2}$.*

A simple, yet interesting, example consists in a zero-sum normal-form game $\langle S_2, S_2, M\rangle$ in which player 1 (resp. 2) also wants to minimize (resp. maximize) the entropy of her mixed strategy, while not degrading too much their expected payoff.

Example 5.3.2 ((Daskalakis 2022))**.** *Consider the classical matching pennies game given by the payoff matrix M*

$$M = \begin{pmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \end{pmatrix}, \tag{5.3}$$

and the bilinear payoff function $\boldsymbol{x}^\top \cdot M \cdot \boldsymbol{y}$. The payoff function of the modified matching pennies game in which player 1 is rewarded for a high entropy strategy but 2 is penalized for a high entropy strategy is formally given by

$$f(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top \cdot M \cdot \boldsymbol{y} + x_1 \log(x_1) + x_2 \log(x_2) + y_1 \log(y_1) + y_2 \log(y_2).$$

In this zero-sum game, no Nash equilibrium exists, and the payoff function is neither convex nor concave. Still, the payoff function is α-Hölder for any $\alpha \in]0, 1[$.

[33] Actually, the games correspond to normal-form games extended to mixed strategies, as in in Theorem 2.1.5.

Definition 5.3.3 (α-Hölder condition)**.** *For any norm $\|\cdot\|$, a function $f : \mathcal{X} \to \mathbb{R}$ satisfies the α-Hölder condition with respect to $\|\cdot\|$ if and only if there exists $\alpha \in]0, 1]$ and $C \in \mathbb{R}^+$ such that*

$$\forall (\boldsymbol{x}, \tilde{\boldsymbol{x}}) \in \mathcal{X}^2, \ |f(\boldsymbol{x}) - f(\tilde{\boldsymbol{x}})| \leqslant C\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^\alpha. \tag{5.4}$$

As mentioned in the introduction, Nash Equilibrium points (*i.e.,* joint strategies which no player has incentive in unilaterally deviating from) do not necessarily exist, in which case max - min > min - max, even in a local sense (Jin et al. 2019). Players thus can not agree on a joint strategy, and might want to search for the individual strategy that has the greatest value against any strategy of their opponent, *i.e.,* (for player 1) compute $\arg\min_{\boldsymbol{x}} \max_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y})$.

Since we tackle games with poor continuity properties of the payoff function, we only assume that f is α-Hölder[34].

5.3.2 Deterministic Optimimistic Optimization (DOO)

We below present the DOO algorithm introduced by Munos (2011) to tackle single-variable optimization problems for functions with mild Lipschitz properties.

Let $f : \mathcal{X} \to \mathbb{R}$ be a λ-Lipschitz function with respect to any semi-metric l (*i.e.,* $\forall (\boldsymbol{x}, \boldsymbol{x}') \in \mathcal{X}^2, |f(\boldsymbol{x}) - f(\boldsymbol{x}')| \leqslant \lambda l(\boldsymbol{x}, \boldsymbol{x}')$) defined over $\mathcal{X} \subset \mathbb{R}^n$. Given any semi-metric l, a cell (compact subset) $\mathcal{S}$ and point $\tilde{\boldsymbol{x}}$ in $\mathcal{S}$, f's Lipschitz continuity with respect to l allows optimistically bounding, *i.e.,* lower bounding, its value within $\mathcal{S}$ by $f(\tilde{\boldsymbol{x}}) - \lambda Diam(\mathcal{S})$, where $Diam(\mathcal{S}) = \max_{\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{S}} l(\boldsymbol{x}, \boldsymbol{x}')$. The smaller the diameter of the cell, the closer the optimistic bound is to the values of f in it.

If $\mathcal{X}$ is bounded, DOO ϵ-optimally solves $\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})$ for a given error $\epsilon > 0$ by creating a non-uniform covering of the domain with a finite number of cells where the variations of the function are controlled. Let us assume that $\mathcal{X}$ is such that (i) there is an analytic way to create a first covering with cells, and (ii) each cell can again be covered with children cells. The algorithm (given in Algorithm 5.1) starts with the first covering of $\mathcal{X}$ (line 2), and computes the optimistic bound in every cell (line 3). Then, each iteration consists in

1. selecting the most promising cell S_{i*} according to the optimistic bound (line 6),
2. covering it (line 7) with subsets $(S_j)_{j \in \mathcal{I}*}$,
3. finding a representative element $\boldsymbol{x}_j$ of any cell S_j (line 8)
4. computing the optimistic bound of each new cell (line 9).

The final returned value is the encountered point with the lowest value. Figure 5.1 gives an illustration of an execution of DOO.

If the cells are "well-formed" (following Assumption 5.3.4 below) and their diameter decreases w.r.t their depth in the tree (following Assumption 5.3.5), the approximation error shrinks with the radius around the most promising area as the algorithm iterates. Thus, the difference $r(n)$ between the smallest value of f and DOO's returned value after performing n coverings of cells with children cells can be bounded (Munos 2011).

Assumption 5.3.4 (Valid cell (Assumption 4 in (Munos 2011)))**.** *There exists $\nu \in \mathbb{R}^{+,*}$ such that, for any cell $\mathcal{S} \subset \mathcal{X}$, there exists $\boldsymbol{x} \in \mathcal{S}$ such that $\mathcal{B}_l(\boldsymbol{x}, \nu Diam(\mathcal{S})) \subset \mathcal{S}$, where $\mathcal{B}_l(\boldsymbol{x}, \rho) = \{y \in \mathcal{X} | l(\boldsymbol{x}, \boldsymbol{y}) \leqslant \rho\}$ denotes the ball of radius ρ centered in $\boldsymbol{x}$ for the semi-metric l.*

Assumption 5.3.5. *There exists a decreasing sequence $\delta(h)$ such that, for any depth $h \geqslant 0$, for any cell $\mathcal{S}_{h,i}$ of depth h in the tree, $Diam(\mathcal{S}_{h,i}) \leqslant \delta(h)$.*

Remark 5.3.6. *Fortunately, for any given norm $\|\cdot\|$, (i) an α-Hölder function with constant C is exactly a C-Lipschitz function with respect to the semi-metric $\|\cdot\|^\alpha$, and (ii) balls $\mathcal{B}_{\|\cdot\|^\alpha}(\boldsymbol{x}, \rho)$ are valid cells. We are now able to adapt Munos' work to tackle* min - max *problems for α-Hölder functions, using simple balls $\mathcal{B}_{\|\cdot\|^\alpha}(\cdot, \cdot)$.*

The next two sections respectively show how DOO can be (i) applied to the special case of optimization over a simplex and (ii) adapted to solve a min - max problem.

[34] For the sake of simplicity, we only expose the results for α-Hölder functions, even though Munos (Munos 2011) presents the result for Lipschitz functions with respect to any semi metric l (not just $\|\cdot\|^\alpha$).

Algorithm 5.1: DOO

Fct DOO$\Big(\big[\mathcal{X} \to \mathbb{R}\,;\, x \mapsto f(x)\big], n\Big)$
  input : $f : \mathbb{R} \to \mathbb{R}$ α-Hölder func. with constant C
  Initialize $\mathcal{I}$ and $(\mathcal{S}_i)_{i\in\mathcal{I}}$ s.t. $\mathcal{X} \subseteq \cup_{i\in\mathcal{I}}\mathcal{S}_i$
  $\forall i \in \mathcal{I}$, compute $f(\boldsymbol{x_i}) - C\rho_i^\alpha$
  /* ρ_i : diameter of $\mathcal{S}_i$;
     x_i : representative element of $\mathcal{S}_i$. */
  $time \leftarrow 0$
  while $time \leqslant n$ **do**
    $i^* \leftarrow \arg\min_{i\in\mathcal{I}} f(\boldsymbol{x_i}) - C\rho_i^\alpha$
    Cover $\mathcal{S}_{i^*}$ by $\cup_{j\in\mathcal{I}^*}\mathcal{S}_j \quad (\supseteq \mathcal{S}_{i^*})$
    $\forall j \in \mathcal{I}^*, \boldsymbol{x_j} \leftarrow Repr(\mathcal{S}_j)$
    /* $Repr(\mathcal{S}_j)$: point $\in \mathcal{S}_j \cap \mathcal{X}$. */
    $\mathcal{I} \leftarrow [\mathcal{I}\backslash\{i^*\}] \cup \mathcal{I}^*$
    $time \leftarrow time + 1$
  return $\langle \arg\min_{x_i} f(x_i) \rangle$

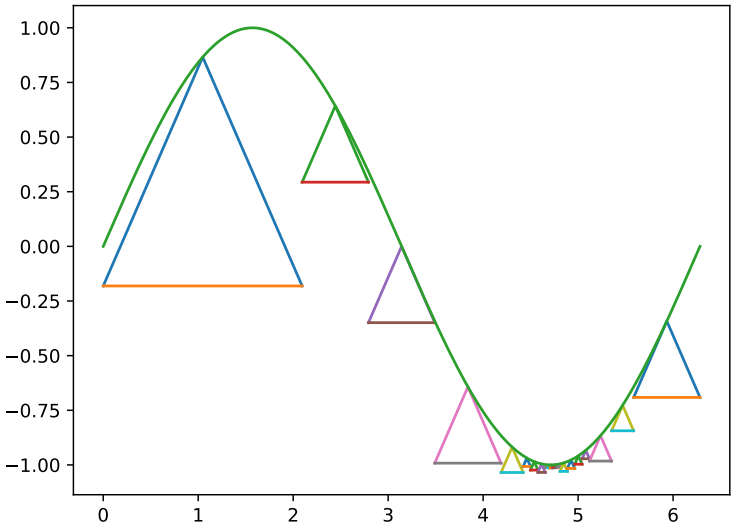


Figure 5.1: Representation of DOO's execution to minimize $x \mapsto \sin(x)$ on $[0, 2\pi]$. The interval is covered with cells (*i.e.*, here, intervals) of different sizes where the function sin is lower-bounded by "Lipschitz cones", themselves lower-bounded by a constant. The cones, along with their constant lower bounds, form the triangular shapes.

5.4 Finite-time convergent DOO for simplex spaces

In this section, we come back to vanilla minimization (rather than min - max optimization) and provide two modifications to the original DOO algorithm in order to match usual requirements of game theory. First, we require our algorithm to converge in finite time to an ϵ-optimum instead of bounding the regret for a given time budget. This leads to introducing a pessimistic bound which will also be useful later as a pruning criterion. Secondly, we show that DOO can be used when the search space is a simplex by applying iterative simplex discretization methods.

5.4.1 Modifying the stopping criterion

The initial analysis of DOO's complexity bounds its error with respect to a given budget n. We change viewpoint to show that, for any $\epsilon > 0$, one can derive a stopping criterion that is reached in finite time. To do so, we introduce upper and lower bounds of the optimal value, which will monotonically shrink as the algorithm iterates. The algorithm returns whenever the difference

between the maximum value of the upper bound and the maximum value of the lower bound is smaller than ϵ.

Proposition 5.4.1 (Upper and lower bounds)**.** *Let f be an α-Hölder function and C be an α-Hölder constant of f. Let $(\mathcal{B}_{\|\cdot\|_\infty}(\boldsymbol{x_i}, \rho_i))_i$ be a set of cells partitioning f's domain. Then, $\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$ is upper- and lower-bounded respectively by $\min_{\boldsymbol{x_i}} f(\boldsymbol{x_i})$ and $\min_{\boldsymbol{x_i}} [f(\boldsymbol{x_i}) - C\rho_i^\alpha]$.*

Proof. The proof is direct. □

A straightforward adaptation of Munos' regret-bounding proof leads to the following convergence theorem.

Theorem 5.4.2 (adapted from (Munos 2011))**.** *Let $\epsilon > 0$ be a given error. Let $(\mathcal{B}_{\|\cdot\|_\infty}(x_i, \rho_i))_i$ be the set of cells covering f's domain, updated as the algorithm iterates. The quantity $|\min_i [f(\boldsymbol{x_i}) - C\rho_i^\alpha] - \min_i [f(\boldsymbol{x_i})]|$ decreases throughout the optimization process and is smaller than ϵ after a finite number of iterations.*

Remark 5.4.3. *One could derive a precise complexity bound, but at the cost of introducing function-dependent topological constants whose computation is, in general, even harder than the optimization problem. Instead, Theorem 5.5.3 (Section 5.5.1) will provide an upper bound of the complexity, based on the worst-case scenario of optimizing a constant function.*

Note that (i) $\min_i f(\boldsymbol{x}_i) - C\rho_i^\alpha$ and $\min_i f(\boldsymbol{x}_i)$ have no reason to be attained in the same cell; and (ii) Theorem 5.4.2 leads to simply modifying line 5 of Algorithm 5.1 to stop whenever the gap between bounds is less than ϵ. The parameter n in Algorithm 5.1 is thus replaced by ϵ.

5.4.2 DOO for simplex spaces

Subdivision process We now detail how to subdivide the n-dimensional unit simplex $S_n(1)$ using smaller and smaller hypercubes as the process iterates.

Even though simplex decomposition techniques are not new (Edelsbrunner et al. 1999; Paulavičius et al. 2014), we introduce a new simple subdivision process that is well suited to the use of DOO. The process starts with the n-dimensional hypercube (*i.e.,* the closed ball $\mathcal{B}_{\|\cdot\|_\infty}(\mathbf{0}, 1) = \{x \in \mathbb{R}^n \,|\, \|\boldsymbol{x} - \mathbf{0}\|_\infty \leqslant 1\}$ centered in $\mathbf{0}$ with a radius 1), and will decompose it through smaller hypercubes. Note that hypercubes are cells that cover $[0, 1]^n$ whereas their intersection with the unit simplex are cells covering the $n-1$-simplex. The following theorem shows how to determine whether a hypercube intersects the simplex or not, and how to compute a *referent* point, on which the function will be evaluated. All the created hypercubes can again be covered with 2^n hypercubes, and so on, which allows creating an iterative covering of the unit simplex with smaller and smaller hypercubes.

Theorem 5.4.4 (Intersection between the probability $n-1$-simplex and an n-dimensional hypercube)**.** *Let $n \in \mathbb{N}^*\backslash\{1\}$. Let H be an n-dimensional cube (*i.e., *a closed ball for $\|\cdot\|_\infty$) of radius η whose center is called m. Then, H and $S_n(1)$ intersect (*i.e., *$(H \cap S_n(1)) \neq \varnothing$) if and only if $\exists(\boldsymbol{x_i}, \boldsymbol{x_j}) \in H^2$ such that $\sum_{k=1}^n x_i^k \leqslant 1$ and $\sum_{k=1}^n x_j^k \geqslant 1$, and an intersection point (called referent point) can be computed.*

Proof. Let us consider the diagonal from the lowest point ($\boldsymbol{x}_{inf} = (m^1 - \eta, \ldots, m^n - \eta)$) to the highest point ($\boldsymbol{x}_{inf} = (m^1 + \eta, \ldots, m^n + \eta)$) of H, and apply the Intermediate Value Theorem. There is no intersection point if and only if $\sum_k \boldsymbol{x}_{inf}^k > 1$ or $\sum_k \boldsymbol{x}_{sup}^k < 1$. Else, the intersection point is $x = \boldsymbol{x}_{inf} + t(\boldsymbol{x}_{sup} - \boldsymbol{x}_{inf})$, where $t = \frac{1-\sum_{k=1}^n \boldsymbol{x}_{inf}^k}{2n\eta}$. Indeed,

$$\sum_{k=1}^n x^k = 1 \;\Leftrightarrow\; \sum_{k=1}^n \left[\boldsymbol{x}_{inf}^k + t(\boldsymbol{x}_{sup}^k - \boldsymbol{x}_{inf}^k)\right] = 1$$

$$\Leftrightarrow\; t = \frac{1 - \sum_{k=1}^n \boldsymbol{x}_{inf}^k}{\sum_k \boldsymbol{x}_{sup}^k - \boldsymbol{x}_{inf}^k} \;\Leftrightarrow\; t = \frac{1 - \sum_{k=1}^n \boldsymbol{x}_{inf}^k}{n \cdot 2\eta}.$$

□

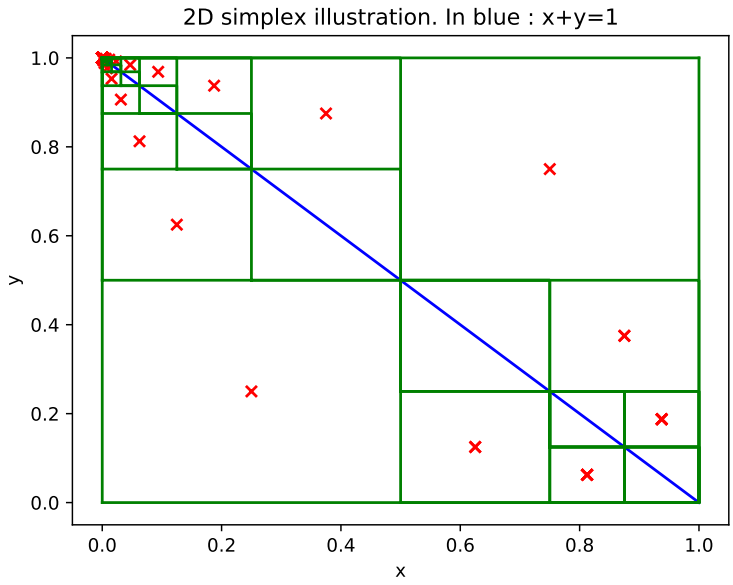


Figure 5.2: Illustration of the subdividing process

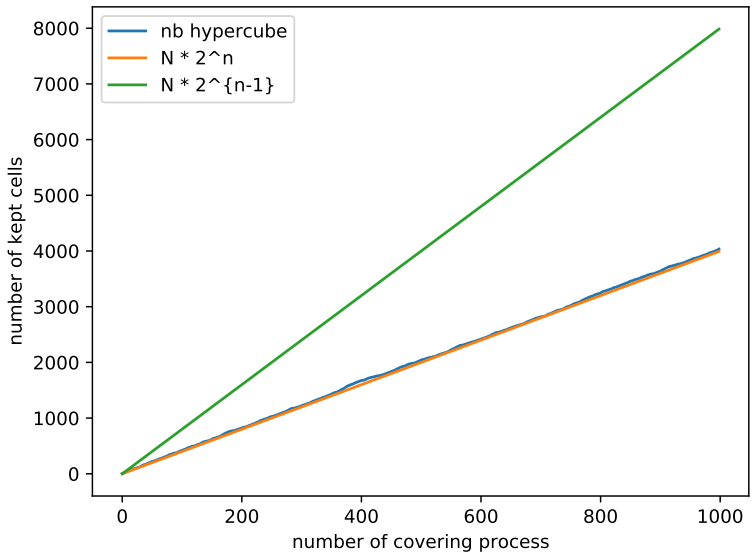


Figure 5.3: Number of hypercubes kept when optimizing over the n-dimensional simplex (here, $n = 3$) as a function of `DOO`'s number of iterations (N).

Figure 5.2 illustrates the iterative subdivision of the 2-dimensional unit simplex by 2-dimensional hypercubes (*i.e.,* squares). Let us point out that the subdivision operation is here concentrated around the optimum: $(0, 1)$. Squares with a red cross do not intersect with the $n-1$-simplex and are pruned, while the other ones do and are therefore kept. The unit-simplex being of dimensionality $n-1$ and the hypercubes of dimensionality n, as illustrated by Figure 5.3, we conjecture that $\lim_{n\mapsto\infty} \#H(N) = N \cdot 2^{n-1}$, where $\#H(N)$ denotes the number of hypercubes kept after N iterations.

5.5 min-max α-Hölder Optimization

In this section, we come back to the min - max problem to provide a global optimization algorithm that converges in finite time to an ϵ-optimal solution.

To do so, one can first notice the α-Hölder continuity of the "outer" function $\boldsymbol{x} \mapsto \max_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y})$.

Lemma 5.5.1. *Let f be a α-Hölder function and let C be one of its α-Hölder constants. The function $\boldsymbol{x} \mapsto \max_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y})$ is α-Hölder and C is one of its α-Hölder constants.*

Proof. Let

$$g(\boldsymbol{x}) \stackrel{\text{def}}{=} \max_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}).$$

Then, for any $\boldsymbol{x}$ and $\boldsymbol{x}'$, assuming wlog that $g(\boldsymbol{x}) > g(\boldsymbol{x}')$,

$$\begin{aligned}
g(\boldsymbol{x}) - g(\boldsymbol{x}') &= \max_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}) - \max_{\boldsymbol{y}} f(\boldsymbol{x}', \boldsymbol{y}) \\
&\leqslant \max_{\boldsymbol{y}} \big(f(\boldsymbol{x}', \boldsymbol{y}) + C\|(\boldsymbol{x}, \boldsymbol{y}) - (\boldsymbol{x}', \boldsymbol{y})\|^\alpha\big) - \max_{\boldsymbol{y}} f(\boldsymbol{x}', \boldsymbol{y}) \\
&= \max_{\boldsymbol{y}} \big(f(\boldsymbol{x}', \boldsymbol{y})\big) + C\|\boldsymbol{x} - \boldsymbol{x}'\|^\alpha - \max_{\boldsymbol{y}} f(\boldsymbol{x}', \boldsymbol{y}) \\
&= C\|\boldsymbol{x} - \boldsymbol{x}'\|^\alpha.
\end{aligned}$$

A similar result holds for $g(\boldsymbol{x}') > g(\boldsymbol{x})$, concluding the proof. □

With this property, we solve our min - max optimization problem by using two nested DOO processes, *i.e.,*

- an outer ϵ_1-optimal DOO minimizing the function $\boldsymbol{x} \mapsto \max_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y})$, using the solution of
- an inner ϵ_2-optimal DOO maximizing the function $\boldsymbol{y} \mapsto f(\boldsymbol{x}, \boldsymbol{y})$ for fixed $\boldsymbol{x}$

(see Algorithm 5.2). Munos' proof can be adapted to this case to show that the final error is $\epsilon = \epsilon_1 + \epsilon_2$.

Algorithm 5.2: BiDOO

```
Fct BiDOO([A^1 × A^2 → ℝ ; x, y ↦ f(x, y)], ε_1, ε_2, C))
    input : f : A^1 × A^2 → ℝ an α-Hölder function
    ⟨x_min, v_min⟩ ← DOO(
        [A^1 → ℝ ; x ↦ −getMin(
            DOO([A^2 → ℝ ; y ↦ −f(x, y)], ε_2, C)
        )], ε_1, C)
    /* getMin(·,·) here returns its second argument, i.e., the minimum of the inner DOO
       computation.                                                              */
return ⟨x_min, v_min⟩
```

Remark 5.5.2 (Pruning the Inner Process)**.** *Using a pessimistic bound for the outer process is not only useful to provide a finite-time convergent algorithm, but also helps to prune some irrelevant parts of the search space. The current pessimistic bound of the outer DOO is passed to the inner DOO when called, so that it stops whenever the value of x is necessarily higher than the pessimistic bound of the inner process (process not shown in Algorithm 5.2).*

5.5.1 Complexity analysis

Let us now upper bound the number of iterations it takes for BiDOO to converge by analyzing it in the worst case, *i.e.,* optimizing a constant function. Let us assume that BiDOO is called to solve $f : S_n \times S_p \mapsto \{a\}$ with $a \in \mathbb{R}$, but is given an (overestimating) α-Hölder constant $C > 0$. In this case, BiDOO iteratively covers up the simplex with an increasingly thin uniform grid, and the exact number of iterations it takes before reaching a stopping criterion ϵ can be found analytically.

Theorem 5.5.3 (Complexity upper-bound)**.** *For any zero-sum game $\langle S_n, S_p, f\rangle$, `BiDOO` returns an ϵ-optimum (x^*, y^*) in less than*

$$\left[\sum_{i=0}^{\log_2(C^\alpha/\epsilon)} (2^n)^i\right] \times \left[\sum_{i=0}^{\log_2(C^\alpha/\epsilon)} (2^p)^i\right]$$

subdivision processes, where C is any α-Hölder constant of f given to `BiDOO`.

Proof. `BiDOO` subdivides the search space. For a given ϵ, and a given overestimating α-Hölder constant C, the minimum diameter of the cells to ensure ϵ-convergence is ϵ/C^α. With hypercubes, the radius of the cells is divided by 2 at each subdividing process, so that a depth of $\log_2(\epsilon/C^\alpha)$ in *DOO*'s tree is required. Now, each evaluation of a node in `BiDOO`'s outer tree has, at worst, the same complexity as a complete 2^p-ary tree of depth $\log_2(\epsilon/C^\alpha)$. The outer tree after convergence being, at worst, a 2^n-ary tree of depth $\log_2(\epsilon/C^\alpha)$, the complexity result holds. The worst case is obtained when f is constant. □

5.5.2 Games with dependent feasible set

In this section, we look at games with dependent feasible sets, formally given by the computation of

$$\min_{x\in\mathcal{A}^1} \max_{y\in F(x)\subset\mathcal{A}^2} f(x,y) \tag{5.5}$$

for some F. Such games received a growing interest recently, especially in the form of Fisher market problems (Fisher 1892), formulated as a game with dependent feasible set in (Goktas et al. 2021). Assuming that f is α-Hölder *in the whole space* $\mathcal{A}^1 \times \mathcal{A}^2$, the following provides a sufficient hypothesis on F to ensure `BiDOO`'s convergence on such a game.

Theorem 5.5.4. *Let H be the Haussdorf distance for compact sets of $\mathcal{B}(\mathcal{A}^2)$, defined by:*

$$\begin{aligned} H : \mathcal{B}(\mathcal{A}^2)^2 &\to \mathbb{R}^+ \\ (A,B) &\mapsto \max\{\sup_{a\in A}\inf_{b\in B}\|a-b\|_\infty, \sup_{b\in B}\inf_{a\in A}\|a-b\|_\infty\}. \end{aligned}$$

*Let $\mathcal{B}(\mathcal{A}^2)$ denote the Borelian algebra of $\mathcal{A}^2$. If, $\forall x \in \mathcal{A}^1$, $F(x)$ is a bounded, closed and convex (*i.e.*, compact convex) element of $\mathcal{B}(\mathcal{A}^2)$ and F is λ-Lipschitz continuous for the Haussdorf distance, then `BiDOO` converges towards a $C\epsilon\lambda \cdot (1 + \frac{\epsilon^\alpha}{\epsilon})$-optimum in finite time.*

Proof. Let $f : \mathcal{A}^1 \times \mathcal{A}^2 \to \mathbb{R}$ be α-Hölder, and C be a Hölder constant of f. Assume that $F : \mathcal{A}^1 \to \mathcal{B}(\mathcal{A}^2)$ is λ-Lipschitz-continuous for the Haussdorf distance. Let $x \in \mathcal{A}^1$ and $\epsilon > 0$. We aim at bounding the variations of $x \mapsto \max_{y\in F(x)} f(x,y)$ within a ball of radius ϵ around x, formally given by $\sup_{\tilde{x}\in B_{\|\cdot\|_\infty}(x,\epsilon)} |\max_{y\in F(x)} f(x,y) - \max_{y\in F(\tilde{x})} f(\tilde{x},y)|$.

A first observation is that, $\forall \tilde{x} \in B_{\|\cdot\|_\infty}(x,\epsilon)$,

$$\forall y \in F(\tilde{x}),\ f(\tilde{x},y) \leqslant f(x,y) + C\|x-\tilde{x}\|_\infty^\alpha,$$

since f is α-Hölder in the whole space $\mathcal{A}^1 \times \mathcal{A}^2$, so that

$$\max_{y\in F(\tilde{x})} f(\tilde{x},y) \leqslant \max_{y\in F(\tilde{x})} [f(x,y)] + C\|x-\tilde{x}\|_\infty^\alpha.$$

Then, for any $\tilde{x} \in B_{\|\cdot\|_\infty}(x,\epsilon)$, let us define I by

$$I \stackrel{\text{def}}{=} |\max_{y\in F(\tilde{x})} f(x,y) - \max_{y\in F(x)} f(x,y)|. \tag{5.6}$$

Let $y^* \in \arg\max_{y\in F(x)} f(x,y)$ and $\tilde{y}^* \in \arg\max_{y\in F(\tilde{x})} f(x,y)$ (whose existence is given by f's continuity on a compact set).

Assuming without loss of generality that $f(x, y^*) \geqslant f(x, \tilde{y}^*)$, we have

$$|I| = f(x, y^*) - \max_{y \in F(\tilde{x})} f(x, y) \tag{5.7}$$

$$\leqslant f(x, y^*) - f(x, \Pi_{F(\tilde{x})}(y^*)), \tag{5.8}$$

where $\Pi_{F(\tilde{x})}(y) \in \arg\min_{\beta \in F(\tilde{x})} \|y^* - \beta\|_\infty$, which exists as a minimum as we are dealing with compact sets of $\mathbb{R}^n$. Now,

$$\|y^* - \Pi_{F(\tilde{x})}(y^*)\|_\infty \leqslant \sup_{y \in F(x)} \|y - \Pi_{F(\tilde{x})}(y)\|_\infty \tag{5.9}$$

$$\leqslant H(F(x), F(\tilde{x})). \tag{5.10}$$

Thus,

$$f(x, y^*) - f(x, \Pi_{F(\tilde{x})}(y^*)) \leqslant C H(F(x), F(\tilde{x}))^\alpha \tag{5.11}$$

$$\leqslant C\lambda \|x - \tilde{x}\|_\infty^\alpha \tag{5.12}$$

and, finally, $|I| \leqslant C\lambda\|x - \tilde{x}\|_\infty^\alpha \leqslant C\lambda\epsilon^\alpha$ so that $\forall \tilde{x} \in B_{\|\cdot\|_\infty}(x, \epsilon)$,

$$\left| \max_{y \in F(x)} f(x, y) - \max_{y \in F(\tilde{x})} f(\tilde{x}, y) \right| \leqslant C\lambda\|x - \tilde{x}\|_\infty^\alpha + C\lambda\|x - \tilde{x}\|_\infty$$

$$\leqslant C\lambda\epsilon \left(1 + \frac{\epsilon^\alpha}{\epsilon}\right).$$

This defines the upper bound of a cell $B_{\|\cdot\|_\infty}(x, \epsilon)$ so that `DOO` applies to the computation of the outer function $g : x \mapsto \max_{y \in F(x)} f(x, y)$. □

5.6 Experiments

The following aims at studying the behavior of `BiDOO` (i) relatively to its hyperparameters (Sections 5.6.1 and 5.6.2) and (ii) for games with dependent feasible set (Section 5.6.3). All experiments ran on an Ubuntu machine with i7-10810U 1.10 GHz Intel processor, 16 GB available `RAM`.

When considering normal-form games, the exact min - max value is given by the resolution of a linear program using Cplex[35], and the α-Hölder constants of the inner and the outer processes are obtained using the following lemma.

Lemma 5.6.1. *$f : \boldsymbol{x} \mapsto \max_{\boldsymbol{y}} \boldsymbol{x}^\top \cdot M \cdot \boldsymbol{y}$ is $\sum_i \max_j |m_{i,j}|$-Lipschitz and, $\forall \boldsymbol{x}$, the function $\boldsymbol{y} \mapsto \boldsymbol{x}^\top \cdot M \cdot \boldsymbol{y}$ is $\sum_j \max_i |m_{i,j}|$-Lipschitz when $\boldsymbol{x}$,$\boldsymbol{y}$ are elements of the unit simplex of $\mathbb{R}^n$.*

Proof. First, we show that $\forall \boldsymbol{x}, \boldsymbol{y} \mapsto \boldsymbol{x}^\top \cdot M \cdot \boldsymbol{y}$ is $\sum_j \max_i |m_{i,j}|$-Lipschitz.

$$\sup_{\boldsymbol{y}, \tilde{\boldsymbol{y}} \in S_n(1)} \frac{|\boldsymbol{x}^\top \cdot M \cdot \boldsymbol{y} - \boldsymbol{x}^\top \cdot M \cdot \tilde{\boldsymbol{y}}|}{\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|_\infty} \tag{5.13}$$

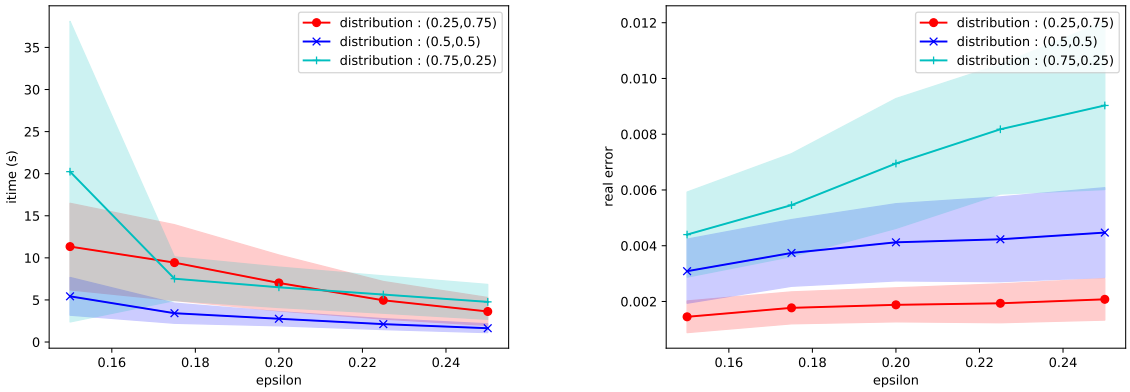
$$= \sup_{\boldsymbol{y}, \tilde{\boldsymbol{y}} \in S_n(1)} \frac{|\boldsymbol{x}^\top \cdot M \cdot (\boldsymbol{y} - \tilde{\boldsymbol{y}})|}{\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|_\infty} \tag{5.14}$$

$$= \sup_{\boldsymbol{y}, \tilde{\boldsymbol{y}} \in S_n(1)} \frac{|\sum_j (\boldsymbol{x}M)_j (\boldsymbol{y}_j - \tilde{\boldsymbol{y}}_j)|}{\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|_\infty} \tag{5.15}$$

$$\leqslant \sup_{\boldsymbol{y}, \tilde{\boldsymbol{y}} \in S_n(1)} \frac{\sum_j |(\boldsymbol{x}M)_j| |y_j - \tilde{y}_j|}{\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|_\infty} \tag{5.16}$$

$$\leqslant \sup_{\boldsymbol{y}, \tilde{\boldsymbol{y}} \in S_n(1)} \frac{\max_j |y_j - \tilde{y}_j| \sum_j |(\boldsymbol{x}M)_j|}{\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|_\infty} \tag{5.17}$$

[35] Cplex 12.1 https://www.ibm.com/fr-fr/analytics/cplex-optimizer



(a) BiDOO's convergence time (average and standard error) as a function of ϵ for 3 different distributions.

(b) BiDOO's observed error (average and standard error) as a function of ϵ for 3 different distributions.

$$= \sup_{\boldsymbol{y},\tilde{\boldsymbol{y}}\in S_n(1)} \sum_j \sum_i x_i |m_{i,j}| \tag{5.18}$$

$$\leqslant \sup_{\boldsymbol{y},\tilde{\boldsymbol{y}}\in S_n(1)} \sum_j \max_i |m_{i,j}| \sum_i x_i \tag{5.19}$$

$$= \sum_j \max_i |m_{i,j}| \qquad \text{since } \boldsymbol{x} \in S_n(1). \tag{5.20}$$

Now, we show that $\boldsymbol{x} \mapsto \max_{\boldsymbol{y}} \boldsymbol{x}^\top \cdot M \cdot \boldsymbol{y}$ is $\sum_i \max_j |m_{i,j}|$-Lipschitz. For all $(\boldsymbol{x}, \tilde{\boldsymbol{x}})$, assuming $\max_{\boldsymbol{y}} \boldsymbol{x}^\top \cdot M \cdot \boldsymbol{y} \geqslant \max_{\boldsymbol{y}} \tilde{\boldsymbol{x}}^\top \cdot M \cdot \boldsymbol{y}$, and writing $\boldsymbol{y}^*$ for an element of $\arg\max_{\boldsymbol{y}} \boldsymbol{x}^\top \cdot M \cdot \boldsymbol{y}$,

$$|\max_{\boldsymbol{y}} \boldsymbol{x}^\top \cdot M \cdot \boldsymbol{y} - \max_{\boldsymbol{y}} \tilde{\boldsymbol{x}}^\top \cdot M \cdot \boldsymbol{y}| \tag{5.21}$$

$$\leqslant \boldsymbol{x}^\top \cdot M \cdot \boldsymbol{y}^* - \tilde{\boldsymbol{x}}^\top \cdot M \cdot \boldsymbol{y}^* \tag{5.22}$$

$$= (\boldsymbol{x} - \tilde{\boldsymbol{x}})^\top \cdot M \cdot \boldsymbol{y}^* \tag{5.23}$$

$$\leqslant \sum_i |(M \cdot \boldsymbol{y}^*)|_i \|x - \tilde{x}\|_\infty \tag{5.24}$$

$$\leqslant (\sum_i \max_j |m_{i,j}|) \|x - \tilde{x}\|_\infty \qquad \text{since } y \in S_n(1), \tag{5.25}$$

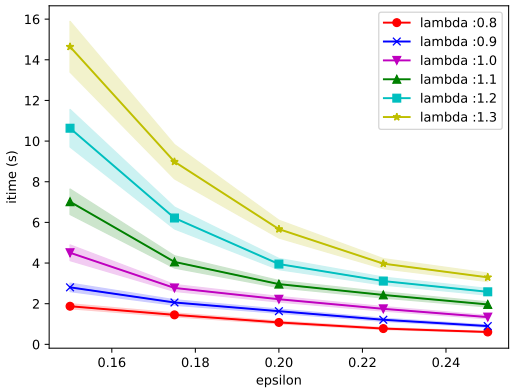
so that $\sup_{x,\tilde{x}} \frac{|\max_{\boldsymbol{y}} \boldsymbol{x}^\top \cdot M \cdot \boldsymbol{y} - \max_{\boldsymbol{y}} \tilde{\boldsymbol{x}}^\top \cdot M \cdot \boldsymbol{y}|}{\|x - \tilde{x}\|_\infty} \leqslant \sum_i \max_j |m_{i,j}|$. □

5.6.1 Choosing the ϵ-distribution

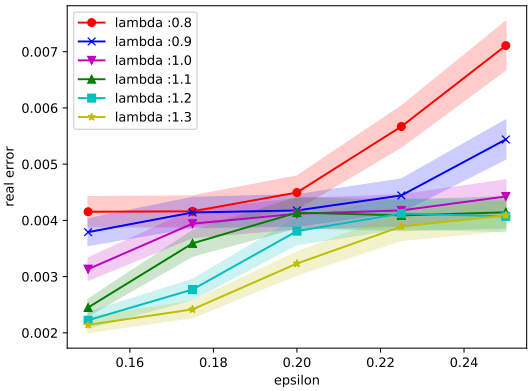
A question is whether one should choose $\epsilon_1 = \epsilon_2 = \frac{1}{2}\epsilon$ or another distribution, such as $\epsilon_1 = \frac{3}{4}\epsilon$ or $\epsilon_2 = \frac{1}{4}\epsilon$. To partially study this, Figures 5.4a and 5.4b respectively provide BiDOOs computation time and actual error when solving the min - max problem for randomly generated 3×3 matrices taken in $[0,1]^{3\times 3}$. Results are randomized over 30 matrices. We observe that (i) with the 50/50 distribution, BiDOO appears to converge faster, but (ii) with the 25/75 distribution, BiDOO provides a lower error in average. While the error results appear very intuitive, the timing ones are a bit more surprising. One could expect the 25/75 distribution BiDOO to be faster than the 50/50 one as it is less demanding with the inner process, which is called exponentially more often than the outer one. Still, as the number of iterations required to provably reach an error smaller than a given $\epsilon = \epsilon_1 + \epsilon_2$ is exponential w.r.t ϵ, the 25/75 distribution BiDOO may require way more time for the outer DOO to converge. This might, as a whole, take more time. In all the following experiments, we pick $\epsilon_1 = \epsilon_2$.

5.6.2 Validating the approach

We consider 100 randomly generated matrices $M \in [0,1]^{3\times 3}$. Figures 5.5a and 5.5b respectively give the computation time as a function of the error imposed to the algorithm and the error



(a) Computation time (in seconds) for Algorithm 5.2 as a function of the error ϵ imposed, for different Hölder constants. If C^1 and C^2 denote the 1-Hölderian constants of the outer and the inner DOO processes, the curve labelled "lambda : 1.1" corresponds to launching BiDOO with $C^1 \cdot 1.1$ and $C^2 \cdot 1.1$.



(b) Error of Algorithm 5.2 compared to the exact value as a function of the error ϵ imposed, for different Hölder constants. If C^1 and C^2 denote the 1-Hölderian constants of the outer and the inner DOO processes, the curve labelled "lambda : 1.1" corresponds to launching BiDOO with $C^1 \cdot 1.1$ and $C^2 \cdot 1.1$.

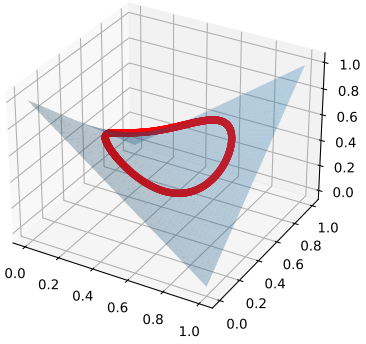


Figure 5.6: GDA applied to the problem $\max_x \min_y x \cdot y^\top$. Red points correspond to points visited by the GDA algorithm.

compared to the exact value.

Several things are to be noted on this figure. Firstly, we observe consistency, *i.e.,* BiDOO's error with respect to the exact value always being lower than the error ϵ imposed to BiDOO. Secondly, and as predicted by Section 5.5.1, the time taken by the algorithm to converge appears exponential with respect to ϵ. Interestingly, BiDOO returns a value that is guaranteed to be ϵ-close to the optimum, but which is actually around $\epsilon/10$-close to the exact value.

Looking back at Theorem 5.5.3, we compared a uniform search iteratively building a small enough uniform grid to BiDOO using the same set of 100 randomly-generated 3×3 matrices, for $\epsilon = 0.15$. It takes the uniform search in average 7.8 s (std. error 0.9 s) to end, whereas BiDOO converges on average in 2.4 s (std. error 0.2 s).

5.6.3 Comparison with the state of the art

Bilinear Games and Gradient Descent Ascent It is known (Zhang et al. 2020) that "vanilla" GDA (*i.e.,* with constant step size) can fail even for the bilinear game $\boldsymbol{x}^\top \cdot I_n \cdot \boldsymbol{y}$, where $x, y \in \mathbb{R}^n$, and I_n is the n-dimensional identity matrix. An illustration of GDA's behavior is given in Figure 5.6. As the game is bilinear, it is *a fortiori* α-Hölder, and BiDOO provably solves the min - max problem.

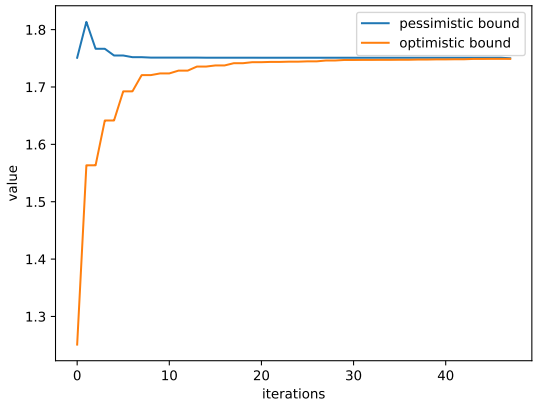


Figure 5.7: Upper and lower bounds as a function of the iteration number for the optimization problem in Expression 5.26.

Games with dependent feasible sets The difficulties with running a gradient-based algorithm to solve general min - max problems can be illustrated (Goktas et al. 2022) through the computation of

$$\min_{x\in[-1,1]} \max_{\substack{y\in[-1,1],\\ 1-(x+y)\geqslant 0}} x^2 + y + 1. \tag{5.26}$$

In such a game, `GDA` converges to $(0, 1)$, which is not a solution of the min - max problem, whereas `BiDOO` does converge to an ϵ-optimum as illustrated in Figure 5.7, provided the subdivision process is modified to satisfy the constraint $y \in [-1, 1]$, and $1 - (x + y) \geqslant 0$. This result is expected as this example satisfies the assumptions of Section 5.5.2.

Interestingly, Figure 5.7 shows a pessimistic bound within ϵ of the optimum way earlier than the optimistic bound. This behavior is frequent with approaches which guarantee ϵ-optimality using optimistic and pessimistic bounds; proving optimality (*i.e.,* lowering the optimistic bound ϵ-closely to the optimum) is harder than heuristically finding near-optimal points. This is coherent with the observation made in Section 5.6.2 that `DOO` often returns values closer to the optimum compared to the imposed error.

5.7 Conclusion

In this chapter, we proposed an algorithmic scheme, `BiDOO`, for the min - max optimization problem of α-Hölder functions, leading to a finite-time convergent algorithm to an ϵ-global optimum. The mild α-Hölder assumption prevents from relying on differentiability for optimization, and we instead built on the `DOO` approach for single-variable optimization problem under Lipschitz-continuity assumptions with respect to any semi metric. Our approach offers some robustness, as we proved its convergence to a ϵ-global optimum for games with dependent feasible sets, under Lipschitz-continuity properties of the dependence. On the downside, the time complexity of `BiDOO` is exponential with respect to both (i) the dimension of the game and (ii) $1/\epsilon$, which limits its scalability.

The relative simplicity of the algorithmic scheme allows considering several lines of future work, some being discussed below.

5.8 Future Work

Two main future lines of research are presented below. One approaches practical experimentation for real-life problems while the other digs deeper into theoretical concerns to study other types of games with weak continuity properties.

5.8.1 BiDOO and Games With Dependent Feasible Sets

Section 5.5.2 discussed to what extent the BiDOO algorithm solving $\max - \min$ optimization problems can be applied to games with dependent feasible sets. These games are not uncommon and, for instance, arise in well-known Fisher market problems (Goktas et al. 2021). Implementing such problems and conducting a comparative analysis of BiDOO against current state-of-the-art methods is a short-term objective.

5.8.2 BiSOO to Solve Some General-Sum Stackelberg Games

Without particular hypotheses on players' payoff functions, Stackelberg games can be extremely difficult to solve. For example, the function to maximize $\mu^1 \mapsto f^1(\pi^1, \pi^{2,*})$ given that $\pi^{2,*} \in \arg\max_{\pi^2} f^2(\pi^1, \pi^2)$ is discontinuous in general and each discontinuity can be reduced to a single point (Bressan et al. 2019). It was however shown that some differentiability assumptions forbid discontinuities with null Lebesgue measure (Bressan et al. 2019), but player 1 is only allowed a 1-dimensional bounded strategy space. We conjecture that an extension of BiDOO relying on the SOO algorithm designed to maximize single-variable Lipschitz functions with unknown smoothness would solve general-sum Stackelberg games, under Bressan et al.'s hypotheses. Extending Bressan et al.'s result to the multi-dimensional case would be of great interest, but currently seems to require powerful differential geometry tools.

6

Conclusion and Perspectives

Contents

6.1 Conclusion

Optimally planning in partially observable stochastic games is a computationally difficult task in general. Multiple research axes exist to improve the empirical performances of ϵ-optimal algorithms. Of particular interest for this manuscript is the introduction of statistics summarizing players' past behaviors and equivalent games induced by the latter statistics.

For some subclasses of POSGs (*e.g.,* zs-SGs, MDPs, POMDPs and cp-POSGs), the induced games involve optimization processes that link solutions of subproblems through simple concatenations, *i.e.,* allow Bellman's optimality principle to apply to the computation of Nash equilibrium strategy profiles. Approaches tackling subclasses of zs-POSGs (*e.g.,* OS-POSGs, PO-POSGs) also introduced such statistics (Horák et al. 2017; Horák et al. 2019b) and games, but linking solutions of subproblems was an open question. Instead, they relied on continual re-solving techniques to ensure retrieving Nash equilibrium strategy profiles for the original game. A novel interpretation of Bellman's optimality principle to optimally planning in zs-oMGs appears worthwhile, as, contrary to before, (i) the game is not fully observable and (ii) the optimization problem prevents from constructing solutions to problems using solutions of subproblems through simple concatenations.

Besides, a downside of this methodology is that it creates a prohibitive bottleneck by entangling exponentially many decision variables in Bellman's optimality equations, whose computation is needed to perform point-based backup routines used by heuristic search (or point-based) value iteration algorithmic schemes.

6.1.1 Planning in zs-POSGs through zs-oMGs

The first contribution of this manuscript contributes a dynamic line of research involving oMGs, aiming at optimally planning in POSGs. We have demonstrated that, building upon zs-oMGs (*e.g.,* after showing that Bellman's optimality principle applies in such games), one can design an HSVI-like algorithm converging in finite time to an ϵ-NES of any zs-POSG. Doing so allows leveraging structure that zs-POSGs possess compared to zs-EFGs (*e.g.,* the possibility to perform

lossy or lossless compression). It also unveils complementary levers (*e.g.,* breaking problems into subproblems, enabling knowledge generalization between subproblems, and guiding search based on relaxations) in addition to those already exhibited by state-of-the-art approaches, *e.g.,* regret minimization and linear programming. Finally, our findings include shading a new light on Bellman's optimality principle for `zs-oMG`s, continuity properties of classical performance-evaluating functions, along with transformations of the usual Bellman operators to enable efficient subgame solving. We believe that the latter point is interesting, as tackling `zs-POSG`s through the introduction of sufficient statistics or dynamic programming recently received particular attention (Brown et al. 2020; Sokota et al. 2023; Vojtěch Kovařík et al. 2022b; Horák 2019) and appears to be a promising research line.

6.1.2 `cp-POSG`s under hierarchical information-sharing

In a second contribution, we exhibit and study a subclass of `cp-POSG`s that permits addressing the entanglement of players' decision variables, consequently reducing the complexity of point-based backups. Players are assumed to be organized in a linear hierarchy. Their knowledge about the game follows an inclusion relation, *i.e.,* player n only knows her private history, player $n-1$ knows what n knows, and so on. While such organization is more general than those considered before (*e.g.,* considering a hierarchy of n players instead of only two, not assuming that 1 knows the state of the game), we have demonstrated that an adaptation of the `PBVI` algorithmic scheme offers both theoretical error-bounding properties and convincing empirical results (*e.g.,* ϵ-optimally solving some `cp-POSG` benchmark problems with up to 10 players, whereas state-of-the-art solvers rely on weaker solution concepts that do not guarantee optimality). Overall, our work opens the door to real-life applications for which (i) optimality guarantees are essential, and (ii) more players are involved than what was previously considered.

6.1.3 Tackling Games with Weaker Hypotheses

Finally, we studied one-shot games in which (i) players have continuous (but compact) sets of actions, and (ii) payoff functions only satisfy mild continuity properties to partly account for the wide variety of real-life situations. Motivated by the emergence of machine learning problems that prevent from relying on differentiability or convex-concavity, we assumed α-Hölder continuity properties. We presented a simple algorithm, based on Deterministic Optimistic Optimization (`DOO`), that relies on an outer minimization using the solutions of an inner maximization. We believe that our results open the way to tackle other types of games for which the lack of continuity properties prevents the application of most other approaches (*e.g.,* general-sum Stackelberg competitions).

6.2 Perspectives

Throughout the three years of the `Ph.D.` program, several ideas came to our minds. Some involve short-term natural extensions or improvements of our proposed algorithmic schemes, while others require long-term theoretical and practical developments.

6.2.1 Planning in `zs-POSG`s

Below, we present possible future works to improve our proposed `HSVI`-like algorithm for solving `zs-POSG`s. We begin with natural improvements (some were already mentioned in the conclusion of Chapter 3 (Section 3.5)) of key operators and then move on to the longer-term perspective of using neural networks to approximate the optimal value functions of `zs-oMG`s.

6.2.1.1 Improving Operators

As further discussed below, important areas of improvement lie in (i) the efficiency of the selection and update operators used in the `HSVI` scheme (*e.g.,* through double-oracle schemes, pruning, block decomposition of occupancy states) and (ii) the tightness of bounding approximators (*e.g.,*

using smart initializations, or approximations relying on stronger continuity properties). Regarding (i), we remind the reader that the selection operator used in our HSVI-like algorithm (Equation (3.34)) involves solving LPs whose size grows linearly with the number of iterations.

Pruning Pruning is a key ingredient for the convincing results of some algorithmic schemes for POMDPs (*e.g.,* HSVI, PBVI, incremental pruning (Cassandra et al. 1997)) and has been the subject of significant research to study different types of pruning strategies (including different criteria, pruning test frequencies). Empirically, it highly helps reduce the size of bounding approximations (*e.g.,* the set of hyperplanes for the lower bound, and the set of points for the upper bound). While we provide a pruning scheme for bounding approximations $\overline{V}_\tau$ and $\underline{V}_\tau$ (Section 3.4.1), pruning the key bounding approximations $\overline{W}_\tau$ and $\underline{W}_\tau$ is still an open question, as it would involve solving time-consuming quadratic optimization problems. We believe our algorithm could benefit from adapting relaxed pruning criteria (Smith et al. 2005) that would be less precise but also less time-consuming.

Double Oracle Double-oracle schemes (see Section 2.2.2.3) for solving zs-POSGs (Bošanský et al. 2014; McAleer et al. 2021; Lanctot et al. 2017) are beneficial whenever there exists a Nash equilibrium with a small number of pure strategies in its support.

The benefits of using double-oracle schemes is highly problem-dependent and we do not yet have enough insight to grasp the overall efficiency improvements the method could bring in our specific case.

Note that both pruning techniques and double-oracle ones aim, but in a complementary manner, at reducing the burden of selection and update Bellman operators, by only considering relevant rows and columns in LPs.

Exhibiting Common Knowledge in Occupancy States Section 3.4.2 detailed a procedure exhibiting common knowledge at planning time through block-diagonal decomposition of occupancy states. It allows branching over common-knowledge blocks by considering them independently. Whenever an exact decomposition is possible, it greatly reduces the time complexity of Bellman's operators (*e.g.,* LPs Equation (3.34)). We further believe that it would be worthwhile to study decomposition techniques that, with a bounded loss, transform an occupancy state in a block-diagonal one. Still, all decompositions with bounded loss are not equivalent. In fact, a decomposition that minimizes the size of the largest block would have higher impact on Bellman's operator than a decomposition isolating blocks of small size while leaving a large block.

Inititializations Heuristic search value iteration schemes are known to empirically benefit from smart initializations (Smith et al. 2005; Horák et al. 2017; Buffet et al. 2020) of bounding functions, which are usually obtained by solving relaxations of the original problem (*e.g.,* giving full observability to a player, assigning a strategy to a player, assuming that player 2 aims at maximizing the expected sum of rewards instead of minimizing it). Relaxations judiciously guide the search towards interesting parts of the occupancy state. Solving the relaxed games permits constructing upper and lower bounds of the optimal value function of the original zs-POSG (*e.g.,* one can construct an upper bound using the solution of the MDP in which 1 is given full observability, while 2's strategy is fixed). Ideally, the relaxation shall be as close as possible to the original problem while having significantly lower time complexity to ensure that solving relaxations requires negligible time compared to the total solving time.

Getting Rid of Lipschitz Approximations Finally, Fehr et al. emphasized in 2018 (though on particular POMDPs, not for zs-POSGs) that Lipschitz approximations might be rather loose, especially when using theoretical global Lipschitz constants that often are too large. We believe that one might be able to replace the Lipschitz generalization over conditional occupancy states by relying on the concavity of the optimal value function with respect to the marginal distribution

of player 1 and the convexity with respect to the marginal distribution of player 2. Still, it remains an open question how to design adequate approximation functions and efficient update operators.

6.2.1.2 Scaling-up Approximation Functions

The introduction of neural networks pre-trained on abstractions/relaxations of the original problem highly contributed to the success of solving algorithms based on continual re-solving for variants of poker (Moravčík et al. 2017; Brown et al. 2018). The neural network was incorporated in the algorithmic scheme to approximate the value of the nodes of a `zs-EFG` beyond a certain depth. Still, the continual re-solving scheme is fundamentally different from our proposed `HSVI`-like algorithm, which relies on occupancy Markov games. Recently, Vojtěch Kovařík et al. (2022) used neural networks to approximate the optimal value of occupancy states. Including such more scalable approximators might help improving the empirical performances of `HSVI` schemes for `zs-POSG`s and subclasses (*e.g.,* zero-sum one-sided `POSG`s, zero-sum `POSG`s with public observations). It, however, remains unknown how to retrieve a solution strategy based on the computed optimal value function and whether re-solving schemes (or other techniques) would be necessary to do so.

Still, this involves open questions. Vojtěch Kovařík et al. (2022) for example point out negative continuity properties of the resulting approximation, while exploiting continuity is key in `HSVI`-like approaches. Besides, the neural network's architecture might add additional constraints to the greedy selection operator, while its scalability is at the core of `HSVI`-like algorithms. Overall, in our humble opinion, the latter algorithmic scheme might not be as well suited as those relying on regret computations (or even double-oracle (Lanctot et al. 2017)) for the use of neural network approximations.

Still, recent work may prove us wrong, as Yan et al. (2023) appear to soundly incorporate neural networks to model players' perception of their environment (though not for optimal value function approximations and under one-sided observability assumptions), while maintaining efficient backup and update operators in an `HSVI`-like algorithm.

6.2.2 `cp-POSG`s under Hierarchical Information-Sharing

Below, we discuss possible extensions of the hierarchical information-sharing structure detailed in the second contribution of this manuscript, and present possible future works inspired by the lessons gained from our contributions to ϵ-optimally planning in `his-cp-POSG`s.

6.2.2.1 Towards Sequential Synchronization for `cp-POSG`s

It can be observed, from our contribution for `cp-POSG`s under hierarchical information-sharing, that sequential decision making allows reducing the complexity of Bellman's operator, which is the main bottleneck in `HSVI`-like algorithms for optimally planning in `cp-POSG`s. In our case, we leveraged the hierarchical information-sharing hypothesis by applying Bellman's optimality principle to the computation of Bellman's operator through agent-based decomposition, in reversed order of the hierarchy. Without any modification, Bellman's operator involves enumerating all decision rule profiles at each time step, which becomes intractable when the number of players grows. Introducing a certain form of sequential decision making to enable efficient computation of Bellman's operator for general `cp-POSG`s, while maintaining finite-time convergence properties is of particular interest. It would, for example, scale-up `HSVI`-like schemes for `cp-oMG`s even if each performed trajectory is less informative. This is still an open problem for general `cp-POSG`s.

Koops et al. (2023) recently introduced one possible solution through iterative construction of the tree of behavioral decision rule profiles. More specifically, the proposed algorithm, called `RS-MAA*`, incrementally expands new actions and observations for each agent sequentially, considering their possible private histories. *Partial strategies* specifying actions only for subsets of all possible histories are consequently constructed. Overall, the approach avoids the major burden of costly expansions of decision rule profiles in classical `MAA*` and scales up much better, even outperforming Dibangoye et al.'s state-of-the-art `HSVI` scheme on some problems. Part of

the algorithm's efficiency lies in the action selection being informed by tight, yet cheap to compute, heuristic techniques (whose details are omitted here, as not entirely linked to the current discussion).

6.2.3 Games With Mild Continuity Properties

The third contribution of this manuscript, which addresses min-max optimization problems for functions with mild α-Hölder properties, might also serve as a building block for open research questions.

6.2.3.1 Application of the Approach to Real-Life Problems

The robustness of our proposed approach to games with dependent feasible sets invites experimentations to assess the efficiency of our approach for Fisher-market problems, for example. The latter are economical models introduced by Fisher (1892) to study strategical interaction between n buyers, each with its own budget, and each wanting to acquire products among m available products, according to their own preferences. Fisher-market problems constrain players' buying strategy by requiring that the union of all players' buying demands exactly corresponds to the set of available products. Goktas et al. (2021) leverage the differentiability of the dependence to design an ϵ-closely solving scheme with $O(1/\epsilon^2)$ time complexity. We believe that Fisher's constraint model can be generalized to allow for more complex dependence interactions between players, yielding possibly non-differentiable dependence functions. In that case, `BiDOO` might offer a solving scheme with theoretical guarantees.

6.2.3.2 General-Sum Stackelberg Games

Tackling general-sum Stackelberg games with mild assumption on players' payoff functions is both of particular interest for real-life applications and particularly challenging. Even though Bressan et al. (2019) showed interesting results on the structure of the leader's optimizing function (especially that discontinuities are not reduced to single points), a `BiSOO` extension of our proposed approach would only converge if the follower perfectly optimizes her best response to any strategy of the leader. This is unrealistic for numerical solving schemes. Unfortunately, a near-optimal behavior of the follower can be catastrophic for the leader, as small deviations of the leader's strategy compared to the optimum can arbitrarily deteriorate the leader's payoff, if the leader plays her optimal strategy that assume the optimality of the leader's behavior. The optimization problem from the leader's viewpoint shall be rethinked to make the leader's strategy robust to small deviations around the optimum in the follower's behavior.

A

Appendix

A.1 Synthetic Tables

For convenience, we provide three synthetic tables: Table A.1 to sum up various theoretical properties that are stated in this chapter (assuming a finite temporal horizon), Table A.1 and Table A.2 to respectively sum up the notations and abbreviations used.

More precisely, Table A.1 indicates, for various functions f and variables x, properties that f is known to exhibit with respect to x. We denote by

-	a function with no known (or used) property (see also comment below);
N/A	a non-applicable case;
Lin	a linear function;
LC	a Lipschitz-continuous function;
Cv	(resp. Cc) a convex (resp. concave) function;
$PWLCv$	(resp. $PWLCc$) a piecewise linear and convex (resp. concave) function;
$\perp\!\!\!\perp$	the function being independent of the variable;
$\neg P$	the negation of some property P (*i.e.*,, P is known not to hold).

Note also that, as $\sigma_\tau = \sigma_\tau^{c,1}\sigma_\tau^{m,1}$, the linearity or Lipschitz-continuity properties of any function w.r.t. σ_τ extends to both $\sigma_\tau^{c,1}$ and $\sigma_\tau^{m,1}$. Reciprocally, related negative results extend from $\sigma_\tau^{c,1}$ or $\sigma_\tau^{m,1}$ to σ_τ. In these three columns, we just indicate results that cannot be derived from one of the two other columns.

Table A.1: Known properties of various functions appearing in this work

	σ_τ	$\sigma_\tau^{m,1}$	$\sigma_\tau^{c,1}$	β_τ^i	$\beta_\tau^{\neg i}$
$T(\sigma_\tau, \boldsymbol{\beta}_\tau)$	Lin (prop. 2.2.17, p. 38)	-	-	Lin (prop. 2.2.17, p. 38)	Lin (prop. 2.2.17, p. 38)
$T_m^i(\sigma_\tau, \boldsymbol{\beta}_\tau)$	Lin (lem. 3.1.8, p. 48)	-	-	Lin (lem. 3.1.8, p. 48)	Lin (lem. 3.1.8, p. 48)
$T_c^i(\sigma_\tau, \boldsymbol{\beta}_\tau)$	-	$\perp\!\!\!\perp$ (lem. 3.1.9, p. 48)	-	$\perp\!\!\!\perp$ (lem. 3.1.9, p. 48)	-
$V_\tau^*(\sigma_\tau)$	LC (sec. 3.1.2.1, p. 51)	$PWLCv$ (thm. 3.1.7, p. 47)	-	N/A	N/A
$W_\tau^{i,*}(\sigma_\tau, \beta_\tau^i)$	LC (from $Q^*_{\tau+1}$ LC)	-	-	$\neg Lin$ (from Q^* $\neg Lin$)	N/A
$\nu^2_{[\sigma_\tau^{c,1}, \beta^2_{\tau:}]}$	N/A	N/A	LC (lem. 3.1.15, p. 53)	N/A	-

Table A.2: Various notations used in this work

$\neg i \stackrel{\text{def}}{=} i$'s opponent. Thus: $\neg 1 = 2$, and $\neg 2 = 1$.

Histories and occupancy states

$\theta_\tau^i \stackrel{\text{def}}{=} (a_0^i, z_1^i, \ldots, a_{\tau-1}^i, z_\tau^i)$ $(\in \Theta^i = \cup_{t=0}^{H-1}\Theta_t^i)$ is a length-τ *action-observation history* (AOH) for i.

$\boldsymbol{\theta}_\tau \stackrel{\text{def}}{=}$ $(\theta^1_\tau, \theta^2_\tau)$ $(\in \boldsymbol{\Theta} = \cup_{t=0}^{H-1} \boldsymbol{\Theta}_t)$ is a *joint* **AOH** at τ.

$\sigma_\tau(\boldsymbol{\theta}_\tau) \stackrel{\text{def}}{=}$ *Occupancy state* (OS) σ_τ $(\in \mathcal{O}^\sigma = \cup_{t=0}^{H-1} \mathcal{O}^\sigma_t$, where $\mathcal{O}^\sigma_\tau \stackrel{\text{def}}{=} \Delta(\boldsymbol{\Theta}_\tau))$, *i.e.,*, probability distribution over joint AOHs $\boldsymbol{\theta}_\tau$ (typically for some applied $\boldsymbol{\beta}_{0:\tau-1}$).

$\sigma^{m,i}_\tau(\theta^i_\tau) \stackrel{\text{def}}{=}$ *Marginal term* of σ_τ from player i's point of view $(\sigma^{m,i}_\tau \in \Delta(\Theta^i_\tau))$.

$\sigma^{c,i}_\tau(\theta^{\neg i}_\tau | \theta^i_\tau) \stackrel{\text{def}}{=}$ *Conditional term* of σ_τ from i's point of view $(\sigma^{c,i}_\tau : \Theta^i_\tau \mapsto \Delta(\Theta^{\neg i}_\tau))$.

$b(s|\boldsymbol{\theta}_\tau) \stackrel{\text{def}}{=}$ *Belief state*, *i.e.,*, probability distribution over states given a joint AOH $(b(s|\boldsymbol{\theta}_\tau) : \mathcal{S} \times \boldsymbol{\Theta}_\tau \mapsto \mathbb{R})$. Can be computed by an HMM filtering process.

$o_\tau \stackrel{\text{def}}{=}$ *Full occupancy state* o_τ $(\in \Delta(\mathcal{S} \times \boldsymbol{\Theta}_\tau))$, *i.e.,*, $Pr(s, \boldsymbol{\theta}_\tau)$ for the current $\boldsymbol{\beta}_{0:\tau-1}$, and thus verifies $\sigma_\tau(\boldsymbol{\theta}_\tau) = \sum_{s \in \mathcal{S}} o_\tau(s, \boldsymbol{\theta}_\tau)$. Is used in the implementation to simplify computations (*e.g.,*, of r_t and $\sigma_{\tau+1}$ through b).

Decision rules and strategies

$\pi^i_{0:\tau} \stackrel{\text{def}}{=}$ A *pure strategy* for i is a mapping $\pi^i_{0:\tau}$ from private histories in Θ^i_t $(\forall t \in \{0\,..\,\tau\})$ to *single* private actions in $\mathcal{A}^i$. By default, $\pi^i \stackrel{\text{def}}{=} \pi^i_{0:H-1}$.

$\mu^i_{0:\tau} \stackrel{\text{def}}{=}$ A *mixed strategy* $\mu^i_{0:\tau}$ for i is a probability distribution over pure strategies. It is used by first sampling one of the pure strategies (at $t = 0$), and then executing it until $t = \tau$.

$\mu^i_{0:\tau'|\sigma_\tau\rangle} \stackrel{\text{def}}{=}$ $(\tau \leqslant \tau')$ is a mixed strategy *compatible* with some OS σ_τ, *i.e.,*, that could induce this OS at τ (assuming an appropriate complementary $\mu^{\neg i}_{0:\tau'|\sigma_\tau\rangle}$).

$\beta^i_\tau \stackrel{\text{def}}{=}$ A *(behavioral) decision rule* (DR) at time τ for i is a mapping β^i_τ from private AOHs in Θ^i_τ to *distributions* over private actions. We denote $\beta^i_\tau(\theta^i_\tau, a^i)$ the probability to pick a^i when facing θ^i_τ.

$\beta^i_{\tau:\tau'} \stackrel{\text{def}}{=}$ $(\beta^i_\tau, \ldots, \beta^i_{\tau'})$ is a *behavioral strategy* for i from time step τ to τ' (included).

$rw^i(\theta^i_\tau, a^i_\tau) \stackrel{\text{def}}{=} \prod_{t=0}^{\tau} \beta^i_{0:}(a^i_t | a^i_0, z^i_1, a^i_1, \ldots, z^i_t)$ is the *realization weight* (RW) of sequence $a^i_0, z^i_1, a^i_1, \ldots, a^i_\tau (= \theta^i_\tau, a^1_\tau)$ under strategy $\beta^i_{0:}$.

$rw^i(\phi^i_{\tau:\tau'} | \theta^i_\tau) \stackrel{\text{def}}{=} \prod_{t=\tau}^{\tau'} \beta^i_{0:}(a^i_t | \theta^i_\tau, a^i_\tau, \ldots, z^i_t)$ is the RW of a *suffix sequence* $\phi^i_{\tau:\tau'} = a^i_\tau, \ldots, a^i_{\tau'}$ "conditioned" on a *prefix sequence/**AOH*** θ^i_τ.

$\boldsymbol{\pi}_{0:\tau} \stackrel{\text{def}}{=}$ is a *pure strategy profile*.

$\boldsymbol{\mu}_{0:\tau} \stackrel{\text{def}}{=}$ is a *mixed strategy profile*.

$\boldsymbol{\mu}_{0:\tau'|\sigma_\tau\rangle} \stackrel{\text{def}}{=}$ $(\tau \leqslant \tau')$ is a mixed strategy profile *compatible* with some OS σ_τ, *i.e.,*, that could induce this OS at τ.

$\boldsymbol{\beta}_\tau \stackrel{\text{def}}{=} \langle \beta^1_\tau, \beta^2_\tau \rangle$ $(\in \mathcal{B} = \cup_{t=0}^{H-1} \mathcal{B}_t)$ is a *decision rule profile*.

$\boldsymbol{\beta}_{\tau:\tau'} \stackrel{\text{def}}{=} \langle \beta^1_{\tau:\tau'}, \beta^2_{\tau:\tau'} \rangle$ is a *behavioral strategy profile*.

Rewards and value functions

$r_{\max} \stackrel{\text{def}}{=} \max_{s,\boldsymbol{a}} r(s, \boldsymbol{a})$ Maximum possible reward.

$r_{\min} \stackrel{\text{def}}{=} \min_{s,\boldsymbol{a}} r(s, \boldsymbol{a})$ Minimum possible reward.

$V_\tau(\sigma_\tau, \boldsymbol{\beta}_{\tau:}) \stackrel{\text{def}}{=} E[\sum_{t=\tau}^{H-1} \gamma^t R_t \mid \sigma_\tau, \boldsymbol{\beta}_{\tau:}]$, *Value* of $\boldsymbol{\beta}_{\tau:H-1}$ in OS σ_τ.
where R_t is the random var. for the reward at t.

$V^*_\tau(\sigma_\tau) \stackrel{\text{def}}{=} \max_{\beta^1_{\tau:}} \min_{\beta^2_{\tau:}} V_\tau(\sigma_\tau, \boldsymbol{\beta}_{\tau:})$ *Optimal value function*

$Q^*_\tau(\sigma_\tau, \boldsymbol{\beta}_\tau) \stackrel{\text{def}}{=} r(\sigma_\tau, \boldsymbol{\beta}_\tau) + \gamma V^*_{\tau+1}(T(\sigma_\tau, \boldsymbol{\beta}_\tau))$ *Opt. (joint) action-value fct.*

$W^{i,*}_\tau(\sigma_\tau, \beta^i_\tau) \stackrel{\text{def}}{=} opt_{\beta^{\neg i}_\tau} Q^*_\tau(\sigma_\tau, \boldsymbol{\beta}_\tau)$, *Opt. (individual) action-value fct.*
where $opt = \max$ if $i = 1$, min otherwise.

$\nu^2_{[\sigma^{c,1}_\tau, \beta^2_{\tau:}]} \stackrel{\text{def}}{=}$ Vector of values (one component per AOH θ^1_τ) for 1's best response to $\beta^2_{\tau:}$ assuming $\sigma^{c,1}_\tau$. This solution of a POMDP allows computing V^*_τ (see Theorem 3.1.7).

Approximations

$\overline{V}_\tau(\sigma_\tau) \stackrel{\text{def}}{=}$ Upper bound approximation of $V^*_\tau(\sigma_\tau)$; relies on data set $\overline{\mathcal{I}}_{\tau-1}$.

$\underline{V}_\tau(\sigma_\tau) \stackrel{\text{def}}{=}$ Lower bound approximation of $V^*_\tau(\sigma_\tau)$; relies on data set $\underline{\mathcal{I}}_{\tau-1}$.

$\overline{W}_\tau(\sigma_\tau, \beta^1_\tau) \stackrel{\text{def}}{=}$ Upper bound approximation of $W^{*,1}_\tau(\sigma_\tau, \beta^1_\tau)$; relies on data set $\overline{\mathcal{I}}_\tau$.

$\underline{W}_\tau(\sigma_\tau, \beta^2_\tau) \stackrel{\text{def}}{=}$ Lower bound approximation of $W^{*,2}_\tau(\sigma_\tau, \beta^2_\tau)$; relies on data set $\underline{\mathcal{I}}_\tau$.

$\overline{\nu}^2_\tau$	$\stackrel{\text{def}}{=}$	Vector (with one component per AOH θ^1_τ) used in $\overline{V}_\tau$ and $\overline{W}_{\tau-1}$ (if $\tau \geqslant 1$).
		Miscellaneous
w_τ	$\stackrel{\text{def}}{=}$	Denotes a triplet $\langle \sigma^{c,1}_{\tau-1}, \beta^1_{\tau-1}, \overline{\nu}^2_\tau, \rangle \in \overline{\mathcal{I}}_\tau$ (or a triplet in $\underline{\mathcal{I}}_\tau$).
ψ^2_τ	$\stackrel{\text{def}}{=}$	Distribution over triplets $w_{\tau+1} \in \overline{\mathcal{I}}_{\tau+1}$ (inducing a recursively defined strategy from τ to $H-1$). Often denotes the strategy it induces.
$x^\top$	$\stackrel{\text{def}}{=}$	The transpose of a (usually column) vector x of $\mathbb{R}^n$.
$c[y]$	$\stackrel{\text{def}}{=}$	Denotes field c of object/tuple y.
$\texttt{supp}(d)$	$\stackrel{\text{def}}{=}$	Support of distribution d, *i.e.*,, set of its non-zero probability elements.

Table A.3: Various abbreviations used in this work

zs-POSG (Hansen et al. 2004)	$\stackrel{\text{def}}{=}$	zero-sum Partially Observable Stochastic Game
zs-oMG	$\stackrel{\text{def}}{=}$	zero-sum Occupancy Markov Game
DP	$\stackrel{\text{def}}{=}$	Dynamic Programming
HS	$\stackrel{\text{def}}{=}$	Heuristic Search
HSVI (Smith et al. 2005)	$\stackrel{\text{def}}{=}$	Heuristic Search Value Iteration
Dec-POMDP (Bernstein et al. 2002)	$\stackrel{\text{def}}{=}$	Decentralized POMDP
EFG (Kuhn 1950)	$\stackrel{\text{def}}{=}$	Extensive Form Game
SFLP (Koller et al. 1996)	$\stackrel{\text{def}}{=}$	Sequence Form Linear Program
CFR (Zinkevich et al. 2007)	$\stackrel{\text{def}}{=}$	Counterfactual Regret Minimization
NEV	$\stackrel{\text{def}}{=}$	Nash Equilibrium Value
NES	$\stackrel{\text{def}}{=}$	Nash Equilibrium Strategy
AOH	$\stackrel{\text{def}}{=}$	Action-Observation History (for player i: θ^i_τ)
DR	$\stackrel{\text{def}}{=}$	Decision Rule (for player i: β^i_τ)

A.2 Strategy Conversion

Here, we come back on the strategy conversion from tree-shaped strategies $\psi^i_{0:}$ to behavioral strategies $\beta^i_{0:}$ evoked in Remark 3.1.20 (Chapter 3, remark 3.1.20).

Let us recall that we use the term "strategy" to refer to any procedure that permits players to take decisions at each time step based on the evolution of the game. Pure, mixed, behavioral strategies match this definition, so as tree strategies (Corollary 3.1.22), which can be executed by

1. executing the decision rule attached to the current node n, then
2. sampling a new node n' according to the distribution over children nodes attached to the current node n, and
3. switching to that node ($n \leftarrow n'$) before repeating/iterating.

Realization weights ((Koller et al. 1994), to be defined later) do not suit the definition, but a behavioral strategy can straightforwardly be derived from them. Conversely, any strategy induces unique realization weights.

Any behavioral strategy can easily be re-written as a tree one with a tree restricted to a single branch. We will here see (in the finite horizon setting) how to derive from a tree strategy ψ^i_0 a unique (since only reachable histories are considered) equivalent behavioral strategy $\beta^i_{0:}$ using realization weights in intermediate steps. To that end, we first define these realization weights in the case of a behavioral strategy (rather than for a mixed strategy as done by Koller et al.) and present some useful properties before briefly describing the conversion process detailed in Algorithm A.1.

Some Properties of Realization Weights Let us denote $rw^i(a_0^i, z_1^i, a_1^i, \dots, a_\tau^i)$ the *realization weight* (RW) of sequence $a_0^i, z_1^i, a_1^i, \dots, a_\tau^i$ under strategy $\beta_{0:}^i$, defined as

$$rw^i(a_0^i, z_1^i, a_1^i, \dots, a_\tau^i) \stackrel{\text{def}}{=} \prod_{t=0}^{\tau} \beta_{0:}^i(a_t^i | a_0^i, z_1^i, a_1^i, \dots, z_t^i) \tag{A.1}$$

$$= rw^i(a_0^i, z_1^i, a_1^i, \dots, a_{\tau-1}^i) \cdot \beta_{0:}^i(a_\tau^i | \underbrace{a_0^i, z_1^i, a_1^i, \dots, z_\tau^i}_{\theta_\tau^i}). \tag{A.2}$$

This definition already leads to useful results such as:

$$\beta_{0:}^i(a_\tau^i | \theta_\tau^i) = \frac{rw^i(\theta_{\tau-1}^i, a_{\tau-1}^i, z_\tau^i, a_\tau^i)}{rw^i(\theta_{\tau-1}^i, a_{\tau-1}^i)}, \tag{A.3}$$

and

$$\forall z_\tau^i, \quad rw^i(\theta_{\tau-1}^i, a_{\tau-1}^i) = rw^i(\theta_{\tau-1}^i, a_{\tau-1}^i) \cdot \underbrace{\sum_{a_\tau^i} \beta(a_\tau^i | \theta_{\tau-1}^i, a_{\tau-1}^i, z_\tau^i)}_{=1} \tag{A.4}$$

$$= \sum_{a_\tau^i} rw^i(\theta_{\tau-1}^i, a_{\tau-1}^i) \cdot \beta(a_\tau^i | \theta_{\tau-1}^i, a_{\tau-1}^i, z_\tau^i) \tag{A.5}$$

$$= \sum_{a_\tau^i} rw^i(\theta_{\tau-1}^i, a_{\tau-1}^i, z_\tau^i, a_\tau^i). \tag{A.6}$$

We now extend Koller et al.'s definition by introducing *conditional realization weights*, where the realization weight of a *suffix sequence* is "conditioned" on a *prefix sequence*:

$$rw^i(\underbrace{a_\tau^i, \dots, a_{\tau'}^i}_{\text{suffix seq.}} | \underbrace{a_0^i, \dots, z_\tau^i}_{\text{prefix seq.}}) \stackrel{\text{def}}{=} \prod_{t=\tau}^{\tau'} \beta_{0:}^i(a_t^i | a_0^i, \dots, z_\tau^i, a_\tau^i, \dots, z_t^i) \tag{A.7}$$

$$= \beta_{0:}^i(a_\tau^i | a_0^i, \dots, z_\tau^i) \cdot rw^i(a_{\tau+1}^i, \dots, a_{\tau'}^i | a_0^i, \dots, z_{\tau+1}^i). \tag{A.8}$$

As can be noted, this definition only requires the knowledge of a partial strategy $\beta_{\tau:}^i$ rather than a complete strategy $\beta_{0:}^i$.

Let $\tau' \geqslant \tau + 1$, and $rw^i[w]$ denote the realization weights of some element w at $\tau + 1$. Let ψ_τ^i be a probability distribution ψ_τ^i over elements w. We already stated that ψ_τ^i defines a strategy, but what are the corresponding realization weights? We have that:

$$rw[\psi_\tau^i](a_{\tau+1}^i, \dots, a_{\tau'}^i | a_0^i, \dots, z_{\tau+1}^i) = \sum_w \psi_\tau^i(w) \cdot rw[w](a_{\tau+1}^i, \dots, a_{\tau'}^i | a_0^i, \dots, z_{\tau+1}^i). \tag{A.9}$$

Indeed,

$$rw[\psi_\tau^i](a_{\tau+1}^i, \dots, a_{\tau'}^i | a_0^i, \dots, z_{\tau+1}^i) \stackrel{\text{def}}{=} \prod_{t=\tau}^{\tau'} \beta_{0:}^i[\psi_\tau^i](a_t^i | a_0^i, \dots, z_\tau^i, a_\tau^i, \dots, z_t^i) \tag{A.10}$$

(where $\beta_{0:}^i[\psi_\tau^i]$ is the behavioral strategy induced by $rw[\psi_\tau^i]$)

$$= \prod_{t=\tau}^{\tau'} Pr_{\beta_{0:}^i[\psi_\tau^i]}(a_t^i | a_0^i, \dots, z_\tau^i, a_\tau^i, \dots, z_t^i) \tag{A.11}$$

$$= \sum_w \psi_\tau^i(w) \prod_{t=\tau}^{\tau'} Pr_{\beta_{0:}^i[w]}(a_t^i | a_0^i, \dots, z_\tau^i, a_\tau^i, \dots, z_t^i) \tag{A.12}$$

$$= \sum_w \psi_\tau^i(w) \prod_{t=\tau}^{\tau'} \beta_{0:}^i[w](a_t^i | a_0^i, \dots, z_\tau^i, a_\tau^i, \dots, z_t^i) \tag{A.13}$$

$$= \sum_w \psi_\tau^i(w) \cdot rw[w](a_{\tau+1}^i, \dots, a_{\tau'}^i | a_0^i, \dots, z_{\tau+1}^i). \tag{A.14}$$

Algorithm A.1: Extracting $\beta^i_{0:}$ from w^i_0

Fct Extract(w^i_0)
  /* Step 1., keeping only $rw(\theta^i_{0:H-1})$ for all $\theta^i_{0:H-1}$ */
  $\left(rw(\theta^i_{0:H-1})\right)_{\theta^i_{0:H-1}} \leftarrow$ **RecGetRWMix**$(0, w^i_0)$
  /* Step 2. */
  for $t = H-2, \dots, 0$ **do**
    forall $\theta^i_{0:t}, a^i_t$ **do**
      $z^i_{t+1} \leftarrow z^i$ s.t. $\beta_t(\cdot|\theta^i_{0:t}, a^i_t, z^i)$ is defined
      $rw(\theta^1_{0:t}, a^i_t) \leftarrow \sum_{a^i_{t+1}} rw(\theta^i_{0:t}, a^i_t, z^i_{t+1}, a^i_{t+1}|-)$
  /* Step 3. */
  for $t = H-1, \dots, 0$ **do**
    forall $\theta^i_{0:t}, a^i_t$ **do**
      $\beta^i_t(a^i_t|\theta^i_{0:t}) \leftarrow \frac{rw^i(\theta^i_{0:t-1}, a^i_{t-1}, z^i_t, a^i_t)}{rw^i(\theta^i_{0:t-1}, a^i_{t-1})}$
  return $\beta^i_{0:}$

Fct RecGetRWMix$(t, w = \langle \beta^i_t, \psi^i_t \rangle)$
  for w' *s.t.* $\psi^i_t(w') > 0$ **do**
    $rwCat[w'] \leftarrow$ **RecGetRWCat**(t, w')
  forall $(a^i_0, \dots, a^i_{H-1})$ **do**
    $rwMix[w](a^i_t, \dots, a^i_{H-1}|a^i_0, \dots, z^i_t)$
    $\leftarrow \sum_{w'} \psi^i_t(w') \cdot rwCat[w'](a^i_{t+1}, \dots, a^i_{H-1}|a^i_0, \dots, z^i_{t+1})$
  return $rwMix[w]$

Fct RecGetRWCat$(t, w = \langle \beta^i_t, \psi^i_t \rangle)$
  if $t = H-1$ **then**
    forall $(a^i_0, \dots, a^i_{H-1})$ **do**
      $rwCat[w](a^i_{H-1}|a^i_0, \dots, z^i_{H-1}) \leftarrow \beta^i_t(a^i_{H-1}|a^i_0, \dots, z^i_{H-1})$
  else
    $rwMix[w] \leftarrow$ **RecGetRWMix**(t, w)
    forall $(a^i_0, \dots, a^i_{H-1})$ **do**
      $rwCat[w](a^i_t, \dots, a^i_{H-1}|a^i_0, \dots, z^i_t) \leftarrow$
       $\beta^i_t(a^i_t|a^i_0, \dots, z^i_t) \cdot rwMix[w](a^i_{t+1}, \dots, a^i_{H-1}|a^i_0, \dots, z^i_{t+1})$
  return $rwCat[w]$

From w^i_0 to $\beta^i_{0:}$ Using the above results, function **Extract** in Algorithm A.1 derives a behavioral strategy $\beta^i_{0:}$ equivalent to the recursive strategy induced by some tuple w^i_0 in 3 steps as follows:

1. **From w^i_0 to $rw(\theta^i_{0:H-1}, a^i_{H-1})$ ($\forall(\theta^i_{0:H-1}, a^i_{H-1})$)** — These (classical) realization weights are obtained by recursively going through the directed acyclic graph describing the recursive strategy, computing *full length* (conditional) realization weights $rw(\theta^i_{t:H-1}, a^i_{H-1}|\theta^i_{0:t})$ (for $t = H-1$ down to 0).

   When in a leaf node, at depth $H-1$, the initialization is given by Equation (A.7) when $\tau = \tau' = H-1$:

$$rw^i(a^i_{H-1}|a^i_0, \dots, z^i_{H-1}) \stackrel{\text{def}}{=} \prod_{t=H-1}^{H-1} \beta^i(a^i_t|a^i_0, \dots, z^i_t)$$
$$= \beta^i(a^i_{H-1}|a^i_0, \dots, z^i_{H-1}).$$

   Then, in the backward phase, we can compute full length realization weights $rw(\theta^i_{t+1:H-1}, a^i_{H-1}|\theta^i_{0:t})$ with increasingly longer suffixes (thus shorter prefixes) using (i) Equation (A.14) (in func-

tion **RecGetRWMix**, line 25) to "mix" several strategies using the distribution ψ_t^i attached to the current w, and (ii) Equation (A.8), with $\tau' = H - 1$, (in function **RecGetRWCat**, line 16) to concatenate the behavioral decision rule β_t^i attached to the current w in front of the strategy induced by the distribution ψ_t^i also attached to w. Note: Memoization can here be used to avoid repeating the same computations.

2. **Retrieving (classical) realization weights** $rw(\theta_{0:t}^i, a_t^i|-)$ **($\forall t$)** — We can now compute realization weights $rw(\theta_{0:t}^i, a_t^i|-)$ for all t's using Equation (A.6) (line 6).

3. **Retrieving behavioral decision rules** β_t^i — Applying Equation (A.3) (line 9) then provides the expected behavioral decision rules.

In practice, lossless compressions are used to reduce the dimensionality of the occupancy state (*cf.* Section 3.2.1), which are currently lost in the current implementation of the conversion. Ideally, one would like to preserve compressions whenever possible or at least retrieve them afterwards, and possibly identify further compressions in the solution strategy.

Bibliography

Abernethy, Jacob et al. (2011). "Blackwell approachability and no-regret learning are equivalent". In: *Proceedings of the 24th Annual Conference on Learning Theory*. JMLR Workshop and Conference Proceedings, pp. 27–46.

Amato, Christopher et al. (2009). "Incremental policy generation for finite-horizon DEC-POMDPs". In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 19, pp. 2–9.

Amato, Christopher et al. (2012). "Optimizing memory-bounded controllers for decentralized POMDPs". In: *arXiv preprint arXiv:1206.5258*.

Amiet, Ben et al. (2021). "Pure nash equilibria and best-response dynamics in random games". In: *Mathematics of Operations Research* 46.4, pp. 1552–1572.

Aras, Raghav et al. (2010). "An investigation into mathematical programming for finite horizon decentralized POMDPs". In: *Journal of Artificial Intelligence Research* 37, pp. 329–396.

Asarin, Eugene et al. (2015). "Entropy games and matrix multiplication games". In: *arXiv preprint arXiv:1506.04885*.

Åström, Karl (1965). "Optimal control of Markov processes with incomplete state information". In: *Journal of Mathematical Analysis and Applications* 10.1, pp. 174–205. ISSN: 0022-247X.

Åström, Karl Johan (1965). "Optimal control of Markov processes with incomplete state information". In: *Journal of mathematical analysis and applications* 10.1, pp. 174–205.

Becker, Raphen et al. (2004). "Solving Transition Independent Decentralized Markov Decision Processes". In: *JAIR* 22, pp. 423–455.

Bellman, Richard E (1957). *Dynamic Programming*. Dover Publications, Incorporated.

Bernstein, Daniel S. et al. (2002). "The complexity of decentralized control of Markov decision processes". In: *Mathematics of operations research* 27.4, pp. 819–840.

Blackwell, David (1956). "An analog of the minimax theorem for vector payoffs." In:

Boissier, Mathilde et al. (2023). "Designing Serious Games to understand the challenges of the Anthropocene". In: *Proceedings of the Design Society* 3, pp. 1397–1406.

Bono, Guillaume et al. (2018). "Cooperative Multi-agent Policy Gradient". In: *ECML-PKDD*, pp. 459–476.

— (2019). "Cooperative multi-agent policy gradient". In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*. Springer, pp. 459–476.

Borel, Emile (1921). "La théorie du jeu et les équations intégralesa noyau symétrique". In: *Comptes rendus de l'Académie des Sciences* 173.1304-1308, p. 58.

Bošanský, Branislav et al. (2014). "An Exact Double-Oracle Algorithm for Zero-Sum Extensive-Form Games with Imperfect Information". In: *Journal of Artificial Intelligence Research* 51, pp. 829–866. DOI: 10.1613/jair.4477.

Bowling, Michael et al. (2015). "Heads-up limit hold'em poker is solved". In: *Science* 347.6218, pp. 145–149.

Brandsen, Sarah et al. (2022). "What is entropy? A perspective from games of chance". In: *Physical Review E* 105.2, p. 024117.

Bressan, Alberto et al. (2019). "On the generic structure and stability of Stackelberg equilibria". In: *Journal of Optimization Theory and Applications* 183, pp. 840–880.

Brown, Noam et al. (2018). "Superhuman AI for Heads-Up No-Limit Poker: Libratus Beats Top Professionals". In: *Science* 359.6374, pp. 418–424.

Brown, Noam et al. (2020). "Combining deep reinforcement learning and search for imperfect-information games". In: *Advances in Neural Information Processing Systems* 33, pp. 17057–17069.

Brunner, Nicolas et al. (2013). "Connection between Bell nonlocality and Bayesian game theory". In: *Nature communications* 4.1, pp. 1–6.

Buffet, Olivier et al. (2020). "Heuristic Search Value Iteration for zero-sum Stochastic Games". In: *IEEE Transactions on Games* 13, pp. 239–248.

Burch, Neil (2018). "Time and space: Why imperfect information games are hard". In:

Burch, Neil et al. (2014). "Solving imperfect information games using decomposition". In: *Twenty-eighth AAAI conference on artificial intelligence*.

Buşoniu, Lucian et al. (2014). "An analysis of optimistic, best-first search for minimax sequential decision making". In: *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*. IEEE, pp. 1–8.

Campbell, Murray et al. (2002). "Deep blue". In: *Artificial intelligence* 134.1-2, pp. 57–83.

Cassandra, Anthony et al. (1997). "Incremental pruning: a simple, fast, exact method for partially observable Markov decision processes". In: *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pp. 54–61.

Chadès, Iadine et al. (2012). "MOMDPs: a Solution for Modelling Adaptive Management Problems". In: *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Chen, Xi et al. (2009). "Settling the complexity of computing two-player Nash equilibria". In: *Journal of the ACM (JACM)* 56.3, pp. 1–57.

Chow, Yinlam et al. (2015). "Risk-sensitive and robust decision-making: a cvar optimization approach". In: *Advances in neural information processing systems* 28.

Cohen-Solal, Quentin (2020). "Learning to play two-player perfect-information games without knowledge". In: *arXiv preprint arXiv:2008.01188*.

Daskalakis, Constantinos (2022). "Non-Concave Games: A Challenge for Game Theory's Next 100 Years". In: *Nobel symposium "One Hundred Years of Game Theory: Future Applications and Challenges*.

Daskalakis, Constantinos et al. (2009). "The complexity of computing a Nash equilibrium". In: *Communications of the ACM* 52.2, pp. 89–97.

Daskalakis, Constantinos et al. (2021). "The complexity of constrained min-max optimization". In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1466–1478.

de Wolf, Olivier (1999). *Optimal strategies in n-person unilaterally competitive games*. LIDAM Discussion Papers CORE 1999049. Université catholique de Louvain, Center for Operations Research and Econometrics (CORE). URL: https://EconPapers.repec.org/RePEc:cor:louvco:1999049.

Delage, Aurélien et al. (May 2023a). "Global Min-Max Computation for α-Hölder Zero-Sum Games". In: *GAIW 2023 - 5th Games, Agents, and Incentives Workshop*. Londres (London), United Kingdom, pp. 1–9. URL: https://inria.hal.science/hal-04382817.

Delage, Aurélien et al. (May 2023b). "Heuristic Search Value Iteration can solve zero-sum Partially Observable Stochastic Games". In: *MSDM 202311th Multiagent Sequential Decision Making under Uncertainty Workshop ; Held as part of the Workshops at the IFAAMAS 2023 - 21st International Conference on Autonomous Agents and Multiagent Systems*. Versions étendues:- https://arxiv.org/abs/2210.14640- https://doi.org/10.1007/s13235-023-00519-6. Londres, United Kingdom. URL: https://inria.hal.science/hal-04382922.

Delage, Aurélien et al. (2023). "HSVI can solve zero-sum partially observable stochastic games". In: *Dynamic Games and Applications*, pp. 1–55.

Dibangoye, Jilles S. et al. (2009). "Point-based incremental pruning heuristic for solving finite-horizon Dec-POMDPs". In: *aamas09*, pp. 569–576.

Dibangoye, Jilles S. et al. (2012). "Scaling Up Decentralized MDPs Through Heuristic Search". In: *UAI*. Ed. by Nando de Freitas et al., pp. 217–226.

Dibangoye, Jilles S. et al. (2013a). "Optimally Solving Dec-POMDPs as Continuous-State MDPs". In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*.

Dibangoye, Jilles S. et al. (2013b). "Optimally Solving Dec-POMDPs as Continuous-State MDPs". In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pp. 90–96.

Dibangoye, Jilles S. et al. (2014a). "Error-bounded approximations for infinite-horizon discounted decentralized POMDPs". In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I 14*. Springer, pp. 338–353.

Dibangoye, Jilles S. et al. (2014b). "Exploiting separability in multiagent planning with continuous-state MDPs". In: *AAMAS 2014-13th International Conference on Autonomous Agents and Multiagent Systems*. ACM.

— (2016). "Optimally Solving Dec-POMDPs as Continuous-State MDPs". In: *Journal of Artificial Intelligence Research* 55, pp. 443–497.

Dibangoye, Jilles S. et al. (2018). "Learning to act in decentralized partially observable MDPs". In: *Proceedings of the Thirty-Fifth International Conference on Machine Learning*.

Du, Jianbo et al. (2017). "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee". In: *IEEE Transactions on Communications* 66.4, pp. 1594–1608.

Edelsbrunner, Herbert et al. (1999). "Edgewise subdivision of a simplex". In: *Proceedings of the fifteenth annual symposium on Computational geometry*, pp. 24–30.

Farina, Gabriele et al. (2021). "Faster game solving via predictive blackwell approachability: Connecting regret matching and mirror descent". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 6, pp. 5363–5371.

Fehr, Mathieu et al. (2018). "rho-POMDPs have Lipschitz-continuous epsilon-optimal value functions". In: *Advances in neural information processing systems* 31.

Fiez, Tanner et al. (2021). "Global convergence to local minmax equilibrium in classes of nonconvex zero-sum games". In: *Advances in Neural Information Processing Systems* 34, pp. 29049–29063.

Fisher, Irving (1892). "Mathematical investigations in the theory of value and prices, and appreciation and interest". PhD thesis. Yale University.

Foerster, Jakob N. et al. (2018). "Counterfactual Multi-Agent Policy Gradients". In: *AAAI*.

Fox, Kyle et al. (2023). "Minimum cut and minimum k-cut in hypergraphs via branching contractions". In: *ACM Transactions on Algorithms* 19.2, pp. 1–22.

Friedman, Daniel (1998). "On economic applications of evolutionary game theory". In: *Journal of evolutionary economics* 8.1, pp. 15–43.

Garcia, Frédérick et al. (2008). *Processus décisionnels de Markov en intelligence artificielle*. Ed. by Olivier Sigaud et al. Vol. 1. Lavoisier - Hermes Science Publications, p. 258.

Garcia, Frédérick et al. (2010). *Markov Decision Processes and Artificial Intelligence*. Ed. by Olivier Sigaud et al. ISBN: 978-1-84821-167-4. ISTE - Wiley, p. 480.

Geibel, Peter (2001). "Reinforcement Learning with Bounded Risk". In: *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pp. 162–169.

Gibbons, Robert S (1992). *Game theory for applied economists*. Princeton University Press.

Gilpin, Andrew et al. (2007). "Lossless abstraction of imperfect information games". In: *Journal of the ACM (JACM)* 54.5, 25–es.

Gmytrasiewicz, Piotr J. et al. (2005). "A Framework for Sequential Planning in Multi-Agent Settings". In: *Journal of Artificial Intelligence Research* 24, pp. 49–79. DOI: 10.1613/JAIR.1579. URL: https://doi.org/10.1613/jair.1579.

Goktas, Denizalp et al. (2021). "Convex-concave min-max Sackelberg games". In: *Advances in Neural Information Processing Systems* 34, pp. 2991–3003.

— (2022). "Gradient Descent Ascent in Min-Max Stackelberg Games". In: *arXiv preprint arXiv:2208.09690*.

Goldberg, Paul W et al. (2006). "Reducibility among equilibrium problems". In: *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pp. 61–70.

Goldsmith, Judy et al. (2007). "Competition adds complexity". In: *Advances in Neural Information Processing Systems* 20.

Gomory, Ralph E et al. (1961). "Multi-terminal network flows". In: *Journal of the Society for Industrial and Applied Mathematics* 9.4, pp. 551–570.

Hadfield-Menell, Dylan et al. (2016). "Cooperative inverse reinforcement learning". In: *Advances in Neural Information Processing Systems*.

Han, Zhu et al. (2012). *Game theory in wireless and communication networks: theory, models, and applications*. Cambridge university press.

Hansen, Eric A. et al. (2004). "Dynamic Programming for Partially Observable Stochastic Games". In: *Proceedings of the Nineteenth National Conference on Artificial Intelligence*.

Hansen, Eric A. et al. (2007). "Anytime Heuristic Search". In: 28, pp. 267–297. DOI: DOI:https://doi.org/10.1613/jair.2096.

Harsanyi, John C. (Jan. 1968). "Games with Incomplete Information Played by "Bayesian" Players, I-III. Part II. Bayesian Equilibrium Points". In: *Management Science* 14.5, pp. 320–334. URL: http://www.jstor.org/stable/2628673.

Harsanyi, John C. et al. (1988). "A general theory of equilibrium selection in games". In: *MIT Press Books* 1.

Hart, Sergiu et al. (2000). "A simple adaptive procedure leading to correlated equilibrium". In: *Econometrica* 68.5, pp. 1127–1150.

Hauert, Christoph et al. (2005). "Game theory and physics". In: *American Journal of Physics* 73.5, pp. 405–414.

Haurie, Alain et al. (2012). *Games and Dynamic Games*. Vol. 1. World Scientific Publishing Company.

Hauskrecht, Milos (2000). "Value-Function Approximations for Partially Observable Markov Decision Processes". In: *Journal of Artificial Intelligence Research* 13, pp. 33–94.

Hopcroft, John et al. (1973). "Algorithm 447: efficient algorithms for graph manipulation". In: *Communications of the ACM* 16.6, pp. 372–378.

Horák, Karel (2019). "Scalable Algorithms for Solving Stochastic Games with Limited Partial Observability". PhD thesis. Czech Technical University in Prague, Faculty of Electrical Engineering.

Horák, Karel et al. (2017). "Heuristic Search Value Iteration for One-Sided Partially Observable Stochastic Games". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 558–564.

Horák, Karel et al. (2019a). "Compact Representation of Value Function in Partially Observable Stochastic Games". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*.

Horák, Karel et al. (2019b). "Solving Partially Observable Stochastic Games with Public Observations". In: *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pp. 2029–2036.

Horák, Karel et al. (2023). "Solving zero-sum one-sided partially observable stochastic games". In: *Artificial Intelligence* 316, p. 103838.

Jagielski, Matthew et al. (2019). "Differentially private fair learning". In: *International Conference on Machine Learning*. PMLR, pp. 3000–3008.

Jin, Chi et al. (2019). *What is Local Optimality in Nonconvex-Nonconcave Minimax Optimization?* DOI: 10.48550/ARXIV.1902.00618. URL: https://arxiv.org/abs/1902.00618.

Johanson, Michael et al. (2011). "Accelerating best response calculation in large extensive games". In: 11, pp. 258–265.

Johanson, Michael et al. (2012). "Efficient Nash equilibrium approximation through Monte Carlo counterfactual regret minimization." In: *Aamas*, pp. 837–846.

Kaelbling, Leslie Pack et al. (1998). "Planning and acting in partially observable stochastic domains". In: *Artificial intelligence*, pp. 99–134.

Koller, Daphne et al. (1994). "Fast Algorithms for Finding Randomized Strategies in Game Trees". In: *Proceedings of the 26th ACM Symposium on the Theory of Computing (STOC'94)*, pp. 750–759.

— (1996). "Efficient Computation of Equilibria for Extensive Two-Person Games". In: *Games and Economic Behavior* 14.51, pp. 220–246.

Konda, Vijay R. et al. (1999). "Actor-Critic Algorithms". In: *Neural Information Processing Systems*.

Koops, Wietze et al. (2023). "Recursive small-step multi-agent A* for dec-POMDPs". In: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*.

Korf, Richard E. (1990). "Real-time heuristic search". In: *Artificial Intelligence* 42.2, pp. 189–211. ISSN: 0004-3702. DOI: https://doi.org/10.1016/0004-3702(90)90054-4.

Kovařík, Vojtěch et al. (2019). "Value Functions for Depth-Limited Solving in Imperfect-Information Games". In: *Corr* abs/1906.06412. arXiv: 1906.06412.

Kovařík, Vojtěch et al. (2022a). "Rethinking formal models of partially observable multiagent decision making". In: *Artificial Intelligence* 303, p. 103645.

Kovařík, Vojtěch et al. (2022b). "Value Functions for Depth-Limited Solving in Zero-Sum Imperfect-Information Games". In: *Artificial Intelligence*, p. 103805. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2022.103805. URL: https://www.sciencedirect.com/science/article/pii/S000437022200145X.

Kozuno, Tadashi et al. (2021). "Learning in two-player zero-sum partially observable Markov games with perfect recall". In: *Advances in Neural Information Processing Systems* 34, pp. 11987–11998.

Kuhn, Harold W. (1950). "Simplified Two-Person Poker". In: *Contributions to the Theory of Games*. Ed. by H. W. Kuhn et al. Vol. 1. Princeton University Press.

Kuhn, Harold W. et al. (1953). *Contributions to the Theory of Games*. 28. Princeton University Press.

Kurniawati, Hanna et al. (2008). "Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces." In: *Robotics: Science and systems*. Vol. 2008. Citeseer.

Lanctot, Marc et al. (2009). "Monte Carlo sampling for regret minimization in extensive games". In: *Advances in Neural Information Processing Systems* 22.

Lanctot, Marc et al. (2017). "A unified game-theoretic approach to multiagent reinforcement learning". In: *Advances in neural information processing systems* 30.

Lanctot, Marc et al. (2019). "OpenSpiel: A Framework for Reinforcement Learning in Games". In: *CoRR* abs/1908.09453. arXiv: 1908.09453 [cs.LG]. URL: http://arxiv.org/abs/1908.09453.

Lemke, Carlton E et al. (1964). "Equilibrium points of bimatrix games". In: *Journal of the Society for industrial and Applied Mathematics* 12.2, pp. 413–423.

Lin, Tianyi et al. (2020). "On gradient descent ascent for nonconvex-concave minimax problems". In: *International Conference on Machine Learning*. PMLR, pp. 6083–6093.

Ling, Chun Kai et al. (2021). "Safe search for stackelberg equilibria in extensive-form games". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 6, pp. 5541–5548.

Liu, Qinghua et al. (2022). "Sample-efficient reinforcement learning of partially observable markov games". In: *Advances in Neural Information Processing Systems* 35, pp. 18296–18308.

López, Víctor Bucarey et al. (2022). "Stationary Strong Stackelberg Equilibrium in Discounted Stochastic Games". In: *IEEE Transactions on Automatic Control*.

Lowe, Ryan et al. (2017). "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments". In: *Advances in Neural Information Processing Systems*. Vol. 30, pp. 6379–6390.

MacDermed, Liam C (2013). "Value methods for efficiently solving stochastic games of complete and incomplete information". PhD thesis. Georgia Institute of Technology.

Madani, Kaveh (2010). "Game theory and water resources". In: *Journal of hydrology* 381.3-4, pp. 225–238.

Madsen, Erik (2013). *Games with no Nash equilibria*. URL: https://www.quora.com/What-are-some-examples-of-games-without-a-Nash-Equilibrium-in-pure-or-mixed-strategies.

Maitra, A et al. (1970). "On stochastic games". In: *Journal of Optimization Theory and Applications* 5, pp. 289–300.

Malik, Dhruv et al. (2018). "An Efficient, Generalized Bellman Update For Cooperative Inverse Reinforcement Learning". In: *Proceedings of the Thirty-Fifth International Conference on Machine Learning*.

Manshaei, Mohammad Hossein et al. (2013). "Game theory meets network security and privacy". In: *ACM Computing Surveys (CSUR)* 45.3, pp. 1–39.

Markovitz, Harry M (1959). *Portfolio selection: Efficient diversification of investments*. John Wiley.

McAleer, Stephen et al. (2021). "XDO: A double oracle algorithm for extensive-form games". In: *Advances in Neural Information Processing Systems* 34, pp. 23128–23139.

McMahan, H Brendan et al. (2003). "Planning in the presence of cost functions controlled by an adversary". In: pp. 536–543.

Meadows, Dennis et al. (2016). *The climate change playbook: 22 systems thinking games for more effective communication about climate change*. Chelsea Green Publishing.

Mescheder, Lars et al. (2018). "Which Training Methods for GANs do actually Converge?" In: DOI: 10.48550/ARXIV.1801.04406. URL: https://arxiv.org/abs/1801.04406.

Moravčík, Matej et al. (2017). "Deepstack: Expert-level artificial intelligence in heads-up no-limit poker". In: *Science* 356.6337, pp. 508–513.

Munos, Rémi (2011). "Optimistic Optimization of a Deterministic Function without the Knowledge of its Smoothness". In: *Advances in Neural Information Processing Systems*. Vol. 24. URL: https://proceedings.neurips.cc/paper/2011/file/7e889fb76e0e07c11733550f2a6c7a5a-Paper.pdf.

Nair, Ranjit et al. (2003). "Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings". In: *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*. Vol. 3, pp. 705–711.

Nair, Ranjit et al. (2005). "Networked Distributed POMDPs: A Synthesis of Distributed Constraint Optimization and POMDPs". In: *AAAI*.

Nash, John F (1950). "Equilibrium points in n-person games". In: *Proceedings of the national academy of sciences* 36.1, pp. 48–49.

Nayyar, Ashutosh et al. (2010). "Optimal control strategies in delayed sharing information structures". In: *IEEE Transactions on Automatic Control*.

Neller, Todd W et al. (2013). "An introduction to counterfactual regret minimization". In: *Proceedings of Model AI Assignments, The Fourth Symposium on Educational Advances in Artificial Intelligence (EAAI-2013)*. Vol. 11.

Nouiehed, Maher et al. (2019). "Solving a class of non-convex min-max games using iterative first order methods". In: *Advances in Neural Information Processing Systems* 32.

Oliehoek, Frans A. (2013). "Sufficient plan-time statistics for decentralized POMDPs". In: *Twenty-Third International Joint Conference on Artificial Intelligence*.

Oliehoek, Frans A. et al. (2008). "Optimal and approximate Q-value functions for decentralized POMDPs". In: *Journal of Artificial Intelligence Research* 32, pp. 289–353.

Oliehoek, Frans A. et al. (2010). "Heuristic search for identical payoff Bayesian games". In: *AAMAS*, pp. 1115–1122.

Oliehoek, Frans A. et al. (2013). "Incremental clustering and expansion for faster optimal planning in Dec-POMDPs". In: *Journal of Artificial Intelligence Research* 46, pp. 449–509.

Oliehoek, Frans A. et al. (2017). "GANGs: Generative adversarial network games". In: *arXiv preprint arXiv:1712.00679*.

Ooi, J.M. et al. (1996). "Decentralized control of a multiple access broadcast channel: performance bounds". In: *CDC*. Vol. 1, 293–298 vol.1. DOI: 10.1109/CDC.1996.574318.

Papadimitriou, Christos H et al. (1987). "The complexity of Markov decision processes". In: *Mathematics of operations research* 12.3, pp. 441–450.

Paulavičius, Remigijus et al. (2014). *Simplicial global optimization*. Springer.

Peralez, Johan et al. (2024). "Solving Hierarchical Information-Sharing Dec-POMDPs: An Extensive-Form Game Approach". In: *arXiv preprint arXiv:2402.02954*.

Perolat, Julien et al. (2022). "Mastering the game of Stratego with model-free multiagent reinforcement learning". In: *Science* 378.6623, pp. 990–996.

Peshkin, Leonid et al. (2001). "Learning to cooperate via policy search". In: *arXiv preprint cs/0105032*.

Pineau, Joelle et al. (2003). "Point-based value iteration: An anytime algorithm for POMDPs". In: *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*. Vol. 3, pp. 1025–1032.

Porter, Ryan et al. (2008). "Simple search methods for finding a Nash equilibrium". In: *Games and Economic Behavior* 63.2, pp. 642–662.

Puterman, Martin L (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Rafique, Hassan et al. (2018). "Weakly-Convex Concave Min-Max Optimization: Provable Algorithms and Applications in Machine Learning". In: DOI: 10.48550/ARXIV.1810.02060. URL: https://arxiv.org/abs/1810.02060.

Rashid, Tabish et al. (2018). "QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning". In: *Proceedings of the Thirty-Fifth International Conference on Machine Learning*.

Russell, Stuart J et al. (2010). *Artificial intelligence a modern approach*. London.

Sanjabi, Maziar et al. (2018). "On the convergence and robustness of training gans with regularized optimal transport". In: *Advances in Neural Information Processing Systems* 31.

Sattigeri, Prasanna et al. (2018). "Fairness gan". In: *arXiv preprint arXiv:1805.09910*.

Schmid, Martin et al. (2021). "Player of Games". In: *CoRR* abs/2112.03178. arXiv: 2112.03178. URL: https://arxiv.org/abs/2112.03178.

Schwalbe, Ulrich et al. (2001). "Zermelo and the early history of game theory". In: *Games and economic behavior* 34.1, pp. 123–137.

Shapley, Lloyd S (1953). "Stochastic games". In: *Proceedings of the national academy of sciences* 39.10, pp. 1095–1100.

Shoham, Yoav et al. (2008). *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.

Silver, David et al. (2018). "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play". In: *Science* 362.6419, pp. 1140–1144.

Sion, Maurice (1958). "On general minimax theorems." In: *Pacific Journal of mathematics* 8.1, pp. 171–176.

Smallwood, Richard D. et al. (1973). "The optimal control of partially observable Markov processes over a finite horizon". In: *Operations research* 21.5, pp. 1071–1088.

Smith, Trey (2007). "Probabilistic Planning for Robotic Exploration". PhD thesis. The Robotics Institute, Carnegie Mellon University.

Smith, Trey et al. (2005). "Point-Based POMDP Algorithms: Improved Analysis and Implementation". In: *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pp. 542–549.

Sokota, Samuel et al. (2023). "Abstracting imperfect information away from two-player zero-sum games". In: *International Conference on Machine Learning*. PMLR, pp. 32169–32193.

Sondik, Edward J. (1971). *The optimal control of partially observable Markov processes*. Stanford University.

Sorin, Sylvain (2003). "Stochastic games with incomplete information". In: *Stochastic Games and applications*. Springer, pp. 375–395.

Spaan, Matthijs T.J. et al. (2005). "Perseus: Randomized point-based value iteration for POMDPs". In: *Journal of Artificial Intelligence Research* 24, pp. 195–220.

Stengel, Bernhard von (1996). "Efficient Computation of Behavior Strategies". In: *Games and Economic Behavior* 14.50, pp. 220–246.

Sutton, R.S. et al. (1998). *Reinforcement Learning: An Introduction*. Vol. 9. 5, pp. 1054–1054. DOI: 10.1109/TNN.1998.712192.

Tammelin, Oskari (2014). "Solving large imperfect information games using CFR+". In: *CoRR*. arXiv: 1407.5042.

Tan, Ming (1998). "Multi-agent Reinforcement Learning: Independent vs. Cooperative Agents". In: *Readings in Agents*. Ed. by Michael N Huhns et al. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 487–494.

Tang, Zhili et al. (2016). "Nash equilibrium and multi criterion aerodynamic optimization". In: *Journal of Computational Physics* 314, pp. 107–126.

Tsitsiklis, John N. (1984). "Problems in decentralized decision making and computation". PhD thesis. Massachusetts Institute of Technology, Cambridge, MA, USA.

Van der Pol, Elise et al. (2016). "Coordinated deep reinforcement learners for traffic light control". In: *Proceedings of learning, inference and control of multi-agent systems (at NIPS 2016)* 8, pp. 21–38.

Vinyals, Oriol et al. (2019). "Grandmaster level in StarCraft II using multi-agent reinforcement learning". In: *Nature* 575.7782, pp. 350–354.

von Neumann, John (1928). "Zur Theorie der Gesellschaftsspiele". In: *Mathematische Annalen* 100. URL: https://doi.org/10.1007/BF01448847.

Wiggers, Auke (2015). "Structure in the Value Function of Two-Player Zero-Sum Games of Incomplete Information". MA thesis. University of Amsterdam.

Wiggers, Auke et al. (2016a). "Structure in the Value Function of Two-Player Zero-Sum Games of Incomplete Information". In: *Computing Research Repository* abs/1606.06888. arXiv: 1606.06888.

— (2016b). "Structure in the Value Function of Two-player Zero-sum Games of Incomplete Information". In: *Proceedings of the Twenty-Second European Conference on Artificial Intelligence*. The Hague, The Netherlands, pp. 1628–1629. DOI: 10.3233/978-1-61499-672-9-1628.

Williams, Ronald J (1992). "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Reinforcement learning*, pp. 5–32.

Xie, Yuxuan et al. (2020). "Optimally Solving Two-Agent Decentralized POMDPs Under One-Sided Information Sharing". In: *Proceedings of the Thirty-Seventh International Conference on Machine Learning*, pp. 10473–10482.

Yan, Rui et al. (2023). "Partially Observable Stochastic Games with Neural Perception Mechanisms". In: *arXiv preprint arXiv:2310.11566*.

Zang, Yifan et al. (2023). "Sequential Cooperative Multi-Agent Reinforcement Learning". In: *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pp. 485–493.

Zermelo, Ernst (1913). "Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels". In: *Proceedings of the fifth international congress of mathematicians*. Vol. 2. Cambridge University Press Cambridge, pp. 501–504.

Zhang, Jiawei et al. (2020). "A single-loop smoothed gradient descent-ascent algorithm for nonconvex-concave min-max problems". In: *Advances in Neural Information Processing Systems* 33, pp. 7377–7389.

Zinkevich, Martin et al. (2007). "Regret Minimization in Games with Incomplete Information". In: *Advances in Neural Information Processing Systems 20*.

FOLIO ADMINISTRATIF

THESE DE L'INSA LYON, MEMBRE DE L'UNIVERSITE DE LYON

NOM : DELAGE
(avec précision du nom de jeune fille, le cas échéant)

DATE de SOUTENANCE : 28/06/2024

Prénoms : Aurélien

TITRE : Theoretical Foundations of Planning in Partially Observable Stochastic Games

NATURE : Doctorat

Numéro d'ordre : 2024ISAL0062

Ecole doctorale : InfoMaths

Spécialité : Informatique

RESUME : Une théorie récente suggère de reformuler les POSG à gain commun en des problèmes non observables via l'introduction d'une statistique suffisante appropriée, ce qui offre des leviers supplémentaires pour rechercher des plans optimaux. Montrer que le principe d'optimalité de Bellman s'applique sur le jeu non-observable permet l'application d'algorithmes efficaces conçus pour les jeux complètement observables (tels que heuristic search value iteration). Les algorithmes exploitant les leviers découverts (par exemple la division des problèmes en sous-problèmes; la généralisation des connaissances entre les sous-problèmes) offrent une garantie de convergence théorique et des résultats compétitifs sur le plan empirique. Cependant, bien que cette approche ait réussi dans des sous-classes de jeux stochastiques partiellement observables à somme nulle et à deux joueurs (zs-POSG), comment l'appliquer dans le cas général reste une question ouverte. De plus, reformuler le problème original en un problème non-observable introduit des problèmes de décision à chaque étape, dont les complexités temporelle et mémorielle deviennent prohibitives pour les jeux de grande envergure.
Dans la première contribution de ce manuscrit, nous abordons la première préoccupation et proposons pour la première fois un solveur de type heuristic search value iteration dont nous démontrons qu'il converge vers une solution ε-optimale en temps fini pour n'importe quel zs-POSG. Cela ouvre la voie à une nouvelle famille d'approches prometteuses et complémentaires à celles reposant sur la programmation linéaire ou les méthodes itératives.
Dans une deuxième contribution de ce manuscrit, nous examinons des jeux impliquant n joueurs et en supposant (i) qu'ils partagent tous la même fonction de récompense et (ii) que les joueurs sont organisés selon une structure de connaissance hiérarchique (c.-à-d. chaque agent sait ce que son subordonné sait, et ainsi de suite). Nous montrons qu'une spécialisation du schéma algorithmique point-based value iteration tire efficacement parti des leviers offerts par cette sous-classe. Ce travail ouvre la voie à de multiples extensions de la structure hiérarchique proposée tout en conservant le passage à l'échelle du schéma algorithmique proposé. Dans la dernier contribution de ce manuscrit, nous présentons une contribution connexe, bien qu'annexe, aux problèmes d'optimisation min-max avec des propriétés de continuité faibles.

MOTS-CLÉS :Théorie des jeux computationnelle, jeux à somme nulle, observabilité partielle, programmation dynamique, recherche heuristique, optimisation min-max, jeux à récompense commune, partage d'information hiérarchique.

Laboratoire (s) de recherche : CITI

Directeur de thèse: Jilles Steeve Dibangoye

Présidente de jury : Christine Solnon

Composition du jury : Christine Solnon, Aurélie Beynier, Régis Sabbadin, Bruno Zanuttini, Jilles Dibangoye, Olivier Buffet, Frédéric Koriche