The most common solution concept for a strategic interaction situation is the Nash equilibrium, in which no agent can do better by deviating unilaterally. However, the Nash equilibrium underlays on the assumption of common information that is hardly verified in many practical situations. When information is not common, rational agents are assumed to learn from their observations to derive beliefs over their opponents’ play and payoffs. In these situations, there are steady states composed of beliefs and strategies in which the strategies do not constitute a Nash equilibrium. These stable states are called in the game theory literature self-confirming equilibria. They are such that every agent plays the best response to her beliefs and these are correct on the equilibrium path, while off the equilibrium path they may be incorrect. We present some mathematical programming formulations for computing self-confirming equilibria and its refinements in two-player extensive-form games and we study their properties.

Abstract

The most common solution concept for a strategic interaction situation is the Nash equilibrium, in which no agent can do better by deviating unilaterally. However, the Nash equilibrium underlays on the assumption of common information that is hardly verified in many practical situations. When information is not common, rational agents are assumed to learn from their observations to derive beliefs over their opponents’ play and payoffs. In these situations, there are steady states composed of beliefs and strategies in which the strategies do not constitute a Nash equilibrium. These stable states are called in the game theory literature self-confirming equilibria. They are such that every agent plays the best response to her beliefs and these are correct on the equilibrium path, while off the equilibrium path they may be incorrect. We present some mathematical programming formulations for computing self-confirming equilibria and its refinements in two-player extensive-form games and we study their properties.

Introduction

Non-cooperative game theory provides elegant models and solution concepts for situations wherein rational agents can strategically interact (Fudenberg and Tirole 1991). The central solution concept is Nash equilibrium: it defines how agents should act in settings where an agent’s best strategy may depend on what the others do. One of the main critiques to the employment of the Nash equilibrium concept in many practical situations lays on the assumption of common information. That is, when information is complete, common information means that each agent knows the private values of her opponents and knows that her opponents know her private values and so on. When the information is uncertain, the constraint of common information is stronger: it requires that each agent has a Bayesian prior over her opponents and that such prior is common. This assumption seems to be unrealistic in a large number of practical situations (e.g., auctions and negotiations).

Basically, a Nash equilibrium provides some prescriptions to the agents without explaining how agents can have formed a common prior. This prior formation process is customarily studied in the literature as a learning process in which each agent has some (generally incorrect) beliefs over the behaviors of her opponents and, by repeatedly observing the moves of the opponents, adjusts her beliefs by means of a learning algorithm. The crucial point is that when we study the problem of finding the agents’ optimal strategies incorporating the problem of prior formation, we have that some steady states may not be Nash equilibria. More precisely, game theory provides some solution concepts, called self-confirming equilibria (Dekel, Fudenberg, and Levine 1999; Fudenberg and Levine 1993), that are appropriate for these situations (regardless of the specific learning algorithm adopted by the agents). The basic idea behind the concept of self-confirming equilibrium (from here on SCE) is that the agents’ beliefs need to be correct only at the information-sets reached on the equilibrium path. Since the agents do not observe the behavior of their opponents off the equilibrium path, their beliefs can be incorrect at those information-sets. The set of SCEs contains the set of the Nash equilibria, a Nash equilibrium being a SCE in which the beliefs are correct at every information-set. While in strategic-form games, all the SCEs are also Nash equilibria (all the information-sets being on the equilibrium path), this is not the case for extensive-form games. In these games, a SCE may not be a Nash equilibrium. The game theory literature provides also several refinements of SCEs to capture different situations (e.g., when a agent can be drawn from a population of individuals or when an agent have some information about the payoffs of the opponents).

In this paper, we focus on two-player extensive-form games and we study the problem of computing a SCE and its refinements regardless of the learning algorithms adopted by the agents. We notice that SCEs have been already considered in some previous works both on learning algorithms (Monderer and Tennenholtz 2007; Wellman and Hu 1998) and practical applications (auctions) (Osepyayshvil et al. 2005), but, to the best of our knowledge, no work discussed how a SCE and its refinements can be computed. We extend the algorithms for finding a Nash equilibrium (Shoham and Leyton-Brown 2008) to compute a SCE and we study their properties (e.g., computational complexity). (The lack of a testbed for extensive-form games (GAMUT (Nudelman et al. 2004) generates only strategic-
form games) pushed us to produce a theoretic/algorhythmic paper and omit any experimental evaluation.) In our mind, the applications of our algorithms are two. They can be used to compute an equilibrium for a given non-cooperative problem (as it happens for Nash equilibrium), or they can be used within learning algorithms with the aim to guide the converge to an equilibrium or to evaluate their performance.

**Extensive-Form Games and Solving Algorithms**

**Definitions**

A finite perfect-information extensive-form game (Shoham and Leyton-Brown 2008) is a tuple \((N, A, V, T, \iota, \rho, \chi, u)\), where: \(N\) is the set of \(n\) agents, \(A\) is a set of actions, \(V\) is the set of decision nodes of the game tree, \(T\) is the set of terminal nodes of the game tree, \(\iota : V \rightarrow N\) is the agent function that specifies the agent that acts at a given decision node, \(\rho : V \rightarrow \mathcal{P}(A)\) returns the actions available to agent \(\iota(w)\) at decision node \(w\), \(\chi : V \times A \rightarrow V \cup T\) 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 \(u_i : T \rightarrow \mathbb{R}\). An extensive-form game is with imperfect-information when some action of some agent is not perfectly observable by the agent’s 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 \(I = (I_1, \ldots, I_n)\) is a partition of set \(V = \{w \in V : \iota(w) = i\}\) with the property that \(\rho(w) = \rho(w')\) whenever there exists a \(j\) for which \(w, w' \in I_{i,j}\). The sets \(I_{i,j}\)s are called information-sets. We focus on games with perfect recall, where every agent recalls all the actions undertaken by her and by the opponents (this assumption induces some constraints over \(I\), omitted here for reason of space).

A pure strategy \(\sigma_i\) is a plan of actions specifying one action for each information-set of agent \(i\). A mixed strategy \(\sigma_i\) is a randomization over pure strategies (plans). An alternative representation is given by behavioral strategies. They are 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 \(\sigma_i\) assigns each information-set \(h \in I_i\) a probability distribution over the actions available at \(h\). With perfect recall, the two representations (plans and behavioral) are equivalent.

Each agent has a system of belief providing her beliefs over the behavior of the opponents. We call \(\mu_i^j\) the system of belief of agent \(i\) over strategy \(\sigma_j\) of agent \(j\). The beliefs are correct if \(\mu_i^j = \sigma_j\) for every \(i\) and \(j\). We call \(\mu_i = \{\mu_i^j : \text{for all } j \neq i\}\). A pair \((\sigma, \mu)\), where \(\sigma\) is the agents’ strategy profile and \(\mu\) is the set of all the agents’ \(\mu_i\)s, is called assessment.

Under the assumption that information is complete and common we can define the concept of Nash equilibrium as an assessment \((\sigma, \mu)\) such that for all \(i \in N\): \(\sigma_i\) is a best response to \(\mu_i\), and the beliefs provided by \(\mu\) are correct.

It is well known that in extensive-form games some Nash equilibria may be not reasonable with respect to the sequential structure of the game. The concept of sequential equilibrium refines the concept of Nash equilibrium removing these equilibria (Kreps and Wilson 1982). A sequential equilibrium is an assessment \((\sigma, \mu)\) such that for all \(i \in N\): strategy \(\sigma_i\) is sequentially optimal with respect to \(\mu_i\) (in the sense of backward induction), and there exists a sequence of fully mixed strategies \(\tilde{\sigma}_{i,m}\) such that for all agents is \(\lim_{m \rightarrow \infty} \tilde{\sigma}_{i,m} = \sigma_i\) and the limit of the sequence of beliefs derived from the fully mixed strategies by using the Bayes rule converges to \(\mu\). The second condition is called Kreps and Wilson consistency. It can be easily seen that this condition entails that the beliefs are correct. (The use of fully mixed strategies is accomplished to characterize beliefs off the equilibrium path where the Bayes rule cannot be applied.)

**The Sequence Form**

The computation of a Nash equilibrium in an extensive-form game can be easily accomplished by transforming the game in normal-form and then by computing a Nash equilibrium. We recall that the normal-form of an extensive-form game is a matrix-based representation where the agents’ actions are plans of actions in the extensive-form game. However, the normal-form is exponential in the size of the extensive-form game, making hard the computation of a Nash equilibrium. One way to avoid this problem is to work directly on the extensive-form representation by employing behavioral strategies. This can be efficiently accomplished by using an alternative representation called sequence-form (Koller, Megiddo, and von Stengel 1996). This is a sparse matrix-based representation where: (i) each agent’s actions are (terminal and non-terminal) sequences \(q\) of her actions in the game tree (consider Fig. 1, \(q = R\) is a non-terminal sequence of agent 1, while \(q = RL1\) is terminal); (ii) given a profile of sequences \(q = (q_1, \ldots, q_n)\) where \(q_i\) is the sequence of agent \(i\), if \(q\) leads to a terminal node, then the agents’ payoffs are their utilities over such a node, otherwise the payoffs are null; and (iii), called \(q' = q|a\) the sequence obtained by extending \(q\) with action \(a\) (e.g., \(q' = RL1\) with \(q = R\) and \(a = L1\)), the probability of a sequence \(q\) is equal to the sum of the probabilities of the sequences that extend it. Once a game is solved in the sequence-form, the behavioral strategies can be easily computed.

We report the sequence-form constraints in mathematical programming fashion for two-player games. We explicitly consider the agents’ beliefs (even if they can be omitted, being correct in a Nash equilibrium) because we shall use them in the next section. We denote the probabilities with which agents make their sequences (i.e., \(\sigma\)) by \(p_i(q)\)s, and we denote the agents’ systems of beliefs (i.e., \(\mu\)) by \(\hat{p}_i(q)\)s, where \(\hat{p}_i(q)\) is the belief of agent \(i\) over the strategy of agent \(j\). We denote by \(Q_i\) the set of sequences of agent \(i\), by \(I_q\) the set of the information-sets of agent \(i\) reachable from sequence \(q \in Q_i\) (consider Fig. 1, \(I_R = \{1, 2\}\)), and by \(h_q\) a generic information-set belonging to \(I_q\). Strategies \(p_i(\cdot)\)s and beliefs \(\hat{p}_i(\cdot)\)s are subject to the following constraints (\(\emptyset\) is the empty sequence):
We introduce two further constraints that we shall use in what follows. We denote by \( v_i(q) \) the utility agent \( i \) receives from taking sequence \( q \) and by \( \pi_a \), the utility an agent expects to gain when it plays at information-set \( h \) (i.e., the largest expected utility among those of the sequences \( q \) where \( a \) is available at \( h \)).

\[
v_i(q) = \sum_{q' \in \mathcal{Q}_i} \hat{p}_{-i}(q') U_i(q, q') + \sum_{h \in I_q} \pi_a \quad \forall i \in N, q \in \mathcal{Q}_i \tag{7}
\]

Constraints (7) state that the agent \( i \)'s expected utility from sequence \( q \) is equal to the sum of the expected utility over the terminal outcomes (if reached) and of the expected utilities of the information-sets reachable by performing \( q \) (if exist).

Constraints (8) state that the utility at \( h \) is not smaller than the utility of all the sequences \( q \) where \( a \) is available at \( h \).

### Computing Equilibria

The computation of a Nash equilibrium is essentially a feasibility mathematical programming problem (Shoham and Leyton-Brown 2008) that always admits at least a solution in mixed strategies. It is known to be \textsc{P}-complete (Daskalakis, Goldberg, and Papadimitriou 2006). We recall that it is generally believed that \textsc{P}-complete and that computing a Nash equilibrium requires exponential time in the worst case. For a two-player game there are three solving algorithms. LH provides a linear complementarity mathematical programming formulation and an algorithm based on pivoting techniques (Lemke and Howson 1964). While LH is applicable to solve an extensive-form game in normal form, it cannot be applied to solve it in sequence-form. In this case, a generalization of LH, called Lemke’s algorithm and applicable to any linear complementarity problem, can be used (Lemke 1978). SGC provides a mixed integer linear mathematical programming formulation (Sandholm, Gilpin, and Conitzer 2005). PNS provides an algorithm based on support enumeration (Porter, Nudelman, and Shoham 2004). SGC and PNS have been never used for extensive-form games. Below we discuss how they can be extended to these games because they play a crucial role for the computation of a SCE.

We report the mathematical programming formulations of the above algorithms. At first, we require that the beliefs are correct:

\[
\hat{p}_i(q) = p_i(q) \quad \forall i \in N, q \in \mathcal{Q}_i \tag{9}
\]

The linear complementarity problem (called ELC) is:

\[
\begin{align*}
\text{constraints \ (1), (2), (3), (4), (5), (6), (7), (8), (9)} \\
(\nabla_h - v_i(q|a))p_i(q|a) = 0 & \quad \forall i \in N, q|a \in \mathcal{Q}_i, a \text{ at } h, h \in I_q \tag{10}
\end{align*}
\]

Constraints (10) state that, if sequence \( q|a \) is played with strictly positive probability, then its utility \( v_i(q|a) \) must be equal to the utility \( \nabla_h \) of the information-set \( h \) at which \( a \) is played (i.e., at every information-set an agent plays her best actions).

The mixed integer linear problem (called ESCG) is based on binary variables \( s_i(q) \in \{0, 1\} \) such that, when sequence \( q \) is in the support of agent \( i \) (i.e., \( p_i(q) > 0 \), \( s_i(q) = 1 \)). We notice that, by the sequence-form constraints, we can have that \( p_i(q) > 0 \) even when \( q \) is played off the equilibrium path (e.g., consider Fig. 1, when \( p_1(L) = 1, p_2(r) = 1 \), \( r \) is off the equilibrium path). The ESCG formulation is:

\[
\begin{align*}
\text{constraints \ (1), (2), (3), (4), (5), (6), (7), (8), (9)} \\
\pi_a \leq v_i(q|a) + M(1 - s_i(q)) & \quad \forall i \in N, q|a \in \mathcal{Q}_i, a \text{ at } h, h \in I_q \tag{11} \\
p_i(q) \leq s_i(q) & \quad \forall i \in N, q \in \mathcal{Q}_i \tag{12}
\end{align*}
\]

where \( M \) is an arbitrarily large constant. Constraints (11), with constraints (8), state that, if \( s_i(q) = 1 \), then \( \pi_a = v_i(q|a) \); constraints (12) state that, if \( s_i(q) = 0 \), then \( q \) is played with a probability of zero.

While in ESGC the problems of checking whether an equilibrium exists with a given support and of scanning the supports are solved together in a mathematical programming fashion, in EPNS they are separated. The first problem is solved by enumeration and heuristics and the second one is formulated as a linear mathematical programming problem that is exactly the ESGC formulation in which the values of \( s_i(q) \)s are fix. Essentially, every problem instance that can be formulated as an ESCG problem can be also formulated as an EPNS problem. We limit our discussion to mixed-integer formulations (i.e., ESGC), it being always possible to provide a corresponding EPNS formulation.

The unique known algorithm to compute a sequential equilibrium is a variation of the Lemke’s algorithm (Milersen and Sorensen 2006). Basically, a perturbation \( \epsilon \geq 0 \) is introduced into the ELC formulation and a strategy satisfying the problem both for \( \epsilon = 0 \) and for an arbitrarily small strictly positive value \( \epsilon > 0 \) is found. The formulation is:

\[
\begin{align*}
\text{constraints \ (1), (2), (3), (4), (5), (6), (7), (8), (9)} \\
(\nabla_h - v_i(q|a))(p_i(q|a) - \epsilon|q|) = 0 & \quad \forall i \in N, q|a \in \mathcal{Q}_i, a \text{ at } h, h \in I_q \tag{13} \\
p_i(q) \geq \epsilon|q| & \quad \forall i \in N, q \in \mathcal{Q}_i \tag{14}
\end{align*}
\]

where \( |q| \) is the length of \( q \). Forcing the perturbation given by constraints (14) assures the solution to be a quasi-perfect equilibrium that is a concept stronger than the sequential equilibrium. The solving algorithm is described in (Milersen and Sorensen 2006). Finding a sequential equilibrium is \textsc{P}-complete.
Self-Confirming Equilibria Computation

Equilibrium Concepts

The basic self-confirming equilibrium solution concept captures the situation in which agents have no \emph{a priori} information about opponents’ strategy or payoffs and learn (in some way) from their observations over the actions played by the opponents. The aim is the study of assessments \((\sigma, \mu)\)s that are steady states. Essentially, they generalize the concept of Nash equilibrium to the case in which information is not common. Indeed, while Nash equilibrium provides a prescription on how rational agents should play, self-confirming equilibrium provides a prescription on what are the beliefs of rational agents and on how they should play. A self-confirming equilibrium requires that agents correctly forecast the actions that the opponents will take only on the equilibrium path, an agent deriving information on her opponents’ behavior only from her observations. Off the equilibrium path the agents’ beliefs can be arbitrary.

Fudenberg and Levine provide some concepts of self-confirming equilibrium (Dekel, Fudenberg, and Levine 1999; Fudenberg and Levine 1993). They distinguish between \emph{unitary} and \emph{heterogeneous} self-confirming equilibria. The idea is that each agent can be characterized by a population of individuals and, every time the game is repeated, a specific individual plays. Potentially, different individuals can have different beliefs and different optimal strategies. In unitary SCEs (from here on USCE), the population is composed of a single individual (therefore each agent has exactly one belief and one optimal strategy). In heterogeneous SCEs (from here on HSCE), the population is composed of multiple individuals. We notice that this last model perfectly apply to practical economic situations, such as, e.g., bargaining and auctions, where different sellers are continuously matched with different buyers.

Fudenberg and Levine show that SEs \(\subseteq\) NEs \(\subseteq\) USCEs \(\subseteq\) HSCEs (where SE and NE mean sequential and Nash equilibrium respectively). They provided also two refinements (applicable to both USCEs and HSCEs).

\textbf{Consistent SCE} captures situations in which agents are occasionally matched with “crazy” opponents, so that even if they stick to their equilibrium strategy themselves, they eventually learn the strategy at all information-sets that can be reached if their opponents deviate. It requires that each agent correctly predicts strategy at all information-sets that can be reached when the agents’ opponents, but not the agents themselves, deviate from their equilibrium strategies. In each two-player game, every SCE is consistent. For the sake of presentation, from here on we omit the adjective ‘consistent’, our work focusing on two-player games.

\textbf{Rationalizable SCE} captures the situations in which agents have some information about the payoffs of their opponents and use it in the sense of rationalizability. Technically speaking, it requires that the agents’ strategies are sequentially rational with respect to the beliefs (as in sequential equilibria) and beliefs are correct on the equilibrium path and on the reachable information-sets (i.e., the information-sets that an agent can reach by perturbing its own strategy and keeping fixed the opponent’s strategy).

In (Fudenberg and Levine 1993), the authors show that an USCE may not be a Nash and that SEs \(\subseteq\) rationalizable USCEs, that there can be rationalizable USCEs that are not NEs and there can be NEs that are not rationalizable USCEs.

\textbf{Unitary SCE}

Formally, an USCE is an assessment \((\sigma, \mu)\) such that for every agent \(i \in N:\)

\begin{itemize}
  \item \textbf{strategy }\sigma_i \text{ is optimal with respect to some } \mu_i,
  \item \textbf{all the beliefs prescribed by } \mu_i \text{ are correct on the equilibrium path.}
\end{itemize}

That is, we need to relax constraints (9), forcing \(\hat{p}_i(q) = p_i(q)\) only if \(q\) is on the equilibrium path. In order to check whether or not a sequence \(q\) is on the equilibrium path we need to consider the strategies of both the agents. Indeed, as discussed in the previous sections, in the sequence-form a sequence \(q\) can present \(p(q) > 0\) even if it is played off the equilibrium path. Basically, a sequence \(q\) of agent \(i\) is played on the equilibrium path if and only if \(q\) is played with strictly positive probability and, called \(q = q^*\), there exists a sequence \(f(q)\) of agent \(i\) played with strictly positive probability that leads to the information-set where agent \(i\) plays \(a\). Formally, \(\hat{p}_i(q) = p_i(q)\) if \(p_i(q) > 0\) and \(p_{-i}(f(q)) > 0\) for at least a \(f(q)\) (given a \(q\) there can be multiple \(f(q)\)s, e.g., consider Fig. 1, \(q = RL_1\) is on the path if \(p_1(RL_1) > 0\) and \(p_2(f(RL_1)) > 0\) \& \(f(RL_1) \in \{l\}\), and \(q = l\) is on the path if \(p_2(\{l\}) > 0\) and \(p_1(f(\{l\})) > 0\) \& \(f(\{l\}) \in \{M, R\}\).

We extend the mathematical programming formulations provided in the previous sections to find a USCE. At first, we consider the ELC formulation. This formulation cannot be extended to find a USCE by introducing exclusively linear complementarity constraints. This is because, checking whether or not a sequence \(q\) is on the path is intrinsically quadratic due to the presence of the operator ‘and’ between conditions \(p_i(q) > 0\) and \(p_{-i}(f(q)) > 0\). A non-linear complementarity constraint (non-solvable by Lemke’s algorithm and requiring different algorithms such as Scarf’s (Shoham and Leyton-Brown 2008)) can be:

\[ p_i(q)p_{-i}(f(q))\hat{p}_i(q) - p_i(q) = 0 \quad \forall i \in N, q \in Q_i, f(q) \in Q_{-i} \]

(We cannot exclude that an alternative linear formulation exists, anyway we have not been able to find it.) Instead, the ESCG (and, consequently, the EPNS) formulation(s) can be extended. The ESCG can be easily modified by substituting constraints (9). More precisely, the ESCG formulation finding a USCE is:

\begin{align*}
\hat{p}_i(q) &\leq p_i(q) + M(2 - s_{-i}(f(q)) - s_i(q)) \quad \forall i \in N, q \in Q_i, \quad (15) \\
\hat{p}_i(q) &\geq p_i(q) - M(2 - s_{-i}(f(q)) - s_i(q)) \quad \forall i \in N, q \in Q_i, \quad (16)
\end{align*}

Constraints (15) and (16) force beliefs to be correct when \(s_{-i}(f(q)) = 1\) and \(s_i(q) = 1\).
We know that \( p_i(\cdot) \)s and \( p_{-i}(\cdot) \)s may not constitute a Nash equilibrium (e.g., see Example 1 in the following example section). However, surprisingly, we have that \( \tilde{p}_i(\cdot) \)s and \( \tilde{p}_{-i}(\cdot) \)s constitute a Nash.

**Theorem 1** Given a USCE, expressed as a set of strategies \( p_i(\cdot) \)s and beliefs \( \tilde{p}_i(\cdot) \)s, strategies \( p_i'(\cdot) = \tilde{p}_i(\cdot) \) constitute a Nash equilibrium.

**Proof**. By definition, on the equilibrium path, the actions played with positive probability in \( p_i'(\cdot) = \tilde{p}_i(\cdot) \) are best responses to \( p_{-i}(\cdot) \). \( \tilde{p}_i(\cdot) \) being the same of \( p_i(\cdot) \). Off the equilibrium path, the actions played with positive probability in \( p_i'(\cdot) \) are potentially different from those in \( p_i(\cdot) \), but, providing a utility of zero, agent \( i \) cannot gain more by deviating from them. Therefore, \( p_i'(\cdot) = \tilde{p}_i(\cdot) \) is a best response to \( p_{-i}(\cdot) \) and then \( (p_i'(\cdot), p_{-i}'(\cdot)) \) is a Nash equilibrium. \( \square \)

As a result, given a USCE, we can find a Nash equilibrium in constant time. We can state the following theorem, whose proof is a trivial application of Theorem 1.

**Corollary 2** For any USCE there exists a Nash equilibrium that induces the same randomization over the outcomes.

We focus on the computational complexity of finding a USCE.

**Theorem 3** The problem of computing a USCE in a two-player game (called USCE-2) is PPAD-complete.

**Proof**. USCE-2 is in PPAD because any USCE-2 instance admits at least one solution and, given an assessment, it can be verified in polynomial time in the size of the game whether or not it is a solution. The PPAD-completeness can be proved by reduction to NASH (the problem of computing a Nash equilibrium). A trivial reduction is due to the fact that in strategic-form games every Nash equilibrium is a USCE. A less-trivial reduction is due to Theorem 1. \( \square \)

**Heterogeneous SCE**

Formally, an HSCE is an assessment \((\sigma, \mu)\) such that for every agent \( i \) in \( N \):

- each pure strategy \( j \) in \( \sigma_i \) is optimal with respect to some (potentially different) \( \mu_i \) (denoted by \( \mu_{i,j} \)),
- the beliefs prescribed by \( \mu_{i,j} \) are correct on the equilibrium path identified by pure strategy \( j \) in \( \sigma_i \).

According to the above definition, we need to introduce different (heterogeneous) beliefs for each agent. More precisely, we define \( \tilde{p}_{-i,q}(q') \) as the belief of agent \(-i\) over the probability with which agent \( i \) plays sequence \( q' \in Q_i \), where the parameter is \( q \in Q_{-i} \). For each \( \tilde{p}_{i,q}(q') \) the sequence-form constraints must hold:

\[
\tilde{p}_{i,q}(q') = \sum_{q' \in Q_i} \tilde{p}_i(q'|q) \quad \forall i \in N, q \in Q_{-i} \quad (17)
\]

\[
\tilde{p}_{-i,q}(q') = \sum_{h \in H_{-i}} \tilde{p}_i(q'|h) \quad \forall i \in N, q' \in Q_i, h \in H_{-i} \quad (18)
\]

\[
\tilde{p}_{i,q}(q') = 0 \quad \forall i \in N, q' \in Q_i, q \in Q_{-i} \quad (19)
\]

Given that the beliefs are parameterized with respect to sequence \( q \) and the expected utility of playing a sequence \( q' \) depends on the beliefs, we need to specify the parameter \( q \) in the expected utility formula. That is, we denote by \( v_{i,q}(q') \) the expected utility received by agent \( i \) when she plays sequence \( q' \) and the beliefs are those parameterized with respect to sequence \( q \) (i.e., \( \tilde{p}_{i,-q}(\cdot) \)s). We can easily check whether or not a sequence \( q \) is a never-best-response (i.e., there is not any belief such that \( q \) is a best response). For simplicity, we safely limit to terminal sequences. A non-terminal sequence \( q \) is not a never-best-response if there exists at least a terminal sequence \( q' \) extending \( q \) that is not a never-best-response. We denote by \( Q_i^* \) the set of terminal sequences of agent \( i \). A sequence \( q \in Q_i^* \) is not a never-best-response if there are some \( \tilde{p}_{-i,q}(q''|n) \)s such that:

\[
v_{i,q}(q') = \sum_{q'' \in Q_{-i}} \tilde{p}_{-i,q}(q''|n) U_i(q', q'') \quad \forall i \in N, q, q' \in Q_i^* \quad (20)
\]

\[
v_{i,q}(q) \geq v_{i,q}(q') \quad \forall i \in N, q, q' \in Q_i^* \quad (21)
\]

According to the definition of HSCE, we need to constrain the beliefs \( \tilde{p}_{i,q}(\cdot) \)s to which a sequence \( q \) is a best response to be correct on the equilibrium path identified by \( q \), e.g., consider Fig. 1, beliefs \( \tilde{p}_{2,L}(\cdot) \) can be any, no information-set of agent 2 being on the equilibrium path identified by sequence \( q = L \), instead beliefs \( \tilde{p}_{2,M}(\cdot) \) must be correct at least at information-set 2.1, this information-set being on the equilibrium path identified by sequence \( q = M \). We state the problem of finding a HSCE as a mixed-integer linear programming problem as follows:

<table>
<thead>
<tr>
<th>Constraints</th>
<th>( i )</th>
<th>( q )</th>
<th>( q' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4) &amp; (5) &amp; (6) &amp; (12) &amp; (17) &amp; (18) &amp; (19) &amp; (20)</td>
<td>( i \in N )</td>
<td>( q \in Q_i^* )</td>
<td>( q' \in Q_i^* )</td>
</tr>
<tr>
<td>(22)</td>
<td>( i \in N )</td>
<td>( q \in Q_i^* )</td>
<td>( q' \in Q_i^* )</td>
</tr>
<tr>
<td>(23)</td>
<td>( i \in N )</td>
<td>( q \in Q_i^* )</td>
<td>( q' \in Q_i^* )</td>
</tr>
<tr>
<td>(24)</td>
<td>( i \in N )</td>
<td>( q \in Q_i^* )</td>
<td>( q' \in Q_i^* )</td>
</tr>
</tbody>
</table>

Constraints (22) with constraints (12) force sequences \( q \)s to be played with a probability of zero if beliefs \( \tilde{p}_{i,-q}(\cdot) \)s are such that \( q \) is not a best response. Constraints (23) and (24) force \( \tilde{p}_{i,q}(\cdot) \)s to be correct only on the equilibrium path identified by \( q \). Differently from what happens for the computation of a USCE, there is a straightforward linear-complementarity formulation for finding a HSCE. This is because the equilibrium path identified by a single sequence \( q \in Q_i \) depends only on \( q \) and the strategy of agent \(-i\), but not on the strategy of agent \( i \). Call \( \Pi_{i,q} \) the largest expected utility among \( v_{i,q}(q') \) for all \( q' \in Q_i \). The formulation for finding a HSCE is:

<table>
<thead>
<tr>
<th>Constraints</th>
<th>( i )</th>
<th>( q )</th>
<th>( q' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4) &amp; (5) &amp; (6) &amp; (17) &amp; (18) &amp; (19) &amp; (20)</td>
<td>( i \in N )</td>
<td>( q \in Q_i^* )</td>
<td>( q' \in Q_i^* )</td>
</tr>
<tr>
<td>(25)</td>
<td>( i \in N )</td>
<td>( q \in Q_i^* )</td>
<td>( q' \in Q_i^* )</td>
</tr>
<tr>
<td>(26)</td>
<td>( i \in N )</td>
<td>( q \in Q_i^* )</td>
<td>( q' \in Q_i^* )</td>
</tr>
<tr>
<td>(27)</td>
<td>( i \in N )</td>
<td>( q \in Q_i^* )</td>
<td>( q' \in Q_i^* )</td>
</tr>
</tbody>
</table>
Constraints (25) force $\mathbf{v}_{i,q}$ to be the largest expected utility among $v_{i,q}(q')$ for all $q' \in Q_i$; constraints (26) force a sequence $q'$ to be played only if it is a best response to $\hat{p}_{i,q}(\cdot)$; constraints (27) force beliefs $\hat{p}_{i,q}(\cdot)$ to be correct on the equilibrium path identified by $q$. We notice that combining the USCE’s constraints with the HSCE’s constraints, we can capture asymmetric situations where there is a single individual for agent $i$ and a population for agent $-i$.

We discuss the relationship between HSCEs and USCEs.

**Theorem 4** An HSCE induces a randomization over outcomes that may not occur in any USCE.

**Proof.** The proof is by an example. In particular, see Example 2 in the following example section.

We focus on the computational complexity of HSCE (the proof is based on the first reduction used in the proof of Theorem 4).

**Theorem 5** The problem of computing an HSCE in a two-player game (called HSCE-2) is PPAD-complete.

**Rationalizable SCE**

We initially consider rationalizable USCEs. Formally, a RUSCE is an assessment $(\sigma, \mu)$ such that for every $i \in \hat{N}$:

- strategy $\sigma_i$ is sequentially optimal with respect to some $\mu_i$,
- all the beliefs prescribed by $\mu_i$ are correct on the equilibrium path and on the off the equilibrium path information-sets reachable by agent $i$ by perturbing her strategy.

Since we must assure agents’ sequentially rationality, we resort to the formulation to find a sequential equilibrium. The formulation for finding a RUSCE is:

\[
\begin{align*}
\text{constraints} \ (1), \ (2), \ (3), \ (4), \ (5), \ (6), \ (7), \ (8), \ (14) \ & \\
\hat{p}_{i,q}(q) \geq \mathbf{v}^i_{q} & \forall i \in \hat{N}, q \in Q_i \ (28) \\
(p_i(q) - \mathbf{v}^i_{q})(\hat{p}_{i,q}(q|\nu) - p_i(q|\nu)) = 0 & \forall i \in \hat{N}, q \in Q_i \ (29)
\end{align*}
\]

Constraints (28) force every belief to have strictly positive probability, granting the sequential rationality with respect to the beliefs; constraints (29) force beliefs at off the equilibrium path reachable information-sets to be correct. Combining the USCE’s constraints with the RUSCE’s constraints we can capture asymmetric situations where only agent $i$ has some information over agent $-i$’s payoffs.

The authors state in (Dekel, Fudenberg, and Levine 1999) that the idea behind rationalizable USCEs can be extended to HSCEs, but they do not discuss how. We study this extension, showing that the sets of RUSCEs and RHSCEs are essentially the same. Then, $\mu_{i,j} = \mu_{i,k}$ on the equilibrium path and at the reachable information-sets for all $j, k$. Beliefs $\mu_{i,j}$ can differ only at non reachable information-sets, but these beliefs do not affect the computation of the agents’ best response. Therefore, any assessment $(\sigma', \mu')$ with $\sigma' = \sigma$ and $\mu'_i = \mu_{i,j}$ for any $j$ is a RUSCE.

We focus on the computational complexity of finding a RSCE with two agents (the proof is trivial, because SEs $\subseteq$ RUSCEs $\subseteq$ USCEs and the problems of finding both a USCE and a SE is PPAD-complete).

**Theorem 7** The problem of computing a RSCE in a two-player game (called RSCE-2) is PPAD-complete.

**Examples**

We depict in Fig. 1 an example of two-player extensive-form game with imperfect information. In what follows we report some equilibria specifying strategies $p_i(\cdot)$s and beliefs $\hat{p}_i(\cdot)$s. For reasons of space, we report only the non-null probabilities.

![Figure 1: Example of two-player extensive-form game. (x,y denotes the y-th information-set of agent x.)](image)

The NEs in pure strategies are: $\sigma = (p_1(M) = 1, p_2(lr_1) = 1)$, $\sigma = (p_1( RR_1 ) = 1, p_2( r ) = 1)$, and $\sigma = (p_1( RL_1 ) = 1, p_2( ll_1 ) = 1)$. The unique SE in pure strategies is: $\sigma = (p_1( RL_1 ) = 1, p_2( ll_1 ) = 1)$. 


Example 1. A USCE that is not a NE is: \( \sigma = (p_1(L) = 1, p_2(l_1) = 1) \) with \( \mu = (\tilde{p}_1(L) = 1, \tilde{p}_2(l_1) = \tilde{p}_2(r) = \frac{1}{2}, \tilde{p}_2(l_2) = \frac{1}{4}) \). The agents’ strategies are not optimal, agent 1 gaining more by playing \( q = RL_1 \), while the beliefs are confirmed on the equilibrium path.

Example 2. An HSCE that is not a USCE: \( \sigma = (p_1(L) = \frac{1}{2}, p_1(M) = \frac{1}{12}, p_1(RL_1) = \frac{5}{12}, \tilde{p}_2(l_1) = \frac{1}{2}, \tilde{p}_2(r) = \frac{1}{2}) \) where sequence \( q = L \) is a best response to beliefs \( \mu_{1,L} = (\tilde{p}_2,L(l_1) = \tilde{p}_2,l(r) = \tilde{p}_2,L(m) = \frac{1}{2}) \) that are all incorrect, agent 2 playing only off the equilibrium path; sequence \( q = M \) is a best response to beliefs \( \mu_{1,M} = (\tilde{p}_2,M(l_1) = \tilde{p}_2,L(r) = \tilde{p}_2,L(m) = \frac{1}{2}) \) that are all incorrect, agent 2 playing only off the equilibrium path; sequence \( q = RL_1 \) is a best response to beliefs \( \mu_{1,RL_1} = (\tilde{p}_2,RL_1(l_1) = \tilde{p}_2,RL_1(r) = \frac{1}{2}) \) that are correct everywhere, all the information-sets of agent 2 being on the equilibrium path; sequence \( q = ll_1,lr_1 \) being off the equilibrium path; sequence \( q = RL_1 \) is a best response to beliefs \( \mu_{1,RL_1} = (\tilde{p}_2,RL_1(l_1) = \tilde{p}_2,RL_1(r) = \frac{1}{2}) \) that are correct everywhere, all the information-sets of agent 1 being on the equilibrium path; and sequence \( q = r \) is a best response to beliefs \( \mu_{2,r} = (p_1(M) = \frac{1}{12}, \tilde{p}_1,r(RR_1) = \frac{5}{12}) \) that are correct on \( q = L, M, R \) being on the equilibrium path, but incorrect on \( q = ll_1,lr_1 \) being off the equilibrium path. Notice that \( (M, r) \) does not occur in any USCE.

Example 3. A NE that is a RSCE is \( \sigma = (p_1(RR_1) = 1, p_2(r) = 1) \) with \( \mu = (\tilde{p}_1(RR_1) = 1, \tilde{p}_2(r) = 1, \tilde{p}_2(lr_1) \rightarrow 1 \) in perturbation). Fixed \( \sigma_2 \), there is no perturbation of agent 1 such that she can observe \( \sigma_2 \) at information-set 2.2 and then the beliefs of agent 1 on the behavior of agent 2 at information-set 2.2 can be any. Fixed \( \sigma_1 \), there is a perturbation of agent 2 such that she can observe \( \sigma_1 \) at information-set 1.2 and then the beliefs of agent 2 on the behavior of agent 1 at 1.2 must be correct.

Example 4. A NE that is not a RSCE is \( \sigma = (p_1(M) = 1, p_2(lr_1) = 1) \). This is because, in perturbation agent 2 takes \( q = ll_1 \) instead of \( q = lr_1 \). It can be shown that there not exists any RSCE when \( p_1(M) = 1 \). Indeed, if \( p_1(M) = 1 \), then, by best response, \( p_2(l_1) = 1 \) (\( p_2(lr_1) = 1 \) is removed by perturbation). Fixed \( \sigma_2 \), there exists a perturbation of agent 1 such that she can observe \( \sigma_2 \) at information-set 2.2 and then the beliefs of agent 1 on the behavior of agent 2 at 2.2 must be correct. Then, by sequential rationality, agent 1 knows that taking \( q = RL_1 \) gives her more than \( q = M \).

Conclusions and Future Works

In a large number of practical applications, the assumption of common information is hardly verified, making not justifiable the adoption of the Nash equilibrium concept. The game theory literature provides a solution concept, called self-confirming equilibrium (SCE), that appropriately captures the situations where agents are rational and form their beliefs by observing the behaviors of their opponents without having a common prior. In this paper, we provide some mathematical programming formulations to compute different notions of SCE and discuss their properties.

In future works, we shall study the computation of SCEs when there is uncertainty both in the situations where agents know it and where they do not know it following the extensions proposed in (Ceppi, Gatti, and Basilico 2009). We are also interested in experimentally evaluating the computational time needed for computing SCEs and comparing the performance of the different formulations we provided. To do this, we shall develop a generator of extensive-form games similar to GAMUT.

References


