New Results on the Verification of Nash Refinements for Extensive–Form Games

Nicola Gatti
Politecnico di Milano
Piazza Leonardo da Vinci 32
Milano, Italy
ngatti@elet.polimi.it

Fabio Panozzo
Politecnico di Milano
Piazza Leonardo da Vinci 32
Milano, Italy
panozzo@elet.polimi.it

ABSTRACT

The computational study of strategic interaction situations has recently deserved a lot of attention in multi–agent systems. A number of results on strategic–form games and zero–sum extensive–form games are known in the literature, while general–sum extensive–form games are not studied in depth. We focus on the problem to decide whether or not a solution is a refinement of the Nash equilibrium (NE) for extensive–form games. Refinements are needed because the NE concept is not satisfactory for this game class. While verifying whether a solution is an NE is in $\mathcal{P}$, verifying whether it is a NE refinement may be not (all the results known so far show $\mathcal{NP}$–hardness). In this paper, we provide the first positive result, showing that verifying a *sequential equilibrium* with any number of agents and a *quasi perfect equilibrium* with two agents are in $\mathcal{P}$. We show also that when the input is expressed in (non–perturbed) sequence form even the problem to verify a subgame perfect equilibrium is $\mathcal{NP}$–complete and that sequence form, if applicable, must be rethought to verify (and therefore to compute) an extensive–form perfect equilibrium.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial

General Terms

Algorithms, Economics

Keywords

Game Theory (cooperative and non-cooperative)

1. INTRODUCTION

The study of formal methods for addressing strategic interaction problems among rational agents has recently received an increasing attention in artificial intelligence and, especially, in the multi–agent system community. The aim is the development of algorithms to automate software agents and robots. Formal methods can allow one to model situations and define what is the optimal behavior an agent can have. Game theory and microeconomics represent the most elegant formal methods for strategic interaction scenarios [8]. Customarily, a scenario is modeled as a *game* in which one distinguishes the *mechanism*, defining the rules of the game (i.e., number of agents, actions available to the agents, game sequential structure, outcomes, and agents' preferences over the outcomes), from the *strategies*, defining how each single agent behaves at every decision node she acts. A *solution* of a game is strategy that is stable according to some solution concept. The basic solution concept is the Nash equilibrium (NE) constraining the strategy of each agent to be optimal given the strategies of the others.

Game theory and microeconomics provide only models and solution concepts, but they do not provide computational tools to deal with games. The development of these tools is an interesting topic, with the name of *equilibrium computation*, in computer science. The main open problems are, e.g., the verification that a solution is a given solution concept and the search for some exact or approximate solution concept. A number of computational results are known on the NE, we cite a few. Computing an exact NE [6, 7] and approximating it [4] are $\mathcal{PPAD}$–complete. $\mathcal{PPAD}$ is in $\mathcal{NP}$, it does not include $\mathcal{NP}$–complete problems unless $\mathcal{NP}$ = co–$\mathcal{NP}$, and it is not known whether $\mathcal{PPAD}$ is in $\mathcal{P}$, but it is commonly believed that it is not. Instead, the problem to verify whether or not a solution is a NE is in $\mathcal{P}$.

A number of works deal with the problem to compute a NE with general–sum strategic–form games (especially with two agents), e.g., [2, 19, 21, 22], and with the problem to solve large zero–sum extensive–form games, e.g., [11, 12]. The problem to study general–sum extensive–form games has received less attention and appears as one of the "next issues of the agenda" according to [28]. With these games, the NE is not satisfactory and refinements are needed. The most common refinements are [25]: the *subgame perfect equilibrium* (SPE) when information is perfect and the *sequential equilibrium* (SE), *quasi perfect equilibrium* (QPE), and *extensive–form perfect equilibrium* (EFPE) when information is imperfect. While the SE is the "natural" extension of the SPE to the case with imperfect information, perfect equilibria (both QPE and EFPE) pose more severe constraints, requiring the strategies to be optimal also when the agents tremble over non–optimal strategies. QPEs and EFPEs differentiate as follows: in a QPE each agent does not consider her own trembles, while in EFPEs she does.

While the verification of an NE is easy, few results are known about the verification of NE refinements for extensive–form games. The verification problem is of extraordinary

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

importance, allowing an user to verify whether a software agent is an optimizer or not. In the case this problem is intractable, we cannot certificate that the behavior of an agent is optimal and therefore the use of autonomous agents appears impractical. This would push one to resort to new approximate solution concepts. The unique results known so far in the literature are negative. More precisely, verifying whether a solution is a QPE or an EFPE is $\mathcal{NP}$–hard with three or more agents [14]. For SEs it is known only an algorithm that can be exponential in the worst case [15].

In the present paper, we provide new contributions on the NE refinement verification. More precisely, we provide two prominent positive results: both problems of verifying a SE with an arbitrary number of agents and a QPE with two agents are in $\mathcal{P}$. This supports the employment of these solution concepts in practice. In addition, we provide two negative results. The first result shows that, when the input is expressed in (non–perturbed) sequence form [27], even verifying an SPE is $\mathcal{NP}$–complete. The second result shows that the sequence form, if applicable, must be rethought to verify (and compute [9]) an EFPE with two agents (if not applicable, verifying an EFPE requires non–linear optimization and therefore the problem is not probably in $\mathcal{P}$).

2. EXTENSIVE–FORM GAMES AND EQUILIBRIUM COMPUTATION

2.1 Game definition and strategies

A *perfect–information* extensive–form game [8] is a tuple $(N, A, V, T, \iota, \rho, \chi, u)$, where: N is the set of agents ($i \in N$ denotes a generic agent), A is the set of actions ($A_i \subseteq A$ denotes the set of actions of agent i and $a \in A$ denotes a generic action), V is the set of decision nodes ($V_i \subseteq V$ denotes the set of decision nodes of i), T is the set of terminal nodes ($w \in V \cup T$ denotes a generic node and w_0 is root node), $\iota : V \to N$ returns the agent that acts at a given decision node, $\rho : V \to \wp(A)$ returns the actions available to agent $\iota(w)$ at w, $\chi : V \times A \to V \cup T$ assigns the next (decision or terminal) node to each pair w, a where a is available at w, and $u = (u_1, \ldots, u_n)$ is the set of agents' utility functions $u_i : T \to \mathbb{R}$. Games with *imperfect information* extend those with perfect information, allowing one to capture situations in which some agent cannot observe some action undertaken by the other agents. We denote by $V_{i,h}$ the h–th *information set* of agent i. An information set is a set of decision nodes such that when an agent plays at one of its nodes she cannot distinguish the node in which she is playing. For the sake of simplicity, we assume that every information set has a different index h, thus we can univocally identify an information set by h. An imperfect–information game is a tuple $(N, A, V, T, \iota, \rho, \chi, u, H)$ where $(N, A, V, T, \iota, \rho, \chi, u)$ is a perfect–information game and $H = (H_1, \ldots, H_n)$ induces a partition $V_i = \bigcup_{h \in H_i} V_{i,h}$ such that for all $w, w' \in V_{i,h}$ we have $\rho(w) = \rho(w')$. We focus on games with *perfect recall* where each agent recalls all her previous actions and her previous observations. Perfect recall poses severe constraints over the structure of the information sets, we omit their description here, not being necessary for our work, and point an interested reader to [8].

There are three representations for extensive–form games: *normal form* [26], *agent form* [17, 23], and *sequence form* [27]. In this paper, we resort to the agent and sequence forms.

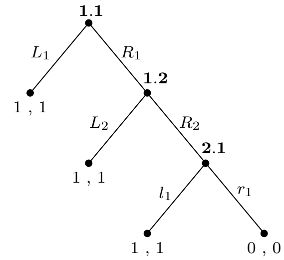

Figure 1: Example of two–agent perfect–information extensive–form game.

In the agent form, it is assumed that at each information set a different agent plays (e.g., in Fig. 1 there are three different agents, one per information set). In this way, a strategy (said *behavioral*) is represented as a probability distribution over the actions available at each single information set independently of the probability with which such an information set is reached. A behavioral strategy profile is $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_{|N|})$ where σ_i is the strategy of agent i. We denote by $\sigma_{i,a}$ the probability associated with action $a \in A_i$.

In the sequence form, a strategy is represented as a probability distribution over *sequences*. A sequence $q \in Q_i$ is a set of consecutive actions $a \in A_i$, where $Q_i \subseteq Q$ is the set of sequences of agent i and Q is the set of all the sequences. A sequence can be *terminal*, if, combined with some sequence of the opponents, it leads to a terminal node, or *non–terminal*, if it cannot lead to any terminal node for every opponents' sequence. In addition, the initial sequence of every agent, denoted by q_0, is said *empty sequence* and, given sequence $q \in Q_i$ leading to some information set $h \in H_i$, we say that q' *extends* q (denoted by $q' = q|a$) if the last action a' of q' (denoted by $a(q)$) belongs to $\rho(w)$ with $w \in V_{i,h}$. We denote a sequence–form strategy profile as a vector by $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_{|N|}]$ where $\mathbf{x}_i$ is the strategy of agent i and we denote by $x_{i,q}$ the probability associated with sequence $q \in Q_i$. Well defined strategies are such that, for every information set $h \in H_i$, the probability $x_{i,q}$ assigned to the sequence q leading to h is equal to the sum of the probabilities $x_{i,q'}$s where q' extends q at h. Sequence form constraints can be conveniently described as $F_i \mathbf{x}_i = \mathbf{f}_i$, where F_i is an opportune matrix and $\mathbf{f}_i$ is an opportune vector. The agent i's utility is represented as a sparse multi–dimensional array, denoted by U_i, specifying the value associated to every combination of terminal sequences of all the agents. The size of the sequence form representation is linear in the size of the game tree.

A sequence–form strategy $\mathbf{x}_i$ is equivalent to a number (precisely, a compact set) of behavioral strategies σ_i and the relationship is non–linear. More precisely, given an information set $h \in H_i$ and called $q \in Q_i$ the sequence leading to h, the behavioral strategy $\sigma_{i,a}$ related to the actions $a \in \rho(w)$ with $w \in V_{i,h}$ and $q' = q|a$ is $\sigma_{i,a(q')} = \frac{x_{i,q'}}{x_{i,q}}$ if $x_{i,q} > 0$ and 0 otherwise. The two representations have different degrees of expressiveness, e.g., sequence–form strategies, differently from behavioral ones, do not specify the actions that would be played at information sets reached with zero probability.

Several solution concepts (see below) are based on the idea of *perturbed strategies*. Call $l_{i,a}(\epsilon) > 0$ the *perturbation* (in terms of probability) over action $a \in A_i$ such that $\lim_{\epsilon \to 0} l_{i,a}(\epsilon) = 0$ and ϵ is a positive value. We denote by $\mathbf{l}_i(\epsilon)$ the vectors of the perturbations over all the agent i's actions. A perturbed behavioral strategy profile $\boldsymbol{\sigma}(\epsilon)$ of $\boldsymbol{\sigma}$ is a fully mixed strategy where $\sigma_{i,a} \geq l_{i,a}(\epsilon)$ for all $a \in A_i$ and $\lim_{\epsilon \to 0} \boldsymbol{\sigma}(\epsilon) = \boldsymbol{\sigma}$. Analogously, the idea of perturbation can be applied to the sequence form. In this case, we denote by $\mathbf{x}_i(\epsilon)$ the perturbed sequence form strategy and by $x_{i,q}(\epsilon)$ the perturbed strategy over $q \in Q_i$. The result in [1] shows that we can deal with perturbations $l_{i,a}(\epsilon)$ defined as polynomials in ϵ keeping ϵ a symbolic parameter by resorting to the concept of lexico positiveness (see Appendix A). In our work, we denote by $\mathbf{x}_i(\epsilon^k)$ the coefficients of ϵ^k in $\mathbf{x}_i(\epsilon)$ and $x_{i,q}(\epsilon^k)$ the coefficients of ϵ^k in $x_{i,q}(\epsilon)$.

2.2 Solution concepts

It is well known that the concept of NE is not satisfactory for extensive–form games, allowing agents to play non–credible threats. The concept of SPE refines the concept of NE, constraining a strategy profile to be a NE in every subgame [8], where a subgame is a portion of the game tree defined as follows: it has a root and for every node $w \in V_{i,h}$ belonging to the subgame the whole information set $V_{i,h}$ belongs to the subgame. (A SPE can be easily found by applying *backward induction* [8].) The concept of SPE is satisfactory with perfect–information games, while it is not when information is imperfect. The "natural" extension of the SPE to situations with imperfect information is the SE [16]. We denote by $\mu_i = (\mu_{i,w})$ for every $w \in V_i$ the *beliefs* of agent i where $\mu_{i,w}$ is the probability with which agent i believes to be at node $w \in V_{i,h}$ when she plays at information set h. We denote by $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{|N|})$ the profile of beliefs. An *assessment* is a pair $(\boldsymbol{\mu}, \boldsymbol{\sigma})$. An SE is an assessment $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ such that: every σ_i is *sequentially optimal* (in the sense of backward induction) with respect to μ_i, and every μ_i is *consistent* (in the sense of Kreps and Wilson) with respect to σ_{-i}. Consistency of $\boldsymbol{\mu}$ with respect to $\boldsymbol{\sigma}$ requires that there exists a perturbed strategy profile $\boldsymbol{\sigma}(\epsilon)$ of $\boldsymbol{\sigma}$ such that, if $\boldsymbol{\mu}(\epsilon)$ the sequence of beliefs derived from $\boldsymbol{\sigma}(\epsilon)$ by using the Bayes rule, $\lim_{\epsilon \to 0} \boldsymbol{\mu}(\epsilon) = \boldsymbol{\mu}$. With perfect information every SPE is also an SE and *vice versa*. Instead, when information is imperfect, the SEs constitute a subset of the SPEs.

EXAMPLE 2.1. *Consider the game in Fig. 1. The pure strategy SEs (and SPEs) are:* $(\sigma_{1,L_1} = 1, \sigma_{1,L_2} = 1, \sigma_{2,l_1} = 1)$, $(\sigma_{1,L_1} = 1, \sigma_{1,R_2} = 1, \sigma_{2,l_1} = 1)$, $(\sigma_{1,R_1} = 1, \sigma_{1,L_2} = 1, \sigma_{2,l_1} = 1), (\sigma_{1,R_1} = 1, \sigma_{1,R_2} = 1, \sigma_{2,l_1} = 1)$.

The idea of *perfection*, introduced by Selten in [23], is strictly correlated to the idea of perturbed strategy. Basically, a strategy profile is perfect when it is optimal even with perturbations over the strategies. The rationale behind perturbations is that agents do not perfectly play their optimal strategy, but they tremble with a very small probability over non–optimal strategies. The application of perturbation to the three (normal, agent, sequence) forms of a game may lead to different concepts of equilibria.

A strategy profile $\boldsymbol{\sigma}$ is a QPE if there exists a perturbed strategy profile $\boldsymbol{\sigma}(\epsilon)$ of $\boldsymbol{\sigma}$ such that $\sigma_{i,a}(\epsilon) \geq l_{i,a}(\epsilon)$ and every σ_i is a best response to $\sigma_{-i}(\epsilon)$ for every $\epsilon \leq \overline{\epsilon}$ for some $\overline{\epsilon} > 0$ [24]. In a QPE every agent takes into account the opponents' trembles, but not own. For every combination of $\mathbf{l}_i(\epsilon)$ there is a potentially different QPE. The authors show in [20] that quasi perfection can be captured by using a specific class of perturbations with the sequence form constraining that for every pair of sequences $q, q' \in Q_i$ with $q = q'|a(q)$ the minimum degree of k such that $l_{i,q}(\epsilon^k)$ is strictly positive is strictly smaller than the minimum degree of k such that $l_{i,q'}(\epsilon^k)$ is strictly positive, formally, $\min_{l_{i,q}(\epsilon^k)>0} \{k\} > \min_{l_{i,q'}(\epsilon^k)>0} \{k\}$.

Other solution concepts are the *normal–form perfect equilibrium* (NFPE; when perturbations are over normal–form strategies, but it is not a satisfactory solution concept) and the *extensive–form perfect equilibrium* (EFPE; it is defined as the QPE except that an agent takes into account her own trembles in addition to those of the opponents).

EXAMPLE 2.2. *Consider the game represented in Fig. 1. The pure strategy QPEs are:* $(\sigma_{1,L_1} = 1, \sigma_{1,L_2} = 1, \sigma_{2,l_1} = 1), (\sigma_{1,R_1} = 1, \sigma_{1,L_2} = 1, \sigma_{2,l_1} = 1)$. *Notice that* $(\sigma_{1,R_1} = 1, \sigma_{1,R_2} = 1, \sigma_{2,l_2} = 1)$ $(\sigma_{1,R_1} = 1, \sigma_{1,R_2} = 1, \sigma_{2,l_1} = 1)$, *that is an SE, is not a QPE. This is because, accounting for any perturbed* $\sigma_{2,l_1}(\epsilon)$*, the utility expected by agent 1 from making action* R_2 *(i.e.,* $\sigma_{2,l_1}(\epsilon) < 1$*) is strictly smaller than the utility she expects from making action* L_2 *(i.e., 1). The unique EFPE, when agents account for own trembles, is* $(\sigma_{1,L_1} = 1, \sigma_{1,L_2} = 1, \sigma_{2,l_1} = 1)$.

2.3 Known computational results

The sequence form is the most efficient representation to compute an NE (normal form is exponentially larger, while agent form poses highly non–linear constraints over the agents' best response optimization problems). The main results on the computation of an NE are with two agents. The problem to search for an NE is formulated as a linear–complementarity problem (LCP) and solved by employing the Lemke's algorithm [18], a generalization of the Lemke–Howson algorithm [19]. The problem to verify whether a strategy profile, both in sequence form and agent form (in this case deriving the corresponding sequence form strategies), is an NE can be easily solved in polynomial time by checking whether or not the constraints are satisfied.

The computation and verification problems for an SE are open [14] and it is not known whether it is possible to address them in sequence form or, as it is commonly believed, it is necessary the agent form. The unique result on the verification of an SE is provided in [15]. They propose finite–step algorithm to verify whether an assessment is an SE, but, as they state it, the number of steps accomplished by the algorithm can be exponential in the worst case. A slightly different problem is studied in [14], where the authors show in that with three or more agents, verifying whether there is an SE with a given strategy is $\mathcal{NP}$–hard.

In [20] the authors use the Lemke's algorithm applied to the sequence form with perturbations $\mathbf{l}_1(\epsilon)$, $\mathbf{l}_2(\epsilon)$ with $l_{i,q}(\epsilon) = \epsilon^{|q|}$ where $|q|$ is the length of sequence q to compute a QPE when agents are two (details are in Appendix B). This places that such a problem in the $\mathcal{PPAD}$ class. Instead, the verification problem is currently open with two agents. Differently from the verification of an NE, verifying whether a strategy profile is a QPE is a search problem in which perturbations $\mathbf{l}_1(\epsilon), \mathbf{l}_2(\epsilon)$ need to be found to satisfy the QPE constraints. With three or more agents the verification problem is shown to be $\mathcal{NP}$–hard [14].

Other known results on the equilibrium verification problem are: the verification of a NFPE with two agents is in $\mathcal{P}$, while the verification of a NFPE and of an EFPE with three or more agents is $\mathcal{NP}$–hard [14]. No result is known for EFPE with two agents and it is not known even whether or not the sequence form can be employed.

3. VERIFICATION WITH AGENT FORM

We report a positive (tractable) result on the verification of an SE by providing an algorithm that works with the agent form. This is possible since we do not need to use perturbations. When instead perturbations must be considered, as for the verification of a QPE, working with the agent form appears hard since the verification problem is equivalent to the problem to search for an appropriate perturbation over the behavioral strategies and this problem is highly non–linear because the perturbations at different information sets would be multiplied. We state the following theorem, whose proof provides a polynomial time algorithm based on linear programming.

THEOREM 3.1. *Given a game with an arbitrary number of agents, it is in $\mathcal{P}$ the problem to decide whether or not an assessment $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ is an SE.*

Proof. This decision problem requires one to verify two correlated properties: sequential rationality and consistency. Sequential rationality of $\boldsymbol{\sigma}$ can be easily verified by backward induction on the basis of $\boldsymbol{\mu}$. This task requires a number of maximizations that is linear in the size of the game, and each single maximization is over a number of actions that is linear in the size of the game. Verifying consistency of $\boldsymbol{\mu}$ is an harder task. By definition, it requires one to find a fully mixed perturbed strategy profile $\boldsymbol{\sigma}(\epsilon)$ such that the beliefs $\boldsymbol{\mu}(\epsilon)$ derived from $\boldsymbol{\sigma}(\epsilon)$ by Bayes rule converges to $\boldsymbol{\mu}$ as $\epsilon \to 0$.

The problem to find a $\boldsymbol{\sigma}(\epsilon)$ can be solved by resorting to the concept of *b–labeling* provided by Kreps and Wilson in [16]. A b–labeling for an assessment $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ is a function $\lambda : A \to \mathbb{N}$ that assigns a label (expressed as a non–negative integer number) to all the actions $a \in A$ such that:

$$\begin{aligned} &\lambda_a = 0 && \iff \sigma_{i,a} > 0 && \forall a \in A_i, i \in N \\ &\sum_{a \to w} \lambda_a = \arg\min_{w'} \sum_{a \to w'} \lambda_a && \iff \mu_{i,w} > 0 && \forall w, w' \in V_{i,h}, i \in N, h \in H_i \end{aligned}$$

We use the symbol '$a \to w$' to denote all the actions $a \in A$ leading to node w from the root node w_0. Given a b–labeling, we can define a fully mixed strategy profile $\overline{\boldsymbol{\sigma}}(\epsilon)$ as:

$$\overline{\sigma}_{i,a}(\epsilon) = \begin{cases} c(\epsilon, h, a) \cdot \sigma_{i,a} & \text{if } \sigma_{i,a} > 0 \\ c(\epsilon, h, a) \cdot \epsilon^{\lambda_a} & \text{otherwise} \end{cases}$$

where a is an action played by some agent at information set h, and $c(\epsilon, h, a)$ is the appropriate normalizing constant. Kreps and Wilson proved that $\boldsymbol{\mu}$ is consistent to $\boldsymbol{\sigma}$ if and only if the above $\overline{\boldsymbol{\sigma}}(\epsilon)$ is well defined (i.e., a b–labeling exists). We show below that the problem to search for a b–labeling can be accomplished in polynomial time.

The bottom line of proof is the following. First, we formulate the problem to find a b–labeling as a linear integer mathematical program [29], second, we show that the coefficient matrix associated with the mathematical program in standard form is *totally unimodular* [3] and the right hand is integer. Therefore, the integer mathematical program can be solved in polynomial time, all the basic solutions of the relaxed continuous mathematical program being integer.

The integer mathematical programming formulation is (γ and ν denote auxiliary variables, while s and t denote slack variables):

$$\begin{aligned} &\min \sum_{a \in A} \lambda_a && && (1) \\ &\lambda_a = 0 && \forall a \in A, \sigma_{i,a} > 0, i \in N && (2) \\ &\lambda_a - s_a = 1 && \forall a \in A, \sigma_{i,a} = 0, i \in N && (3) \\ &\gamma_{w_0} = 0 && && (4) \\ &\gamma_{w'} + \lambda_a - \gamma_w = 0 && \forall w, w' \in V, a \in A, w = \chi(w', a) && (5) \\ &\gamma_w - \nu_h = 0 && \forall h \in H_i, w \in V_{i,h}, i \in N, \mu_{i,w} > 0 && (6) \\ &\gamma_w - \nu_h - t_n = 1 && \forall h \in V_{i,h}, i \in N, \mu_{i,w} = 0 && (7) \\ &\lambda_a \in \mathbb{N} && \forall a \in A && (8) \\ &s_a \geq 0 && \forall a \in A && (9) \\ &t_w \geq 0 && \forall w \in V && (10) \end{aligned}$$

Constraints (2) force labels λ_a of actions a played with positive probability to be equal to zero; constraints (3) force labels λ_a of actions a played with zero probability to be at least one; constraint (4) assigns a value of zero to auxiliary variable γ_{w_0} associated with root node w_0; constraints (5) assign the auxiliary variable γ_w associated with node w a value equal to the sum of the value of the parent node w' and the label of the action connecting w' to w; constraints (6) force the values of all the γ_ws associated with the nodes ws with $\mu_{i,w} > 0$ belonging to the same information set to be same (i.e., ν_h); constraints (7) force the other nodes w (those with $\mu_{i,w} = 0$) to have a value γ_w strictly larger than the minimum value of the information set (i.e., ν_h); constraints (8)–(10) fix the domains of the variables (notice that, with these domains, all the variables have non–negative values).

The above constraints can be expressed as $M\mathbf{y} = \mathbf{b}$ with $\mathbf{y} \geq 0$ and $\boldsymbol{\lambda}$ constrained to have non–negative integer values, where:

$$M = \left[\begin{array}{c|c|c|c|c} C & 0 & 0 & 0 & 0 \\ \hline C' & 0 & 0 & -I & 0 \\ \hline D & E & 0 & 0 & 0 \\ \hline 0 & G & K & 0 & 0 \\ \hline 0 & G' & K' & 0 & -I \end{array}\right], \mathbf{y} = \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\gamma} \\ \boldsymbol{\nu} \\ \mathbf{s} \\ \mathbf{t} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix}$$

such that $C\boldsymbol{\lambda} = 0$ codes constraints (2), $C'\boldsymbol{\lambda} - I\mathbf{s} = 1$ codes constraints (3), $D\boldsymbol{\lambda} + E\boldsymbol{\gamma} = 0$ codes constraints (4) and (5), $G\boldsymbol{\gamma} + K\boldsymbol{\nu} = 0$ codes constraints (6), and $G'\boldsymbol{\gamma} + K'\boldsymbol{\nu} - I\mathbf{t} = 0$ codes constraints (7). The above submatrices have the following properties: C, C', G, and G' have one 1 per row and zero or one 1 per column; D is composed of a row of zero and identity matrix I; E has one '1' in the first row and one '1' and one '−1' in all the other rows; K presents one '−1' per row.

Given that a matrix M is totally unimodular if and only if the transpose M^T is totally unimodular [3], we can restate the theorem of Ghoulia–Houri [10] as: M is totally unimodular if and only if for every subset M' of columns of M it is possible to find a partition of columns $\{M'_1, M'_2\}$ such that (call m'_{kj} a generic element of matrices M'_1 and M'_2):

$$\forall k \left(\sum_{j, m'_{kj} \in M'_1} m'_{kj} - \sum_{j, m'_{kj} \in M'_2} m'_{kj} \right) \in \{-1, 0, 1\} \qquad (11)$$

To prove that this condition holds for M, call Λ_i the k–th block of rows of M (from the top to the bottom) and call Δ_j the j–th block of columns of M (from the left to the right).

At first, we notice that we can remove the last two blocks of columns Δ_4 and Δ_5. Indeed, if constraints (11) are satisfied limiting to the first three blocks of columns (i.e., considering only the elements m'_{ij} belonging to $M'_k \cap \{\Delta_1 \cup \Delta_2 \cup$

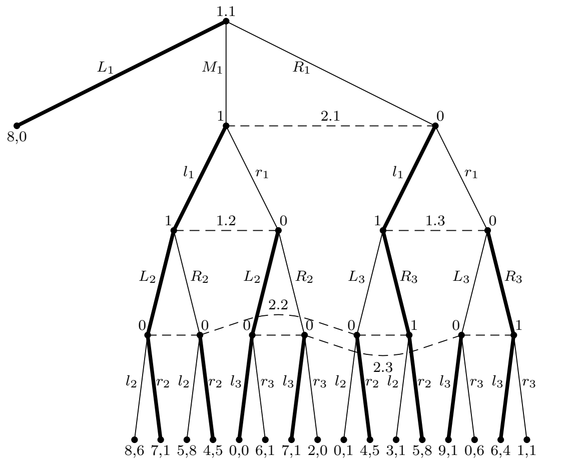

Figure 2: Example of assessment $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ where $\boldsymbol{\sigma}$ is sequentially rational, but $\boldsymbol{\mu}$ is not consistent ($\boldsymbol{\sigma}$ is represented by using bold lines to denote actions played with positive probability and $\boldsymbol{\mu}$ is represented reporting the beliefs close to the nodes of each information set). No b–labeling exists because the constraints due to information set $h = \mathbf{2.1}$ (i.e., $\lambda_{R_1} \geq \lambda_{M_1} + 1$) and due to information set $h = \mathbf{2.2}$ (i.e., $\lambda_{M_1} \geq \lambda_{R_1} + 1$) cannot be satisfied simultaneously.

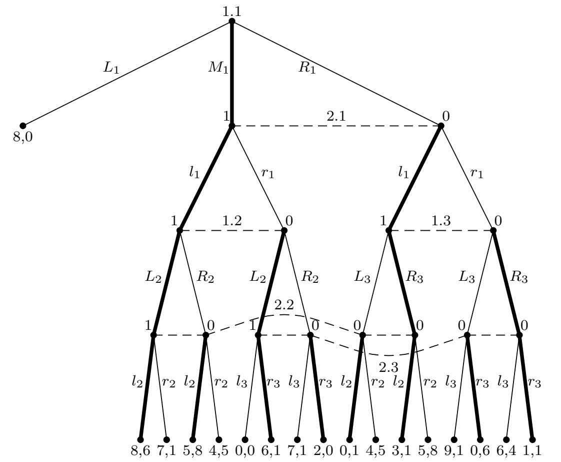

Figure 3: Example of assessment $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ where $\boldsymbol{\sigma}$ is sequentially rational and $\boldsymbol{\mu}$ is consistent ($\boldsymbol{\sigma}$ is represented by using bold lines to denote actions played with positive probability and $\boldsymbol{\mu}$ is represented reporting the beliefs close to the nodes of each information set). The b–labeling is: $\lambda_a = 1$ for all $a \in A_i$ with $\sigma_{i,a} = 0$.

$\Delta_3\}$), then we can always put columns of M' belonging to Δ_4 or Δ_5 into M'_1 or M'_2 to make constraints (11) satisfied along all the columns. We build M'_1 and M'_2 as follows. Put all the columns of M' belonging to Δ_2 or Δ_3 into M'_1. It can be easily seen that constraints (11) for the rows belonging to Λ_4 and Λ_5 are satisfied (the sum of elements belongs to $\{-1, 0, 1\}$) independently of whether the columns of M' belonging to Δ_1 are put into M'_1 or M'_2. Consider the columns of M'_1 belonging to Δ_2: the sum of the elements of rows belonging to Λ_3 can be $\{-1, 0, 1\}$. It can be easily seen that, D having no more than one '1' per column, we can always put the columns of Δ_1 into M'_1 or M'_2 to make constraints (11) satisfied along the rows belonging to Λ_3. Finally, we observe that constraints (11) are always satisfied along the rows belonging to Λ_1 and Λ_2. Thus, M is totally unimodular and, $\mathbf{b}$ being integer, a b–labeling, if it exists, can be found by linear (continuous) mathematical programming. □

We provide two examples to which we apply the algorithm discussed in the proof of Theorem 3.1.

EXAMPLE 3.2. *Consider the game depicted in Fig. 2 and the assessment $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ where $\boldsymbol{\sigma} = (\sigma_{1,L_1} = 1, \sigma_{1,L_2} = 1, \sigma_{1,R_3} = 1, \sigma_{2,l_1} = 1, \sigma_{2,r_2} = 1, \sigma_{2,l_3} = 1)$ and beliefs $\boldsymbol{\mu}$ are reported in the figure aside the corresponding nodes. No b–labeling exists because the constraints due to information set $h = \mathbf{2.1}$ (i.e., $\lambda_{R_1} \geq \lambda_{M_1} + 1$) and due to information set $h = \mathbf{2.2}$ (i.e., $\lambda_{M_1} \geq \lambda_{R_1} + 1$) cannot be satisfied simultaneously. Therefore, the assessment is not an SE.*

EXAMPLE 3.3. *Consider the game depicted in Fig. 3 and the assessment $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ where $\boldsymbol{\sigma} = (\sigma_{1,M_1} = 1, \sigma_{1,L_2} = 1, \sigma_{1,R_3} = 1, \sigma_{2,l_1} = 1, \sigma_{2,l_2} = 1, \sigma_{2,r_3} = 1)$ and beliefs $\boldsymbol{\mu}$ are reported in the figure aside the corresponding nodes. A b–labeling is: $\lambda_a = 1$ for all $a \in A_i$ with $\sigma_{i,a} = 0$. Therefore, the assessment is an SE.*

4. VERIFICATION WITH SEQUENCE FORM

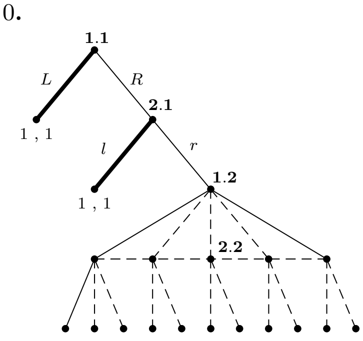

Figure 4: Game used in the proof of Theorem 4.1.

We provide some verification results when we use the sequence form. Initially, we report a negative result even for the SPE when the input strategies of the verification problem are expressed in (non–perturbed) sequence form.

THEOREM 4.1. *Given a game with two agents, it is $\mathcal{NP}$–complete the problem to decide whether or not non–fully perturbed strategies $\mathbf{x}_1, \mathbf{x}_2$ constitute an SPE.*

Proof. We reduce this to the problem to decide whether there is a NE with some property. This problem was shown $\mathcal{NP}$–complete in [5]. The reduction is based on the game tree depicted in Fig. 4. The subgame starting with information set $h = \mathbf{1.2}$ and including information set $h = \mathbf{2.2}$ is a generic general–sum strategic–form game Γ with two agents. Consider the following non–perturbed strategies (in the case strategies are perturbed, but not fully mixed, the proof is analogous): $\mathbf{x}_1$ prescribes that action L is played with a probability of one, and $\mathbf{x}_2$ prescribes that action l is played with a probability of one. For all the other actions, $\mathbf{x}_1, \mathbf{x}_2$ prescribe a probability of zero. Strategies $\mathbf{x}_1, \mathbf{x}_2$ constitute an SPE if and only if the subgame starting at $h = \mathbf{1.2}$ admits an NE that provides agent 2 an expected utility smaller than 1. Since $\mathbf{x}_1, \mathbf{x}_2$ prescribe a probability of zero in Γ, they do not pose any constrain over the problem to search for an NE for Γ providing agent 2 with no more than 1. Hence, our problem reduces to the problem to decide whether there is a NE with some property. □

The above theorem can be easily extended showing that the verification of an SE (with an arbitrary number of agents) is $\mathcal{NP}$–complete when the input is in sequence form. Furthermore, it is trivial to show that, when the input to the verification problem is a fully perturbed sequence form strategy profile, we have positive results. Indeed, given a fully mixed perturbed sequence form strategy, we can always derive an equivalent perturbed behavioral strategy and from this a non–perturbed behavioral strategy. Thus, we can apply the positive results with agent form.

Now, we consider the problem to verify a QPE. While this problem appears hard by working with the agent form, we have a tractable result with the sequence form.

THEOREM 4.2. *Given a game with two agents, it is in $\mathcal{P}$ the problem to decide whether or not a strategy profile $\boldsymbol{\sigma}$ is a QPE.*

Proof. In order to verify whether a strategy profile $\sigma = (\sigma_1, \sigma_2)$ is a QPE, we need to verify:

- the existence of a perturbed $\sigma_1(\epsilon)$ such that $\sigma_1(\epsilon) \to \sigma_1$ as $\epsilon \to 0$ and σ_2 is a best response to $\sigma_1(\epsilon)$,
- the existence of a perturbed $\sigma_2(\epsilon)$ such that $\sigma_2(\epsilon) \to \sigma_2$ as $\epsilon \to 0$ and σ_1 is a best response to $\sigma_2(\epsilon)$.

This is equivalent to verify the existence of a *lexicographic belief structure* according to [1, 13]. Since characterization of a QPE can be accomplished in sequence form without resorting the agent form we can formulate our problem with the sequence form exploiting the LCP formulation discussed in Appendix B. We can formulate the search for $\sigma_1(\epsilon)$ and $\sigma_2(\epsilon)$ as the search for two perturbed strategies $\mathbf{x}_1(\epsilon)$ and $\mathbf{x}_2(\epsilon)$ such that the following constraints hold (the constraints over $\mathbf{x}_2(\epsilon)$ are analogous):

$$F_1\mathbf{x}_1(\epsilon) = \mathbf{f}_1 \quad (12)$$

$$\mathbf{x}_1(\epsilon) >_L \mathbf{0} \quad (13)$$

$$F_2^T\mathbf{v}_2(\epsilon) - U_2^T\mathbf{x}_1(\epsilon) \geq_L \mathbf{0} \quad (14)$$

$$(F_2^T\mathbf{v}_2(\epsilon) - U_2^T\mathbf{x}_1(\epsilon))_q = 0 \qquad \forall q \in Q_2, \sigma_{2,a(q)} > 0 \quad (15)$$

$$\min_{x_{1,q}(\epsilon^k)>0} k < \min_{x_{1,q'}(\epsilon^k)>0} k \qquad \begin{array}{l}\forall a(q) \in \rho(w), a(q') \in \rho(w'),\\ w, w' \in H_{1,h}, h \in H_1,\\ \sigma_{1,a(q)} > 0, \sigma_{1,a(q')} = 0\end{array} \quad (16)$$

where constraints (12) state that the strategy is well defined according to sequence form definition; constraints (13) state that the strategy is fully mixed ($>_L$ means 'lexico–positive'); constraints (14) are the dual best response constraints; constraints (15) state that, if the behavioral strategy $\sigma_{2,a(q)}$ has strictly positive value for the last action of sequence q, then the best response constraint associated with q must hold with equality; constraints (16) provide a hierarchical structure over the lexicographic perturbation of $\mathbf{x}_1(\epsilon)$ forcing in every information set that the minimum degree k, such that $x_{1,q}(\epsilon^k)$ is positive when the last action action $a(q)$ of q is played with $\sigma_{1,a(q)} > 0$, is strictly lower than the minimum degree k' related to sequences q's whose last action is played with $\sigma_{1,a(q')} = 0$.

The above feasibility problem can be solved iteratively as follows. At each iteration k, we find the values of $\mathbf{x}_1(\epsilon^k)$. Each iteration can be formulated as a linear mathematical programming problem. Iteration $k = 0$ requires the resolution of the following mathematical program:

$$F_1\mathbf{x}_1(\epsilon^0) = \mathbf{f}_1 \quad (17)$$

$$x_{1,q}(\epsilon^0) \geq 0 \qquad \forall a(q) \in A_1, \sigma_{1,a(q)} > 0 \quad (18)$$

$$x_{1,q}(\epsilon^0) = 0 \qquad \forall a(q) \in A_1, \sigma_{1,a(q)} = 0 \quad (19)$$

$$F_2^T\mathbf{v}_2(\epsilon^0) - U_2^T\mathbf{x}_1(\epsilon^0) \geq \mathbf{0} \quad (20)$$

$$(F_2^T\mathbf{v}_2(\epsilon^0) - U_2^T\mathbf{x}_1(\epsilon^0))_q = 0 \qquad \forall q \in Q_2, \sigma_{2,a(q)} > 0 \quad (21)$$

where constraints (17) are analogous to (12); constraints (18) and (19) correspond to (16); constraints (20) and (21) correspond to (14) and (15). The above program is feasible if $\boldsymbol{\sigma}$ is a Nash equilibrium. Therefore, if the above program is infeasible, then the algorithm stops and $\boldsymbol{\sigma}$ is not a QPE.

From $k = 1$ on, the mathematical program to solve is:

$$\max \sum_{\forall k'<k, x_{1,q}(\epsilon^{k'})=0} x_{1,q}(\epsilon^k) \quad (22)$$

$$F_1\mathbf{x}_1(\epsilon^k) = 0 \quad (23)$$

$$x_{1,q}(\epsilon^k) \geq 0 \qquad \forall q \in Q_1, x_{1,q}(\epsilon^{k'}) = 0, k' < k \quad (24)$$

$$x_{1,q}(\epsilon^k) \leq 1 \qquad \forall q \in Q_1 \quad (25)$$

$$x_{1,q}(\epsilon^k) = 0 \qquad \begin{array}{l}\forall q, q' \in Q_1, w, w' \in V_{1,h},\\ a(q) \in \rho(w), a(q') \in \rho(w'),\\ \sigma_{1,a(q)} = 0, \sigma_{1,a(q')} > 0,\\ x_{1,q'}(\epsilon^{k'}) = 0, k' < k, h \in H_1\end{array} \quad (26)$$

$$(F_2^T\mathbf{v}_2(\epsilon^k) - U_2^T\mathbf{x}_1(\epsilon^k))_q \geq 0 \qquad \begin{array}{l}\forall q \in Q_2, (F_2^T\mathbf{v}_2(\epsilon^{k'}) -\\ U_2^T\mathbf{x}_1(\epsilon^{k'}))_q = 0, k' < k\end{array} \quad (27)$$

$$(F_2^T\mathbf{v}_2(\epsilon^k) - U_2^T\mathbf{x}_1(\epsilon^k))_q = 0 \qquad \forall q \in Q_2, \sigma_{2,a(q)} > 0 \quad (28)$$

where constraints (23) grant the strategy to be well defined; constraints (24) grant that $\mathbf{x}_1(\epsilon)$ is lexico–positive; constraints (25) pose an upper bound of 1 over the coefficients of ϵ^k (this value does not affect the feasibility of the problem); constraints (26) force constraints (16); constraints (27) and (28) force constraints (14) and (15), respectively. The objective function aims at maximizing the sum of the coefficients $x_{1,q}(\epsilon^k)$ such that $x_{1,q}(\epsilon^{k'}) = 0$ for all $k' < k$.

The algorithm stops either when $\mathbf{x}_1(\epsilon)$ is strictly lexico–positive or when the objective function is 0. In the latter case, it is not possible to find any strictly lexico–positive $\mathbf{x}_1(\epsilon)$ that satisfies the above constraints and therefore $\boldsymbol{\sigma}$ is not a QPE. Otherwise, if there are strictly lexico–positive $\mathbf{x}_1(\epsilon)$ and $\mathbf{x}_2(\epsilon)$, $\boldsymbol{\sigma}$ is a QPE.

We discuss the completeness of the algorithm. Note that the constraints at iteration k depend on the solutions of the optimization problems at the previous iterations. Given that a linear optimization problem can admit different optimal solutions, we have that the possible paths the algorithm can follow are different. However, it can be observed that the set of constraints strictly relaxes from iteration k to k'. Therefore, for all the paths the algorithm can follow, the algorithm always terminates with the same outcome in terms of existence or non–existence of a strictly lexico–positive $\mathbf{x}_1(\epsilon)$ (notice that in the case of existence, different paths may lead to different strictly lexico–positive strategies).

Finally, we show that the number of iteration is in the worst case linear in the size of the game. At each iteration k, either some $x_{1,q}(\epsilon^k)$ that is zero for every $k' < k$ becomes strictly positive or the algorithm stops with failure.

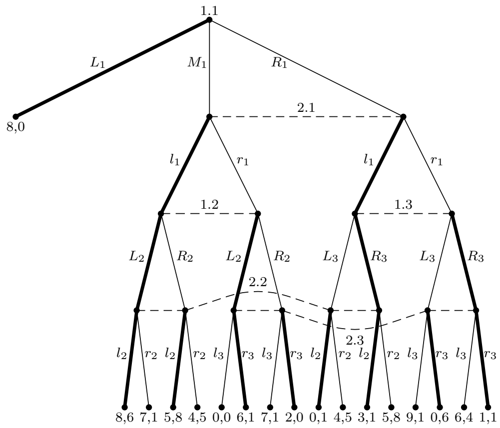

Figure 5: Example of strategy profile $\boldsymbol{\sigma}$ expressed in behavioral strategies that is a quasi perfect equilibrium ($\boldsymbol{\sigma}$ is represented by using bold lines to denote actions played with positive probability).

In the worst case, only one sequence becomes strictly lexico–positive per iteration and therefore the number of iteration is equal to the number of sequences. Thus, linear mathematical programming being polynomial time, the theorem is proved. □

We provide two examples to which we apply the algorithm described in the proof of Theorem 4.2.

EXAMPLE 4.3. *Consider the game depicted in Fig. 3 and the strategy profile* $\boldsymbol{\sigma} = (\sigma_{1,M_1} = 1, \sigma_{1,L_2} = 1, \sigma_{1,R_3} = 1, \sigma_{2,l_1} = 1, \sigma_{2,l_2} = 1, \sigma_{2,r_3} = 1)$*. (As shown in Example 3.3, it is an SE.) We check whether or not it is a QPE. From the application of the algorithm we provide in the proof of Theorem 4.2, we obtain the following* $\mathbf{x}_2(\epsilon)$*:*

	l_1	r_1	l_2	r_2	l_3	r_3
ϵ^0	1	0	1	0	0	0
ϵ^1	0	0	0	0	0	0

At iteration 1, the algorithm stops because the objective function is zero. Indeed, the algorithm cannot put a positive value on r_2 *without violating the constraints of best response of agent 1. As a result, no fully mixed* $\sigma_2(\epsilon)$ *makes* σ_1 *to be a best response and, therefore,* $\boldsymbol{\sigma}$ *is not a QPE.*

EXAMPLE 4.4. *Consider the game depicted in Fig. 5 and the strategy profile* $\boldsymbol{\sigma} = (\sigma_{1,L_1} = 1, \sigma_{1,L_2} = 1, \sigma_{1,R_3} = 1, \sigma_{2,l_1} = 1, \sigma_{2,l_2} = 1, \sigma_{2,r_3} = 1)$*. We check whether or not it is a QPE. From the application of the algorithm we provide in the proof of Theorem 4.2, we obtain the following fully mixed* $\mathbf{x}_2(\epsilon)$*:*

	l_1	r_1	l_2	r_2	l_3	r_3
ϵ^0	1	0	1	0	0	0
ϵ^1	−1	1	−2	1	0	1
ϵ^2	0	0	0	0	1	−1

and the following fully mixed $\mathbf{x}_1(\epsilon)$*:*

	L_1	M_1	R_1	L_2	R_2	L_3	R_3
ϵ^0	1	0	0	0	0	0	0
ϵ^1	−2	1	1	1	0	0	1
ϵ^2	0	0	0	−1	1	1	−1

therefore $\boldsymbol{\sigma}$ *is a QPE.*

Finally, we show that the employment of sequence form, when each agent takes into account also her own perturbations, presents several problems to verify an EFPE.

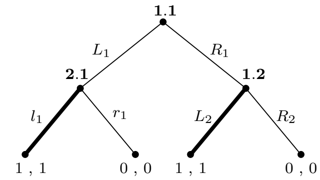

Figure 6: Game used in the proof of Proposition 4.5.

PROPOSITION 4.5. *The best response optimization problem with the sequence form, when each agent takes into account also her own perturbations, cannot be used to verify an EFPE.*

Proof. Consider the game tree depicted in Fig. 6. At $h = \mathbf{1.2}$, the unique optimal strategy is $\sigma_{1,L_2} = 1$. Analogously, at $h = \mathbf{2.1}$, the unique optimal strategy is $\sigma_{2,l_1} = 1$. At $h = \mathbf{1.1}$, L_1 and/or R_1 can be optimal on the basis of the perturbation at the two subgames. For instance, with a perturbation over behavioral strategies such that $l_{2,r_1} = \epsilon^2$ and $l_{1,R_2} = \epsilon$, L_1 is strictly better than R_1 for agent 1. We show that maximizing over the expected utility provided by the sequences L_1 cannot be an optimal action. The expected utility, considering also the own perturbation, provided by sequence L_1 is: $EU_1(L_1) = (1 - l_{1,R_1}(\epsilon))(1 - l_{2,r_1}(\epsilon)) + l_{1,R_1}(\epsilon) - l_{1,R_2}(\epsilon) = 1 - l_{1,R_2}(\epsilon) - l_{2,r_1}(\epsilon) + l_{1,R_1}(\epsilon) l_{2,r_1}(\epsilon)$. The expected utility, considering also the own perturbation, provided by sequence R_1 is: $EU_1(R_1) = l_{1,L_1}(\epsilon)(1 - l_{2,r_1}(\epsilon)) + 1 - l_{1,L_1}(\epsilon) - l_{1,R_2}(\epsilon) = 1 - l_{1,R_2}(\epsilon) - l_{1,L_1}(\epsilon) l_{2,r_1}(\epsilon)$. It can be observed that for every possible combination of $l_{1,L_1}(\epsilon), l_{1,R_1}(\epsilon), l_{2,r_1}(\epsilon), l_{1,R_2}(\epsilon)$ the inequality $EU_1(R_1) > EU_1(L_1)$ holds, since $EU_1(R_1) - EU_1(L_1) = l_{2,r_1}(\epsilon)(1 - l_{1,L_1}(\epsilon) - l_{1,R_1}(\epsilon))$. Therefore, by maximizing over perturbed sequences we cannot verify correctly any EFPE that prescribes $\sigma_{1,L_1} = 1$. □

Notice that the above result does not show that the sequence form cannot be used to verify an EFPE at all, but that, if applicable, the sequence form must be rethought for this problem (and for the problem to compute an EFPE).

5. CONCLUSIONS AND FUTURE WORKS

We studied the problem to verify whether a solution is a given solution concept refining the NE for extensive–form games. This problem is of extraordinary importance. If the verification of a solution concept is intractable, such a solution concept cannot be adopted in practice. While verifying a NE is easy, this may be not the case for NE refinements. In this paper, we complete the results known in the literature concerning the verification of a SE and of an QPE, proving that problems to verify an SE with an arbitrary number of agents and a QPE with two agents are in $\mathcal{P}$ and we provide two pertinent algorithms based on linear programming. We show also that when the input solution is expressed in (non–perturbed) sequence form even verifying an SPE is $\mathcal{NP}$–complete and that sequence form, if applicable, must be rethought for the verification of an EFPE.

In future, we aim at completing our results, exploring the verification of an EFPE with two agents and of a Myerson's proper equilibrium with two agents [8].

6. REFERENCES

[1] L. Blum, A. Brandenburger, and E. Dekel. Lexicographic probabilities and equilibrium refinements. *ECONOMETRICA*, 59(1):81–98, 1991.

[2] S. Ceppi, N. Gatti, G. Patrini, and M. Rocco. Local search techniques for computing equilibria in two–player general–sum strategic–form games. In *AAMAS*, pages 1469–1470, 2010.
[3] R. Chandrasekaran. Total unimodularity of matrices. *SIAM J APPL MATH*, 17(6):1032–1034, 1969.
[4] X. Chen, X. Deng, and S. Teng. Computing Nash equilibria: approximation and smoothed complexity. In *FOCS*, pages 603–612, 2006.
[5] V. Conitzer and T. Sandholm. New complexity results about Nash equilibria. *GAME ECON BEHAV*, 63(2):621–641, 2008.
[6] C. Daskalakis, P. Goldberg, and C. Papadimitriou. The complexity of computing a Nash equilibrium. In *STOC*, pages 71–78, 2006.
[7] C. Daskalakis, A. Mehta, and C. Papadimitriou. A note on approximate Nash equilibria. *THEOR COMPUT SCI.*, 410(17):1581–1588, 2009.
[8] D. Fudenberg and J. Tirole. *Game Theory*. 1991.
[9] N. Gatti and C. Iuliano. Computing an extensive–form perfect equilibrium in two–player games. In *AAAI*, 2011.
[10] A. Ghouila-Houri. Caracterisation des matrices totalement unimodulaires. *C. R. Acad. Sci. Paris*, 254:1192–1194, 1962.
[11] A. Gilpin and T. Sandholm. Lossless abstraction of imperfect information games. *J ACM*, 54(5):25, 2007.
[12] A. Gilpin, T. Sandholm, and T. Sørensen. Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of texas hold'em poker. In *AAAI*, pages 50–57, 2007.
[13] S. Govindan and T. Klumpp. Perfect equilibrium and lexicographic beliefs. *INT J GAME THEORY*, 31(2):229–243, 2003.
[14] K. Hansen, P. Miltersen, and T. Sørensen. The computational complexity of trembling hand perfection and other equilibrium refinments. In *SAGT*, pages 198–209, 2010.
[15] E. Kohlberg and P. Reny. Independence on relative probability spaces and consistent assessments in game trees. *J ECON THEORY*, 75(2):280–313, 1997.
[16] D. R. Kreps and R. Wilson. Sequential equilibria. *ECONOMETRICA*, 50(4):863–894, 1982.
[17] H. Kuhn. Extensive games. *Prooceedings of the National Academy of Sciences*, 36:570–576, 1950.
[18] C. Lemke. Some pivot schemes for the linear complementarity problem. *Mathematical Programming Study*, 7:15–35, 1978.
[19] C. Lemke and J. Howson. Equilibrium points of bimatrix games. *SIAM J APPL MATH*, 12(2):413–423, 1964.
[20] P. Miltersen and T. Sørensen. Computing a quasi–perfect equilibrium of a two–player game. *ECON THEOR*, 42(1):175–192, 2010.
[21] R. Porter, E. Nudelman, and Y. Shoham. Simple search methods for finding a Nash equilibrium. In *AAAI*, pages 664–669, 2004.
[22] T. Sandholm, A. Gilpin, and V. Conitzer. Mixed–integer programming methods for finding Nash equilibria. In *AAAI*, pages 495–501, 2005.
[23] R. Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *INT J GAME THEORY*, 4(1):25–55, 1975.
[24] E. van Damme. A relation between perfect equilibria in extensive form games and proper equilibria in normal form games. *INT J GAME THEORY*, 13(2):1—13, 1984.
[25] E. van Damme. *Stability and Perfection of Nash Equlibria*. Springer, 1991.
[26] J. von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1944.
[27] B. von Stengel. Efficient computation of behavior strategies. *INT J GAME THEORY*, 14(2):220–246, 1996.
[28] B. von Stengel. *Algorithmic Game Theory*, chapter Equilibrium computation for two–player games in strategic and extensive form, pages 53–78. Cambridge, 2007.
[29] L. Wolsey. *Integer Programming*. Wiley–Interscience, 1998.

APPENDIX

A. LEXICOGRAPHIC PERTURBATIONS

Given an ordered vector $\mathbf{z}_1 \in \mathbb{R}^n$, we say that $\mathbf{z}_1$ is *lexico–positive* if the first non–zero element of $\mathbf{z}_1$ is positive. Formally, we write $\mathbf{z}_1 \geq_L \mathbf{0}$. $\mathbf{z}_1$ is strictly lexico–positive if it is lexico–positive and there is at least a strictly positive element. Easily, given a pair of ordered vectors $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^n$, we say that $\mathbf{z}_1 \geq_L \mathbf{z}_2$ if and only if $\mathbf{z}_1 - \mathbf{z}_2$ is lexico positive.

A perturbation $l_{i,a}(\epsilon)$ over action a is a polynomial in ϵ, e.g., $l_{i,a}(\epsilon) = c_1\epsilon + c_2\epsilon^2 + c_3\epsilon^3 + c_4\epsilon^4 + \ldots$, where $c_k \in \mathbb{R}$. A perturbation can be represented as one ordered vector in which the first element is the coefficient c_1 of ϵ, the second element is the coefficient c_2 of ϵ^2, and so on. That is, when ϵ goes to zero, $c_{k_1}\epsilon^{k_1}$ and $c_{k_2}\epsilon^{k_2}$ are comparable if and only if $k_1 = k_2$. Similarly, a perturbed strategy $\boldsymbol{\sigma}_i(\epsilon)$ (analogously, $\mathbf{x}_i(\epsilon)$) can be represented by using an lexico positive ordered vector per action (sequence). Requiring that a perturbed strategy $\boldsymbol{\sigma}_i(\epsilon)$ (analogously, $\mathbf{x}_i(\epsilon)$) is fully mixed is equivalent to requiring that each element $\sigma_{i,a}(\epsilon)$ is strictly lexico positive.

B. QPE COMPUTATION

A QPE with two–agent games can be computed by applying a specific symbolic perturbation $\mathbf{l}_1(\epsilon), \mathbf{l}_2(\epsilon)$ (see [20] for the details on the perturbation) to the LCP to find an NE (the solving algorithm is the same for the computation of NE). Given a perturbed strategy $\mathbf{x}_i \geq \mathbf{l}_i(\epsilon)$, we substitute $\mathbf{x}_i$ with $\tilde{\mathbf{x}}_i + \mathbf{l}_i(\epsilon)$ where $\tilde{\mathbf{x}}_i \geq \mathbf{0}$. The resulting symbolically perturbed LCP is:

$$\tilde{\mathbf{x}}_i \geq \mathbf{0} \qquad \forall i \in \{1,2\} \tag{29}$$
$$F_i \tilde{\mathbf{x}}_i = \mathbf{f}_i - F_i \mathbf{l}_i(\epsilon) \qquad \forall i \in \{1,2\} \tag{30}$$
$$F_i^T \mathbf{v}_i - U_i \tilde{\mathbf{x}}_{-i} \geq \mathbf{0} + U_i \mathbf{l}_{-i}(\epsilon) \qquad \forall i \in \{1,2\} \tag{31}$$
$$\tilde{\mathbf{x}}_i^T \cdot (F_i^T \mathbf{v}_i - U_i \tilde{\mathbf{x}}_{-i} - U_i \mathbf{l}_{-i}(\epsilon)) = 0 \qquad \forall i \in \{1,2\} \tag{32}$$

where $\mathbf{v}_i$ are the dual variables of the best response optimization problems and their values are the expected utilities associated with the best actions for each information set of agent i.