Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence

Computing an Extensive-Form Perfect Equilibrium in Two-Player Games

Nicola Gatti and **Claudio Iuliano**
Dipartimento di Elettronica e Informazione, Politecnico di Milano
Piazza Leonardo da Vinci 32, 20133, Milano, Italy

Abstract

Equilibrium computation in games is currently considered one of the most challenging issues in AI. In this paper, we provide, to the best of our knowledge, the first algorithm to compute a Selten's *extensive–form perfect equilibrium* (EFPE) with two–player games. EFPE refines the Nash equilibrium requiring the equilibrium to be robust to slight perturbations of both players' behavioral strategies. Our result puts the computation of an EFPE into the PPAD class, leaving open the question whether or not the problem is *hard*. Finally, we experimentally evaluate the computational time spent to find an EFPE and some relaxations of EFPE.

Introduction

The interaction among intelligent rational agents is elegantly captured by non–cooperative game theory (Fudenberg and Tirole 1991) that provides *models* and *solution concepts*, but leaves open the problem to compute an equilibrium. This problem is instead central in the computer science community. Prominent theoretical and experimental results have been recently pursued in equilibrium computation, especially (but not exclusively) for finding a Nash equilibrium (NE) in two–player general–sum strategic–form games and in extensive–form zero–sum games. We cite the best known results. Computing an exact NE (Daskalakis, Mehta, and Papadimitriou 2009) and approximating it (Chen, Deng, and Teng 2006) are PPAD–complete problems; an NE of strategic–form games can be computed by algorithms based on linear complementarity programming (Lemke and Howson 1964), static (Porter, Nudelman, and Shoham 2004) and local–search based (Ceppi et al. 2010) support–enumeration, and mixed–integer linear programming (Sandholm, Gilpin, and Conitzer 2005). Extensive–form games can be represented by the *sequence form* (Koller, Megiddo, and von Stengel 1996) that is exponentially smaller than the normal form and with zero–sum games abstractions can be used both *with* information loss (Gilpin, Sandholm, and Sørensen 2007) and *without* information loss (Gilpin and Sandholm 2007); gradient based algorithms (Hoda et al. 2010) are demonstrated to be very efficient to find approximate solutions.

The problem of solving two–player general–sum extensive–form games has received less attention and appears as the "natural next issue of the agenda" according to (von Stengel 2007). The use of sequence form and the inadequateness of the concept of NE with extensive–form games (sequential rationality of the strategies is not granted) make the algorithms for the strategic form non–directly applicable. Appropriate solutions concepts are: *subgame perfect equilibrium* (SPE) when information is perfect and *sequential equilibrium* (SE), *quasi–perfect equilibrium* (QPE), and *extensive–form perfect equilibrium* (EFPE) when information is imperfect. (Due to reason of space we omit normal–form refinements.) There are three main computational results: an NE can be computed (with sequence form) by using the Lemke algorithm (LA) (Koller, Megiddo, and von Stengel 1996); a more efficient variation is presented in (von Stengel, van den Elzen, and Talman 2002); a QPE can be computed by using the LA with symbolic perturbation (Miltersen and Sørensen 2010) and this problem is in the PPAD class, but it is not known whether the problem is hard. No significative experimental results are provided in the literature.

In this paper, we develop the first algorithm to compute an EFPE with two–player games. This concept refines NE requiring the equilibrium to be robust w.r.t slight perturbations of all the agents, while QPE takes into account only the opponents' perturbations. Although an EFPE can include weakly dominated strategies and, due to this reason, can be considered less appealing than QPE (Mertens 1995), in many practical applications an agent cannot perfectly control its own strategy and then she will play an EFPE. The fact that an EFPE is not merely a refinement of QPE, the set of QPEs and EFPEs being disjoint for some game instances, strengthens the need for *ad–hoc* algorithms to compute an EFPE. Finally, EFPE computation can pave the way to the study of situations (currently mostly unexplored) where an agent incurs costs in controlling its strategy (van Damme 1991). We improve the state of the art as follows.

- To find an EFPE, we extend the work presented in (Miltersen and Sørensen 2010) providing a linear complementarity programming formulation (satisfied by a wide class of EFPEs) where the perturbation is uniform (the same for both players) and we modify the LA to solve it showing that is computationally expensive.

- To overcome this problem, we provide an alternative linear complementary formulation (satisfied by a restricted but non–empty class of EFPE) where the perturbation is non–uniform (the two players having different perturbations) and we show that in this case the LA does not require additional computational effort w.r.t. to that required in (Miltersen and Sørensen 2010).
- We prove that computing an EFPE is into the PPAD class, our algorithm being a path–following, but we leave open the question whether or not the problem is hard.
- We provide, to the best of our knowledge, the first experimental analysis, in terms of computational time, memory, and numerical stability, of algorithms computing NE refinements (QPE and EFPE) in extensive–form games.

Extensive–Form Games

A perfect–information extensive–form game (Fudenberg and Tirole 1991) is a tuple $(N, A, V, T, \iota, \rho, \chi, u)$, where: N is the set of agents, A is the set of actions, V is the set of decision nodes, T is the set of terminal nodes, $\iota : V \to N$ is the function that specifies the agent that acts at a given decision node, $\rho : V \to \wp(A)$ returns the actions available to agent $\iota(w)$ at decision node w, $\chi : V \times A \to V \cup T$ assigns the next (decision or terminal) node to each pair composed of a decision node w and an action a available at w, and $u = (u_1, \ldots, u_n)$ is the set of agents' utility functions where $u_i : T \to \mathbb{R}$. An extensive–form game is with imperfect information when an action of some agent is not perfectly observable by the opponents. Formally, it is a tuple $(N, A, V, T, \iota, \rho, \chi, u, I)$ where $(N, A, V, T, \iota, \rho, \chi, u)$ is a perfect–information extensive–form game and $I = (I_1, \ldots, I_n)$ with $I_i = (I_{i,1}, \ldots, I_{i,k_i})$ is a partition of set $V_i = \{w \in V : \iota(w) = i\}$ with the property $\rho(w) = \rho(w')$ whenever there exists a j for which $w, w' \in I_{i,j}$. The sets $I_{i,j}$ are called *information sets*. We focus on games with *perfect recall* (Fudenberg and Tirole 1991) where each agent recalls all the own previous actions and those of the opponents.

In our work we consider only *behavioral strategies*, i.e., the strategies in which each agent's (potentially probabilistic) choice at each information set is made independently of the choices at other nodes. Essentially, a behavioral strategy σ_i assigns each information set $h \in I_i$ a probability distribution over the actions available at h. From here on, we shall refer to "behavioral strategies" simply as "strategies".

We call $\mu = \{\mu_i : i \in N\}$ the beliefs of the agents, specifying (for every information set) what strategies an agent believes that the opponents will do. An SE (that is the "natural" extension of the SPE to situations with imperfect information) is an *assessment* (σ, μ) such that for all $i \in N$ strategy σ_i is sequentially optimal with respect to μ_i (in the sense of backward induction), and there exists a perturbed (fully mixed) strategy profile $\sigma(\epsilon)$ with $\lim_{\epsilon \to 0} \sigma(\epsilon) = \sigma$ and the limit of the sequence of beliefs derived from $\sigma(\epsilon)$ by using the Bayes rule converges to μ. QPE and EFPE pose more severe constraints. A QPE is a strategy profile σ such that there exists a perturbed strategy profile $\sigma(\epsilon)$, with $\lim_{\epsilon \to 0} \sigma(\epsilon) = \sigma$, that for all $i \in N$ strategy σ_i is optimal with respect to $\sigma_{-i}(\epsilon)$ for all $\epsilon \in [0, \epsilon_0]$ with some $\epsilon_0 > 0$. A QPE is an SE requiring also optimality w.r.t. slight perturbations of the opponents' strategies. An EFPE is a strategy profile σ such that there exists a perturbed strategy profile $\sigma(\epsilon)$, with $\lim_{\epsilon \to 0} \sigma(\epsilon) = \sigma$, that for all $i \in N$ strategy σ_i is optimal with respect to $\sigma(\epsilon)$ for all $\epsilon \in [0, \epsilon_0]$ with some $\epsilon_0 > 0$. An EFPE is an SE requiring also optimality w.r.t perturbations of the opponents' and own strategies.

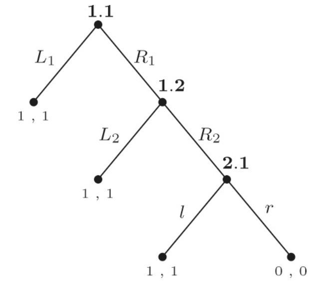

Figure 1: $(L_1, *)$ is the unique EFPE; $(L_1, *), (R_1 L_2, *)$ are QPEs; $(L_1, *), (R_1 L_2, *), (R_1 R_2, l)$ are SEs.

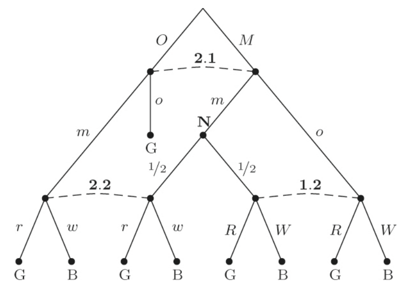

Figure 2: **N** is nature (1/2 is a probability); G and B are outcomes: both players prefer G to B. $(O, mr), (MR, o)$ are EFPEs; (MR, mr) is the unique QPE.

Example 1 *Consider the game represented in Fig. 1. The SPEs are: $(L_1, *), (R_1 L_2, *), (R_1 R_2, l)$. They are SEs, the game being with perfect information. Suppose $\sigma_2 = l$. When agent 1 accounts for ϵ perturbations over agent 2's strategy (i.e., $\sigma_2 = r$ is played with a probability of ϵ), we have $E[u_1(R_1 R_2)] = 1 - \epsilon < 1 = E[u_1(R_1 L_2)] = E[u_1(L_1)]$ and therefore only $(L_1, *), (R_1 L_2, *)$ are QPEs (where $E[u_i(s)]$ is the expected utility of agent i from making strategy s). When agent 1 accounts for ϵ perturbations also over her own strategy, we have $E[u_1(R_1)] = 1 - \epsilon + \epsilon(1 - \epsilon) < 1 = E[u_1(L_1)]$ and therefore only $(L_1, *)$ is an EFPE.*

Surprisingly, the sets of EFPEs and of QPEs can be disjoint as shown in Example 2, see also (Mertens 1995).

Example 2 *Consider the game represented in Fig. 2, where two agents choose who will be the agent (i.e., 'me' or the 'other') that will take a decision (i.e., 'R'/'r' or 'W'/'w'). The game presents a chance node. (O, mr) and (MR, o) are EFPEs, and (MR, mr) is the unique QPE.*

The efficient computation of an NE and its refinements is based on the *sequence form* (Koller, Megiddo, and von Stengel 1996). It represents a game as a pair composed of a bimatrix and a set of linear constraints where: each agent's actions are (terminal and non–terminal) sequences q of her actions in the game tree (e.g., in Fig. 1, $q = R_1$ is a non–terminal, while $q = R_1 R_2$ is terminal); given a profile of sequences specifying a sequence per player, if it leads to a terminal node, then the agents' payoffs are their utilities over such a node, otherwise the payoffs are null; and, called $q' = q|a$ the sequence obtained by extending q with action a (e.g., in Fig. 1, $q' = R_1 R_2$ with $q = R_1$ and $a = R_2$), the probability of sequence q is equal to the sum of the probabilities of the sequences q' that extend it. Once a game is solved in sequence form, the behavioral strategies can be easily derived.

Called $\mathbf{x}_i$ the probability vector of the strategy of agent i, the sequence form constraints can be formulated as $F_i \mathbf{x}_i = \mathbf{f}_i$ where F_i is a matrix and $\mathbf{f}_i$ is a vector. Called U_i the utility matrix of agent i and $\mathbf{v}_i$ the dual variables expressing the expected utility for each information set, the (mixed) linear complementarity program for finding an NE is:

$$F_i \mathbf{x}_i = \mathbf{f}_i \qquad \forall i \in \{1,2\} \tag{1}$$

$$F_i^T \mathbf{v}_i - U_i \mathbf{x}_{-i} - \geq \mathbf{0} \qquad \forall i \in \{1,2\} \tag{2}$$

$$\mathbf{x}_i \geq 0 \qquad \forall i \in \{1,2\} \tag{3}$$

$$\mathbf{x}_i^T \cdot (F_i^T \mathbf{v}_i - U_i \mathbf{x}_{-i}) = 0 \qquad \forall i \in \{1,2\} \tag{4}$$

Example 3 *Consider the game depicted in Fig. 2. The sequences are* $\{\varnothing, O, M, MR, MW\}$ *for agent 1 and* $\{\varnothing, o, m, mr, mw\}$ *for agent 2. We have (with abuse of notation we use G and B as payoffs, where G > B):*

$$U_i = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & G & 0 & G & B \\ 0 & 0 & 0 & \frac{G}{2} & \frac{B}{2} \\ 0 & G & \frac{G}{2} & 0 & 0 \\ 0 & B & \frac{B}{2} & 0 & 0 \end{bmatrix} F_i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 \end{bmatrix} \mathbf{f}_i = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

The mathematical program (1)–(4) can be solved with LA (Koller, Megiddo, and von Stengel 1996). This algorithm is a simplex–like algorithm, effectively represented by a tableau, that, starting from a non–complementary basic solution, finds a complementary basic solution by changing the set of basic variables by a sequence of complementary pivoting steps. The complementary pivoting rule prescribes that at each step the entering variable is the complementary of the previous leaving variable and the leaving variable is determined by the *minimum ratio test* where ties are broken lexicographycally (von Stengel 2007).

Computation of NE Refinements through Perturbed Games

In order to compute refinements of NE based on the idea of perfection, we need to introduce a symbolic perturbation, defined on ϵ, over the strategies. The result is an ϵ–perturbed game in which there is a strictly positive lower bound (as function of ϵ) on the probability with which each strategy is played. Following (Miltersen and Sørensen 2010), to find a QPE we can construct a game in which the perturbation over the probability with which a sequence q is played is $\epsilon^{|q|+1}$ where $|q|$ is the length of q. Formally, sequence $\varnothing$ has a length equal to zero and any other sequence has a length equal to the number of actions it contains. We call $\mathbf{l}_i(\epsilon)$ the perturbation vector such that $\mathbf{x}_i \geq \mathbf{l}_i(\epsilon)$. An NE of such a game is a QPE of the original non–perturbed game.

Example 4 *Consider Fig. 2,* $\mathbf{l}_i(\epsilon) = [\epsilon, \epsilon^2, \epsilon^2, \epsilon^3, \epsilon^3]$.

By introducing $\mathbf{l}_i(\epsilon)$, the mathematical program (1)–(4) can be written as:

$$F_i \tilde{\mathbf{x}}_i = \mathbf{f}_i - F_i \mathbf{l}_i(\epsilon) \qquad \forall i \in \{1,2\} \tag{5}$$

$$F_i^T \mathbf{v}_i - U_i \tilde{\mathbf{x}}_{-i} - U_i \mathbf{l}_{-i}(\epsilon) \geq \mathbf{0} \qquad \forall i \in \{1,2\} \tag{6}$$

$$\tilde{\mathbf{x}}_i \geq 0 \qquad \forall i \in \{1,2\} \tag{7}$$

$$\tilde{\mathbf{x}}_i^T \cdot (F_i^T \mathbf{v}_i - U_1 \tilde{\mathbf{x}}_{-i} - U_1 \mathbf{l}_{-i}(\epsilon)) = 0 \qquad \forall i \in \{1,2\} \tag{8}$$

where $\tilde{\mathbf{x}}_i = \mathbf{x}_i - \mathbf{l}_i(\epsilon)$. The above program can be solved by a variant of LA in which perturbation is symbolic and it only affects the minimum ratio test.

Finding an EFPE in Uniform ϵ–Perturbed Games

Now, we construct a sequence of ϵ–perturbed games and the EFPE is the limit of equilibria of the sequence of games. Consider the program (5)–(8): although both agents' strategies are perturbed, the reason why such a program does not lead to an EFPE is that in the constraints that assure each agent to play her best response, i.e. (6), no agent takes into account her own perturbation. More precisely, call $E[u_i(q)]$ the utility expected from playing sequence q, defined as $\mathbf{e}(q)^T U_i \mathbf{x}_{-i}$, where $\mathbf{e}(q)$ is a vector in which there is one in position of q and zero elsewhere. For each sequence q of agent i such that $q = q'|a$ and a is played at h, there exists a constraint (6) that assures that $E[u_i(q)]$ plus the utility of the information sets reachable by q is not larger that the expected utility of agent i at information set h, i.e., $v_i(h)$.

Example 5 *Consider the information sets of agent 1 in Fig. 2. For* $\mathbf{1.1}$*, constraints (6) can be written as* $v_1(\mathbf{1.1}) - E[u_1(O)] \geq 0$ *and* $v_1(\mathbf{1.1}) - v_1(\mathbf{1.2}) - E[u_1(M)] \geq 0$. *It can be easily seen that* $v_1(\mathbf{1.2})$ *takes into account perturbations only over agent 2's strategy. Indeed, by constraints (6) we have* $v_1(\mathbf{1.2}) - E[u_1(MR)] \geq 0$ *and* $v_1(\mathbf{1.2}) - E[u_1(MW)] \geq 0$ *and constraints (8) force* $v_1(\mathbf{1.2}) = E[u_1(MR)] = G(\tilde{\mathbf{x}}(o) + \epsilon^2) + {}^1\!/_2 G(\tilde{\mathbf{x}}(m) + \epsilon^2)$.

In order to allow each agent to take into account perturbations over her own strategy, we modify constraints (6) by introducing an additional perturbation. Exactly, we need that: for each sequence q of agent i such that $q = q'|a$ and a is played at h, there exists a variant of constraints (6) assuring that the expected utility of agent i at information set h (i.e., $v_i(h)$) is not smaller than the sum of the following terms:

- $(1 - (n_h - 1)\epsilon) E[u_i(q)]$, where n_h is the number of actions available at h,
- $(1 - (n_h - 1)\epsilon)$ multiplied by the expected utility (captured by $\mathbf{v}_i$) of the information sets reachable by q,
- the sum of ϵ multiplied by $E[u_i(r)]$ plus the expected utility of the information sets reachable by r for every $r = r'|a$ with a played at h except q.

Example 6 *Consider the information sets of agent 1 in the game depicted in Fig. 2. We need that* $v_1(\mathbf{1.1}) - (1 - \epsilon)E[u_1(O)] - \epsilon E[u_1(M)] - \epsilon v_1(\mathbf{1.2}) \geq 0$ *and* $v_1(\mathbf{1.1}) - (1 - \epsilon) v_1(\mathbf{1.2}) - (1 - \epsilon)E[u_1(M)] - \epsilon E[u_1(O)] \geq 0$*, and that* $v_1(\mathbf{1.2}) - (1 - \epsilon)E[u_1(MR)] - \epsilon E[u_1(MW)] \geq 0$ *and* $v_1(\mathbf{1.2}) - (1 - \epsilon)E[u_1(MW)] - \epsilon E[u_1(MW)] \geq 0$.

To distinguish the perturbation introduced to compute a QPE from the one introduced by us to compute an EFPE, we call $\mathbf{l}_i$ *primal perturbation*, being defined on the primal variables, i.e., $\mathbf{x}_i$; we call our perturbation *dual perturbation*, being defined on the dual constraints of $\mathbf{x}_i$. What we obtain is an ϵ–perturbed game with primal and dual perturbation, whose NEs are, by definition of the EFPE concept, EFPEs of the original non–perturbed game.

Example 7 *Consider information set* $\mathbf{1.1}$ *in the game depicted in Fig. 2. Suppose that agent 2's strategy prescribes to play* mr*. The constraint corresponding to* $q = O$ *is* $v_1(\mathbf{1.1}) \geq G + (2G + B)\epsilon + (3/2B - 1/2G)\epsilon^2$*, while the constraint corresponding to* $q = M$ *is* $v_1(\mathbf{1.1}) \geq G + (3/2G + B)\epsilon + (3/2B\epsilon^2 - 1/2)\epsilon^2$*. By lexicographic order, we have that* $q = O$ *provides a larger expected utility than* $q = M$ *and thus* $q = O$ *is a best response to* mr*. It can be easily observed that* $q = mr$ *is a best response to* O *and then* (O, mr) *is an NE of the perturbed game and an EFPE of the original non–perturbed game. The same holds for* (MR, o)*.*

The above perturbation can be captured in the formulation by redefining matrices U_i and F_i. We call $U_i(\epsilon)$ and $F_i(\epsilon)$ the new matrices. We denote by $U_i(q)$ and $U_i(\epsilon, q)$ the row vectors corresponding to sequence q, by $F_i(q)$ and $F_i(\epsilon, q)$ the column vectors corresponding to sequence q, and by $q \in h$ if $q = q'|a$ where a is played at h. We have: $U_i(\epsilon, q) = (1 - (n_h - 1)\epsilon)U_i(q) + \epsilon \sum_{q':q' \in h, q \in h} U_i(q')$ and $F_i(\epsilon, q) = (1 - (n_h - 1)\epsilon)F_i(q) + \sum_{q':q' \in h q \in h} F_i(q')$ (we recall that n_h is the number of sequences $q \in h$). The formulation for finding an EFPE is:

constraints (5) and (7)

$$F_i(\epsilon)^T \mathbf{v}_i - U_i(\epsilon)\tilde{\mathbf{x}}_{-i} - U_i(\epsilon)\mathbf{l}_{-i}(\epsilon) \geq \mathbf{0} \qquad \forall i \in \{1, 2\} \tag{9}$$

$$\tilde{\mathbf{x}}_i^T \cdot (F_i(\epsilon)^T \mathbf{v}_i - U_i(\epsilon)\tilde{\mathbf{x}}_{-i} - U_i(\epsilon)\mathbf{l}_{-i}(\epsilon)) = 0 \qquad \forall i \in \{1, 2\} \tag{10}$$

Example 8 *For the game in Fig. 2, matrix* $U_i(\epsilon) =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & (1-\epsilon)G & 0 & (1-\epsilon)G + \epsilon\frac{G}{2} & (1-\epsilon)B + \epsilon\frac{B}{2} \\ 0 & \epsilon G & 0 & (1-\epsilon)\frac{G}{2} + \epsilon G & (1-\epsilon)\frac{B}{2} + \epsilon B \\ 0 & (1-\epsilon)G + \epsilon B & (1-\epsilon)\frac{G}{2} + \epsilon\frac{B}{2} & 0 & 0 \\ 0 & (1-\epsilon)B + \epsilon G & (1-\epsilon)\frac{B}{2} + \epsilon\frac{G}{2} & 0 & 0 \end{bmatrix}$$

and $F_i(\epsilon) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 \\ 0 & -\epsilon & -(1-\epsilon) & 1 & 1 \end{bmatrix}$.

Notice that the above constraints are not satisfied by all the EFPEs, but exclusively by the EFPEs in which each behavioral strategy has the same perturbation ϵ; the same is in (Miltersen and Sørensen 2010) for finding QPEs.

The program (5), (7), (9), (10) cannot be directly solved by the LA, but it can be appropriately modified. In particular, the peculiarity of the program for finding an EFPE (with respect to the program for finding a QPE) is that the symbolic perturbation is multiplied by the variables. To address it, the LA must be modified as follows.

Tableau representation. Each element of the tableau is a polynomial in ϵ. In order to represent it, we need that each element of the tableau is a vector reporting the coefficients of the corresponding polynomial. Notice that, in the initial tableau, the coefficients of the variables ($F_i(\epsilon)$ and $U_i(\epsilon)$) are polynomial with a degree smaller than or equal to one, whereas the degree of the polynomials related to the perturbation over the constants can have degree larger than one. The vectors we use to represent the polynomial must be dynamic since the degree of the polynomials can rise during the pivoting steps.

Pivoting. The classical pivoting step requires one to divide elements in tableau. In our case, the elements are polynomial in ϵ and their division may not be done since their remainder may be non–zero. In order to overcome this problem, *integer pivoting* (von Stengel 2007) can be used.[1] It can be easily observed that at each step of the integer pivoting the degrees of the polynomials rises by one.

The algorithm is complete, being granted to terminate always; the proof is the same of (Miltersen and Sørensen 2010). In addition, since the above algorithm is a path–following algorithm, computing an EFPE is in the PPAD class, but it is not known whether the problem is hard.

Finding an EFPE in Non–Uniform ϵ–Perturbed Games

The algorithm described in the previous section is not efficient because the pivoting step requires multiplications and divisions between polynomials in ϵ whose length (in terms of number of coefficients) rises with the number of pivoting steps. This is because the dual perturbation is multiplicative instead of being additive (as it is the primal one) w.r.t. the variables. By considering a different (non–uniform) perturbation it is possible to formulate the problem in such a way that the dual symbolic perturbation is additive w.r.t. the variables and therefore only the constants are polynomials in ϵ. In order to use an additive dual perturbation, we need to redefine the primal and dual perturbations.

Primal perturbation redefinition. The basic idea here is to force different primal perturbations, in terms of degrees of ϵ, to the two agents. Call l_i the length of the longest sequence of agent i. When both agents have the same perturbation ϵ over the behavioral strategies, in $\mathbf{l}_i(\epsilon)$ the minimum degree of ϵ is 1 and the maximum one is $l_i + 1$. Now, we assume that the perturbation of one agent, say agent 1, over all her strategies has a lower degree than the perturbation of the opponent, say agent 2 (the study of the reverse case is analogous). An NE of the associated perturbed game keeps to be an EFPE of the original non–perturbed game. We can redefine the perturbations over the strategies as, called $\mathbf{l}'_i(\epsilon)$ the new perturbations, $\mathbf{l}'_1(\epsilon) = \mathbf{l}_1(\epsilon)$ and $\mathbf{l}'_2(\epsilon) = \epsilon^{l_1+1}\mathbf{l}_2(\epsilon)$.

Example 9 *Consider Fig. 2:* $\mathbf{l}'_1(\epsilon) = [\epsilon, \epsilon^2, \epsilon^2, \epsilon^3, \epsilon^3]$ *and* $\mathbf{l}'_2(\epsilon) = [\epsilon^4, \epsilon^5, \epsilon^5, \epsilon^6, \epsilon^6]$.

Dual perturbation redefinition. Consider constraints (9). For consistency, the redefinition of the primal perturbation affects also $U_2(\epsilon)$ and $F_2(\epsilon)$: all the terms in ϵ are multiplied by ϵ^{l_1+1}. After this redefinition, it can be easily ob-

[1]Call $a_{i,j}$ an element of the tableau, and i' and j' the pivot row and column, respectively, such that $a_{i',j'}$ is the pivot. Integer pivoting works as follows. To eliminate all the coefficients in the pivot column, multiply each row $i \neq i'$ by $a_{i',j'}$ and subtract it from row i' multiplied by $a_{i,j'}$. Then, divide each row $i \neq i'$ by the pivot at the previous step. This last step requires division between polynomials and can be done as described in (Barnard 2008).

served that the primal and the dual perturbations have "non–overlapping" degrees of ϵ, i.e., in constraints (9) when $i = 1$ the largest degree of the dual perturbation is smaller than the smallest degree of $U_1(\epsilon)\mathbf{l}'_2(\epsilon)$, and the reverse when $i = 2$. Therefore, primal and dual perturbations can be treated separately. Finally, it can be easily observed that, limiting to the dual perturbation, there always exists an EFPE when agent i breaks ties by preferring the outcome such that the agent i's sequence leading to it is the shortest (i.e., when the possible trembles of agent i are minimized).

Example 10 *Consider agent 1 in Fig. 2: if O and MR provide the same expected utility, there is an EFPE where agent 1 prefers O.*

The dual perturbation can be easily defined as a *malus* that is minimum for the dual constraints corresponding to the smallest sequence and is maximum in correspondence of the longest sequence. We denote the dual (additive) perturbation by $\mathbf{r}_i(\epsilon)$. Called $\mathbf{r}_i(\epsilon, q)$ the element of $\mathbf{r}_i(\epsilon)$ corresponding to sequence q, we have $\mathbf{r}_1(\epsilon, q) = -\frac{\epsilon^{l_1+2}}{\mathbf{l}_1(\epsilon,q)}$ and $\mathbf{r}_2(\epsilon, q) = -\frac{\epsilon^{l_1+l_2+3}}{\mathbf{l}_2(\epsilon,q)}$

Example 11 *Consider Fig. 2: $\mathbf{r}_1(\epsilon) = -[\epsilon^3, \epsilon^2, \epsilon^2, \epsilon^1, \epsilon^1]$ and $\mathbf{r}_2(\epsilon) = -[\epsilon^6, \epsilon^5, \epsilon^5, \epsilon^4, \epsilon^4]$.*

The formulation for finding an EFPE is:

constraints (5) and (7)

$$F_i^T \mathbf{v}_i - U_i \tilde{\mathbf{x}}_{-i} + \mathbf{r}_i(\epsilon) - U_i \mathbf{l}'_{-i}(\epsilon) \geq \mathbf{0} \quad (11)$$

$$\tilde{\mathbf{x}}_i^T \cdot (F_i^T \mathbf{v}_i + \mathbf{r}_i(\epsilon) - U_i \tilde{\mathbf{x}}_{-i} - U_i \mathbf{l}'_{-i}(\epsilon)) = 0 \quad (12)$$

The above program can be solved with the Lemke's algorithm as in (Miltersen and Sørensen 2010) and it is granted to terminate. Notice that fixing the agent 1's perturbation degree higher than the agent 2's can lead to different solutions.

Example 12 *Consider Fig. 2. Only (O, mr) is an NE of the above perturbed game. We have $E[u_1(O)] = G - \epsilon^2 - \epsilon^6(G - B)$, $E[u_1(M)] = 1/2G - \epsilon^2 - 1/2\epsilon^6(G - B) + v_1(\mathbf{1.2})$, and $v_1(\mathbf{1.2}) = E[u_1(MR)] = 1/2G - \epsilon + \epsilon^5 G$. It can be observed that O is the agent 1's best response to mr. Furthermore, it can be observed that when the perturbation of agent 1 has a lower degree of the perturbation of agent 2, (MR, o) is not an NE of the above perturbed game. In order for (MR, o) to be an NE (of the above perturbed game) we need to make that the agent 2's perturbation has a degree lower than agent 1's.*

Experimental Evaluation

We implement in C language the algorithm to find an NE, its variation to find a QPE, and our non–uniform ϵ–perturbed variation to find an EFPE (about our uniform ϵ–perturbed variation to find EFPE we provide some consideration below). For each algorithm we develop four implementations: either based on the original LA (denoted by L1) or based on (von Stengel, van den Elzen, and Talman 2002) (denoted by L2) and with either floating point or integer variables with arbitrary numerical precision. Furthermore, to improve efficiency, we take explicitly only the inverse of the basis and we apply pivoting only on it, whereas for the remaining portion of the tableau a sparse representation is used. The algorithms were executed on a 2.00 GHz 4 GB RAM UNIX computer.

We develop a random generator of extensive–form instances, no generator being currently available. The parameters are: tree depth l, tree branching b, tree *information set density* $\rho \in [0, 1]$ where $\rho = 0$ means that all the information sets contain a single decision node (i.e., the game is with perfect information), while $\rho = 1$ means that all the information sets contain the maximum set of decision nodes (i.e., the game is equivalent to a strategic–form game). The agent who plays at level l' is randomly chosen. Payoffs are generated uniformly from 0 to 100.

Our first experimental analysis is directed on the use of floating points vs. integer variables. The LA and its variations are well known to suffer from numerical stability. (Due to numerical approximation with floating point variables the whole pivot column may have non–positive coefficients and the pivoting stops without finding any equilibrium.) We evaluate our algorithms with 100 game instances with parameters $\rho \in [0, 1]$, $l \in \{2, 4\}$ and $b \in \{2, 3\}$. For all the game instances, all the floating point implementations (L1 and L2) terminate without finding equilibria. We obtain the same result with integer variables. This shows that only arbitrary precision variables can be used.

For all the possible combinations of the following values of the parameters $\rho \in \{0.0, 0.5, 0.8\}$, $l \in \{2, \ldots, 12\}$, $b \in \{2, 3, 4\}$, we generate five game instances and we apply our algorithms with arbitrary numerical precision. In Tab. 1 we report the averages of the most significant results: the size of the instances in terms of number of agents' information sets ($|H_i|$) and of agents' sequences ($|S_i|$), the number of pivoting steps (p.s.) and the computational time for all the possible algorithm configurations. Non–terminate executions are due to deadline (30 m) expiration.

L2 always outperforms L1 both in terms of computational time and in number of pivoting steps. This is because both the number of variables in L1 is larger than those in L2 and L2 is more efficient in the search. The computational time and number of pivoting steps for computing a NE and its refinements QPE and EFPE differ for no more than 8% with both L1 and L2. Anyway, the equilibria found by the algorithms are usually different: L1–NE, L1–QPE, L1–EFPE find different equilibria in the 80% of the cases, while L2–NE, L2–QPE, L2–EFPE always find the same equilibria. It is worth remarking that as ρ increases the number of $|S_i|$ and $|H_i|$ decreases and the problem becomes easier.

On the basis of the above results we can estimate the computational time to solve our uniform ϵ–perturbed formulation for finding an EFPE. Tableau elements being polynomials in ϵ, the number of operations per pivoting step rises linearly in the number of pivoting steps, say s, and it is reasonable to expect that s is of the same magnitude of the number of pivoting steps, say s', for the non–uniform ϵ–perturbed formulation. Therefore, we expect that the ratio between the computational time for the uniform ϵ–perturbed formulation and that for the uniform one goes asymptotically with $O(s')$.

ρ	b	l	$\lvert H_1\rvert$	$\lvert H_2\rvert$	$\lvert S_1\rvert$	$\lvert S_2\rvert$	L1-NE		L1-QPE		L1-EFPE		L2-NE		L2-QPE		L2-EFPE	
							p.s.	time [s]	p.s.	time [s]	p.s.	time [s]	p.s.	time [s]	p.s.	time [s]	p.s.	time [s]
0.0	2	8	118	138	237	276	502	17.96	505	18.32	512	18.65	158	3.20	158	3.39	158	3.19
0.0	2	9	271	241	543	482	1040	146.25	1041	147.60	1049	149.00	153	12.22	153	12.18	153	12.16
0.0	2	10	457	567	915	1134	2040	1142.22	2068	1175.74	2037	1152.14	220	69.18	220	68.91	220	68.75
0.0	2	11	1177	871	2354	1743	–	–	–	–	–	–	360	443.69	360	442.68	360	441.63
0.0	3	5	67	54	202	163	253	3.28	259	3.45	253	3.38	120	1.00	120	1.00	120	1.00
0.0	3	6	254	111	762	333	699	77.56	761	92.59	700	79.27	209	15.16	209	15.09	209	16.53
0.0	3	7	758	336	2274	1008	–	–	–	–	–	–	290	183.51	290	182.66	290	182.46
0.0	4	5	125	217	498	869	672	94.54	692	98.94	690	99.79	279	27.55	279	27.46	279	27.53
0.0	4	6	421	945	1682	3781	–	–	–	–	–	–	758	1155.66	758	1194.11	758	1153.84
0.0	5	5	559	222	2796	1111	1356	1305.60	1388	1349.54	1311	1278.20	292	211.85	292	211.61	292	210.91
0.5	2	8	116	62	232	125	276	5.10	288	5.45	285	5.42	87	0.85	87	0.85	87	0.84
0.5	2	9	157	194	315	389	461	31.83	487	36.05	478	32.60	115	4.35	115	4.33	115	4.33
0.5	2	10	357	332	715	665	1070	268.88	1117	282.00	1104	280.63	268	38.94	268	38.78	268	38.86
0.5	2	11	751	502	1503	1005	–	–	–	–	–	–	293	141.24	293	140.83	293	140.50
0.5	3	6	135	83	405	250	330	14.10	352	15.20	344	15.65	114	3.31	114	3.30	114	3.30
0.5	3	7	108	564	324	1693	986	370.64	1021	386.67	967	363.85	146	37.36	146	34.77	146	35.68
0.5	3	8	1152	773	3456	2320	–	–	–	–	–	–	243	456.83	243	455.80	243	454.72
0.5	4	5	116	97	465	387	320	19.07	315	19.33	324	19.71	97	4.10	97	4.10	97	4.11
0.5	4	6	254	564	1017	2255	1015	797.55	1095	871.46	1090	869.06	150	85.37	150	85.17	150	85.16
0.5	5	5	486	110	2427	547	990	795.46	1088	853.23	1003	820.85	697	477.23	697	478.35	697	476.15
0.8	2	9	138	74	277	149	251	8.79	275	10.17	266	10.12	60	0.76	60	0.76	60	0.76
0.8	2	10	135	322	271	644	685	93.87	682	109.58	668	93.69	132	9.91	132	9.88	132	9.87
0.8	2	11	299	476	599	953	1096	455.30	1124	473.59	1100	462.78	165	42.46	165	42.28	165	42.21
0.8	2	12	621	1070	1243	2141	–	–	–	–	–	–	275	280.86	275	265.73	275	265.49
0.8	3	6	112	60	336	180	310	10.13	309	10.35	304	10.31	66	1.10	66	1.10	66	1.09
0.8	3	7	279	106	838	319	451	79.27	465	83.59	462	83.62	53	4.51	53	4.50	53	4.50
0.8	3	8	597	398	1790	1193	1253	1023.35	1251	1028.51	1242	1019.76	130	82.74	130	82.20	130	82.23
0.8	4	5	100	24	401	94	187	4.50	202	4.96	202	5.05	50	0.76	50	0.76	50	0.77
0.8	4	6	114	354	456	1416	725	215.23	746	224.22	724	215.58	149	31.73	149	31.58	149	37.87
0.8	5	5	120	95	601	474	287	24.40	297	25.95	288	25.06	56	3.39	56	3.37	56	3.36

Table 1: Experimental results. We denote by 'L1' the Lemke algorithm, by 'L2' the variation proposed in (von Stengel, van den Elzen, and Talman 2002), by '–' algorithm non–terminations due to deadline expiration (30 m).

Concluding Remarks

Developing algorithms for general–sum extensive–form games is one of the next issue of the equilibrium computation agenda. The concept of EFPE is particularly interesting since it captures the situation wherein every agent accounts for possible trembles of her own and opponents' strategies. We provide the first algorithm to compute an EFPE. The algorithm is based on linear complementarity programming. Furthermore, we provide the first experimental analysis of algorithms to find NE refinements (QPE and EFPE) with extensive–form games showing numerical limitations and comparing computational times.

In future, we will extend our algorithms to the situations in which agents are more than two and/or can have costs in controlling their strategies, and to find extensive–form *proper equilibria* (van Damme 1991).

References

Barnard, S. 2008. *Higher Algebra*. Barnard Press.

Ceppi, S.; Gatti, N.; Patrini, G.; and Rocco, M. 2010. Local search techniques for computing equilibria in two-player general-sum strategic-form games. In *AAMAS*, 1469–170.

Chen, X.; Deng, X.; and Teng, S. 2006. Computing Nash equilibria: approximation and smoothed complexity. In *FOCS*, 603–612.

Daskalakis, C.; Mehta, A.; and Papadimitriou, C. 2009. A note on approximate Nash equilibria. *THEOR COMPUT SCI*. 410(17):1581–1588.

Fudenberg, D., and Tirole, J. 1991. *Game Theory*.

Gilpin, A., and Sandholm, T. 2007. Lossless abstraction of imperfect information games. *J ACM* 54(5):25.

Gilpin, A.; Sandholm, T.; and Sørensen, T. 2007. Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of texas hold'em poker. In *AAAI*, 50–57.

Hoda, S.; Gilpin, A.; Peña, J.; and Sandholm, T. 2010. Smoothing techniques for computing Nash equilibria of sequential games. *MATH OPER RES* 35(2):494–512.

Koller, D.; Megiddo, N.; and von Stengel, B. 1996. Efficient computation of equilibria for extensive two-person games. *GAME ECON BEHAV* 14(2):220–246.

Lemke, C., and Howson, J. 1964. Equilibrium points of bimatrix games. *SIAM J APPL MATH* 12(2):413–423.

Mertens, J. 1995. Two examples of strategic equilibrium. *GAME ECON BEHAV* 8(2):378—388.

Miltersen, P., and Sørensen, T. 2010. Computing a quasi-perfect equilibrium of a two-player game. *ECON THEOR* 42(1):175–192.

Porter, R.; Nudelman, E.; and Shoham, Y. 2004. Simple search methods for finding a Nash equilibrium. In *AAAI*, 664–669.

Sandholm, T.; Gilpin, A.; and Conitzer, V. 2005. Mixed-integer programming methods for finding Nash equilibria. In *AAAI*, 495–501.

van Damme, E. 1991. *Stability and Perfection of Nash Equlibria*. Springer.

von Stengel, B.; van den Elzen, A.; and Talman, D. 2002. Computing normal form perfect equilibria for extensive two-person games. *ECONOMETRICA* 70(2):693–715.

von Stengel, B. 2007. *Algorithmic Game Theory*. Cambridge. chapter Equilibrium computation for two-player games in strategic and extensive form, 53–78.