Subgame-Perfect Equilibria in Mean-Payoff Games

Léonard Brice ✉
LIGM, Univ. Gustave Eiffel, CNRS, F-77454 Marne-la-Vallée, France

Jean-François Raskin ✉
Université libre de Bruxelles, Brussels, Belgium

Marie van den Bogaard ✉
LIGM, Univ. Gustave Eiffel, CNRS, F-77454 Marne-la-Vallée, France

Abstract

In this paper, we provide an effective characterization of all the subgame-perfect equilibria in infinite duration games played on finite graphs with mean-payoff objectives. To this end, we introduce the notion of requirement, and the notion of negotiation function. We establish that the plays that are supported by SPEs are exactly those that are consistent with the least fixed point of the negotiation function. Finally, we show that the negotiation function is piecewise linear, and can be analyzed using the linear algebraic tool box. As a corollary, we prove the decidability of the SPE constrained existence problem, whose status was left open in the literature.

2012 ACM Subject Classification Software and its engineering → Formal methods; Theory of computation → Logic and verification; Theory of computation → Solution concepts in game theory

Keywords and phrases Games on graphs, subgame-perfect equilibria, mean-payoff objectives.

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2021.8

Related Version *Full Version*: `https://arxiv.org/abs/2101.10685` [3]

Funding This work is partially supported by the ARC project Non-Zero Sum Game Graphs: Applications to Reactive Synthesis and Beyond (Fédération Wallonie-Bruxelles), the EOS project Verifying Learning Artificial Intelligence Systems (F.R.S.-FNRS & FWO), the COST Action 16228 GAMENET (European Cooperation in Science and Technology), and by the PDR project *Subgame perfection in graph games* (F.R.S.- FNRS).

1 Introduction

The notion of Nash equilibrium (NE) is one of the most important and most studied solution concepts in game theory. A profile of strategies is an NE when no rational player has an incentive to change their strategy unilaterally, i.e. while the other players keep their strategies. Thus an NE models a stable situation. Unfortunately, it is well known that, in sequential games, NEs suffer from the problem of *non-credible threats*, see e.g. [18]. In those games, some NE only exists when some players do *not* play rationally in subgames and so use non-credible threats to force the NE. This is why, in sequential games, the stronger notion of *subgame-perfect equilibrium* is used instead: a profile of strategies is a subgame-perfect equilibrium (SPE) if it is an NE in all the subgames of the sequential game. Thus SPE imposes rationality even after a deviation has occured.

In this paper, we study sequential games that are infinite-duration games played on graphs with mean-payoff objectives, and focus on SPEs. While NEs are guaranteed to exist in infinite duration games played on graphs with mean-payoff objectives, it is known that it is not the case for SPEs, see e.g. [19, 5]. We provide in this paper a constructive characterization of the entire set of SPEs, which allows us to decide, among others, the SPE (constrained) existence problem. This problem was left open in previous contributions on the subject. More precisely, our contributions are described in the next paragraphs.

Contributions. First, we introduce two important new notions that allow us to capture NEs, and more importantly SPEs, in infinite duration games played on graphs with mean-payoff objectives[1]: the notion of *requirement* and the notion of *negotiation function*.

A requirement λ is a function that assigns to each vertex $v \in V$ of a game graph a value in $\mathbb{R} \cup \{-\infty, +\infty\}$. The value $\lambda(v)$ represents a requirement on any play $\rho = \rho_0\rho_1 \dots \rho_n \dots$ that traverses this vertex: if we want the player who controls the vertex v to follow ρ and to give up deviating from ρ, then the play must offer a payoff to this player that is at least $\lambda(v)$. An infinite play ρ is λ-consistent if, for each player i, the payoff of ρ for player i is larger than or equal to the largest value of λ on vertices occurring along ρ and controlled by player i.

We first use those notions to rephrase a classical result about NEs: if λ maps a vertex v to the largest value that the player that controls v can secure against a fully adversarial coalition of the other players, i.e. if $\lambda(v)$ is the zero-sum worst-case value, then the set of plays that are λ-consistent is exactly the set of plays that are supported by an NE (Theorem 24).

As SPEs are forcing players to play rationally in all subgames, we cannot rely on the zero-sum worst-case value to characterize them. Indeed, when considering the worst-case value, we allow adversaries to play fully adversarially after a deviation and so potentially in an irrational way w.r.t. their own objective. In fact, in an SPE, a player is refrained to deviate when opposed by a coalition of *rational adversaries*. To characterize this relaxation of the notion of worst-case value, we rely on our notion of *negotiation function*.

The negotiation function nego operates from the set of requirements into itself. To understand the purpose of the negotiation function, let us consider its application on the requirement λ that maps every vertex v on the worst-case value as above. Now, we can naturally formulate the following question: given v and λ, can the player who controls v improve the value that they can ensure against all the other players, if only plays that are consistent with λ are proposed by the other players? In other words, can this player enforce a better value when playing against the other players if those players are not willing to give away their own worst-case value? Clearly, securing this worst-case value can be seen as a minimal goal for any *rational* adversary. So $\text{nego}(\lambda)(v)$ returns this value; and this reasoning can be iterated. One of the contributions of this paper is to show that the least fixed point λ^* of the negotiation function is exactly characterizing the set of plays supported by SPEs (Theorem 28).

To turn this fixed point characterization of SPEs into algorithms, we additionally draw links between the negotiation function and two classes of zero-sum games, that are called *abstract* and *concrete* negotiation games (see Theorem 32). We show that the latter can be solved effectively and allow, given λ, to compute $\text{nego}(\lambda)$ (Lemma 36). While solving concrete negotiation games allows us to compute $\text{nego}(\lambda)$ for any requirement λ, and even if the function $\text{nego}(\cdot)$ is monotone and Scott-continuous, a direct application of the Kleene-Tarski fixed point theorem is not sufficient to obtain an effective algorithm to compute λ^*. Indeed, we give examples that require a transfinite number of iterations to converge to the least fixed point. To provide an algorithm to compute λ^*, we show that the function $\text{nego}(\cdot)$ is piecewise linear and we provide an effective representation of this function (Theorem 41). This effective representation can then be used to extract all its fixed points and in particular its least fixed point using linear algebraic techniques, hence the decidability of the SPE (constrained) existence problem (Theorem 45). Finally, all our results are also shown to extend to ε-SPEs, those are quantitative relaxations of SPEs.

[1] A large part of our results apply to the larger class of games with prefix-independent objectives. For the sake of readability of this introduction, we focus here on mean-payoff games but the technical results in the paper are usually covering broader classes of games.

Related works. Non-zero sum infinite duration games have attracted a large attention in recent years, with applications targeting reactive synthesis problems. We refer the interested reader to the following survey papers [2, 7] and their references for the relevant literature. We detail below contributions more closely related to the work presented here.

In [6], Brihaye et al. offer a characterization of NEs in quantitative games for cost-prefix-linear reward functions based on the worst-case value. The mean-payoff is cost-prefix-linear. In their paper, the authors do not consider the stronger notion of SPE, which is the central solution concept studied in our paper. In [8], Bruyère et al. study secure equilibria that are a refinement of NEs. Secure equilibria are not subgame-perfect and are, as classical NEs, subject to non-credible threats in sequential games.

In [20], Ummels proves that there always exists an SPE in games with ω-regular objectives and defines algorithms based on tree automata to decide constrained SPE problems. Strategy logics, see e.g. [12], can be used to encode the concept of SPE in the case of ω-regular objectives with application to the rational synthesis problem [15] for instance. In [13], Flesch et al. show that the existence of ε-SPEs is guaranteed when the reward function is *lower-semicontinuous*. The mean-payoff reward function is neither ω-regular, nor lower-semicontinuous, and so the techniques defined in the papers cited above cannot be used in our setting. Furthermore, as already recalled above, see e.g. [21, 5], contrary to the ω-regular case, SPEs in games with mean-payoff objectives may fail to exist.

In [5], Brihaye et al. introduce and study the notion of weak subgame-perfect equilibria, which is a weakening of the classical notion of SPE. This weakening is equivalent to the original SPE concept on reward functions that are *continuous*. This is the case for example for the quantitative reachability reward function, on which Brihaye et al. solve the problem of the constrained existence of SPEs in [4]. On the contrary, the mean-payoff cost function is not continuous and the techniques used in [5], and generalized in [10], cannot be used to characterize SPEs for the mean-payoff reward function.

In [17], Meunier develops a method based on Prover-Challenger games to solve the problem of the existence of SPEs on games with a finite number of possible outcomes. This method is not applicable to the mean-payoff reward function, as the number of outcomes in this case is uncountably infinite.

In [14], Flesch and Predtetchinski present another characterization of SPEs on games with finitely many possible outcomes, based on a game structure that we will present here under the name of *abstract negotiation game*. Our contributions differ from this paper in two fundamental aspects. First, it lifts the restriction to finitely many possible outcomes. This is crucial as mean-payoff games violate this restriction. Instead, we identify a class of games, that we call *with steady negotiation*, that encompasses mean-payoff games and for which some of the conceptual tools introduced in that paper can be generalized. Second, the procedure developed by Flesch and Predtetchinski is *not* an algorithm in CS acceptation: it needs to solve infinitely many games that are not represented effectively, and furthermore it needs a transfinite number of iterations. On the contrary, our procedure is effective and leads to a complete algorithm in the classical sense: with guarantee of termination in finite time and applied on effective representations of games.

Structure of the paper. In Sect. 2, we introduce the necessary background. Sect. 3 defines the notion of requirement and the negotiation function. Sect. 4 shows that the set of plays that are supported by an SPE are those that are λ^*-consistent, where λ^* is the least fixed point of the negotiation function. Sect. 5 draws a link between the negotiation function and negotiation games. Sect. 6 establishes that the negotiation function is effectively piecewise linear. Finally, Sect. 7 applies those results to prove the decidability of the SPE constrained existence problem on mean-payoff games, and adds some complexity considerations. The detailed proofs of our results, as well as additional examples, can be found in appendices of [3], the full version of this paper.

2 Background

In all what follows, we will use the word *game* for the infinite duration turn-based quantitative games on finite graphs with complete information.

▶ **Definition 1** (Game). A *game* is a tuple $G = (\Pi, V, (V_i)_{i\in\Pi}, E, \mu)$, where:
- Π is a finite set of *players*;
- (V, E) is a finite directed graph, whose vertices are sometimes called *states* and whose edges are sometimes called *transitions*, and in which every state has at least one outgoing transition. For the simplicity of writing, a transition $(v, w) \in E$ will often be written vw.
- $(V_i)_{i\in\Pi}$ is a partition of V, in which V_i is the set of states *controlled* by player i;
- $\mu : V^\omega \to \mathbb{R}^\Pi$ is an *outcome function*, that maps each infinite word ρ to the tuple $\mu(\rho) = (\mu_i(\rho))_{i\in\Pi}$ of the players' *payoffs*.

▶ **Definition 2** (Initialized game). An *initialized game* is a tuple (G, v_0), often written $G_{\restriction v_0}$, where G is a game and $v_0 \in V$ is a state called *initial state*. Moreover, the game $G_{\restriction v_0}$ is *well-initialized* if any state of G is accessible from v_0 in the graph (V, E).

▶ **Definition 3** (Play, history). A *play* (resp. history) in the game G is an infinite (resp. finite) path in the graph (V, E). It is also a play (resp. history) in the initialized game $G_{\restriction v_0}$, where v_0 is its first vertex. The set of plays (resp. histories) in the game G (resp. the initialized game $G_{\restriction v_0}$) is denoted by $\mathrm{Plays}G$ (resp. $\mathrm{Plays}G_{\restriction v_0}, \mathrm{Hist}G, \mathrm{Hist}G_{\restriction v_0}$). We write Hist_iG (resp. $\mathrm{Hist}_iG_{\restriction v_0}$) for the set of histories in G (resp. $G_{\restriction v_0}$) of the form hv, where v is a vertex controlled by player i.

▶ Remark. In the literature, the word *outcome* can be used to name plays, and the word *payoff* to name what we call here outcome. Here, the word *payoff* will be used to refer to outcomes, seen from the point of view of a given player – or in other words, an *outcome* will be seen as the collection of all players' payoffs.

▶ **Definition 4** (Strategy, strategy profile). A *strategy* for player i in the initialized game $G_{\restriction v_0}$ is a function $\sigma_i : \mathrm{Hist}_iG_{\restriction v_0} \to V$, such that $v\sigma_i(hv)$ is an edge of (V, E) for every hv. A history h is *compatible* with a strategy σ_i if and only if $h_{k+1} = \sigma_i(h_0 \dots h_k)$ for all k such that $h_k \in V_i$. A play ρ is compatible with σ_i if all its prefixes are.

A *strategy profile* for $P \subseteq \Pi$ is a tuple $\bar{\sigma}_P = (\sigma_i)_{i\in P}$, where for each i, σ_i is a strategy for player i in $G_{\restriction v_0}$. A *complete* strategy profile, usually written $\bar{\sigma}$, is a strategy profile for Π. A play or a history is *compatible* with $\bar{\sigma}_P$ if it is compatible with every σ_i for $i \in P$.

When i is a player and when the context is clear, we will often write $-i$ for the set $\Pi \setminus \{i\}$. We will often refer to $\Pi \setminus \{i\}$ as the *environment* against player i. When $\bar{\tau}_P$ and $\bar{\tau}'_Q$ are two strategy profiles with $P \cap Q = \emptyset$, $(\bar{\tau}_P, \bar{\tau}'_Q)$ denotes the strategy profile $\bar{\sigma}_{P\cup Q}$ such that $\sigma_i = \tau_i$ for $i \in P$, and $\sigma_i = \tau'_i$ for $i \in Q$.

Before moving on to SPEs, let us recall the notion of Nash equilibrium.

▶ **Definition 5** (Nash equilibrium). Let $G_{\restriction v_0}$ be an initialized game. The strategy profile $\bar{\sigma}$ is a *Nash equilibrium* – or *NE* for short – in $G_{\restriction v_0}$ if and only if for each player i and for every strategy σ'_i, called *deviation of* σ_i, we have the inequality $\mu_i(\langle \sigma'_i, \bar{\sigma}_{-i}\rangle_{v_0}) \leq \mu_i(\langle \bar{\sigma} \rangle_{v_0})$.

To define SPEs, we need the notion of subgame.

▶ **Definition 6** (Subgame, substrategy). Let hv be a history in the game G. The *subgame* of G after hv is the initialized game $(\Pi, V, (V_i)_i, E, \mu_{\upharpoonright hv})_{\upharpoonright v}$, where $\mu_{\upharpoonright hv}$ maps each play to its payoff in G, assuming that the history hv has already been played: formally, for every $\rho \in \text{Plays}G_{\upharpoonright hv}$, we have $\mu_{\upharpoonright hv}(\rho) = \mu(h\rho)$.

If σ_i is a strategy in $G_{\upharpoonright v_0}$, its *substrategy* after hv is the strategy $\sigma_{i\upharpoonright hv}$ in $G_{\upharpoonright hv}$, defined by $\sigma_{i\upharpoonright hv}(h') = \sigma_i(hh')$ for every $h' \in \text{Hist}_i G_{\upharpoonright hv}$.

▶ Remark. The initialized game $G_{\upharpoonright v_0}$ is also the subgame of G after the one-state history v_0.

▶ **Definition 7** (Subgame-perfect equilibrium). Let $G_{\upharpoonright v_0}$ be an initialized game. The strategy profile $\bar{\sigma}$ is a *subgame-perfect equilibrium* – or *SPE* for short – in $G_{\upharpoonright v_0}$ if and only if for every history h in $G_{\upharpoonright v_0}$, the strategy profile $\bar{\sigma}_{\upharpoonright h}$ is a Nash equilibrium in the subgame $G_{\upharpoonright h}$.

The notion of subgame-perfect equilibrium can be seen as a refinement of Nash equilibrium: it is a stronger equilibrium which excludes players resorting to non-credible threats.

▶ Example 8. In the game represented in Figure 1a, where the square state is controlled by player □ and the round states by player ○, if both players get the payoff 1 by reaching the state d and the payoff 0 in the other cases, there are actually two NEs: one, in blue, where □ goes to the state b and then player ○ goes to d, and both win, and one, in red, where player □ goes to the state c because player ○ was planning to go to e. However, only the blue one is an SPE, as moving from b to e is irrational for player ○ in the subgame $G_{\upharpoonright ab}$.

An ε-SPE is a strategy profile which is *almost* an SPE: if a player deviates after some history, they will not be able to improve their payoff by more than a quantity $\varepsilon \geq 0$.

▶ **Definition 9** (ε-SPE). Let $G_{\upharpoonright v_0}$ be an initialized game, and $\varepsilon \geq 0$. A strategy profile $\bar{\sigma}$ from v_0 is an ε-SPE if and only if for every history hv, for every player i and every strategy σ'_i, we have $\mu_i(\langle \bar{\sigma}_{-i\upharpoonright hv}, \sigma'_{i\upharpoonright hv}\rangle_v) \leq \mu_i(\langle \bar{\sigma}_{\upharpoonright hv}\rangle_v) + \varepsilon$.

Note that a 0-SPE is an SPE, and conversely.

Hereafter, we focus on *prefix-independent* games, and in particular *mean-payoff* games.

▶ **Definition 10** (Mean-payoff game). A *mean-payoff game* is a game $G = (\Pi, V, (V_i)_i, E, \mu)$, where μ is defined from a function $\pi : E \to \mathbb{Q}^\Pi$, called *weight function*, by, for each player i:

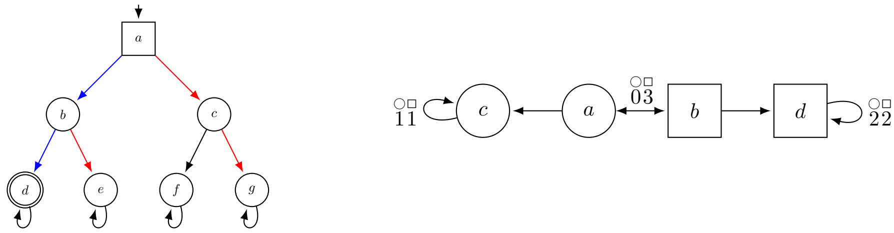
$$\mu_i : \rho \mapsto \liminf_{n \to \infty} \frac{1}{n} \sum_{k=0}^{n-1} \pi_i (\rho_k \rho_{k+1}) .$$

In a mean-payoff game, the weight given by the function π represents the immediate reward that each action gives to each player. The final payoff of each player is their average payoff along the play, classically defined as the limit inferior over n (since the limit may not be defined) of the average payoff after n steps.

▶ **Definition 11** (Prefix-independent game). A game G is *prefix-independent* if, for every history h and for every play ρ, we have $\mu(h\rho) = \mu(\rho)$. We also say, in that case, that the outcome function μ is prefix-independent.

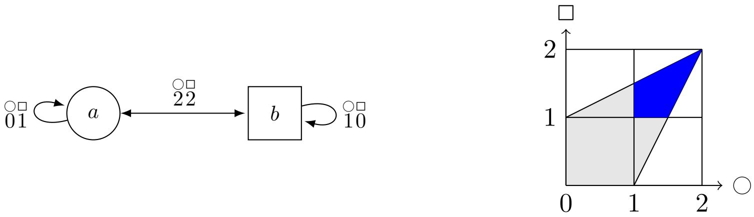
Mean-payoff games are prefix-independent. We now recall a classical result about two-player zero-sum games.

▶ **Definition 12** (Zero-sum game). A game G, with $\Pi = \{1, 2\}$, is *zero-sum* if $\mu_2 = -\mu_1$.



(a) Two NEs and one SPE. **(b)** A game without SPE.

Figure 1 Two examples of games.



(a) The game G. **(b)** The outcomes of plays and SPE plays in G.

Figure 2 A game with an infinity of SPEs.

▶ **Definition 13** (Borel game). A game G is *Borel* if the function μ, from the set V^ω equipped with the product topology to the Euclidian space $\mathbb{R}^\Pi$, is Borel, i.e. if, for every Borel set $B \subseteq \mathbb{R}^\Pi$, the set $\mu^{-1}(B)$ is Borel.

▶ **Proposition 14** (Determinacy of two-player zero-sum Borel games, [16]). *Let $G_{\restriction v_0}$ be an initialized zero-sum Borel game, with $\Pi = \{1, 2\}$. Then, we have the following equality:*

$$\sup_{\sigma_1} \inf_{\sigma_2} \mu_1(\langle \bar{\sigma} \rangle_{v_0}) = \inf_{\sigma_2} \sup_{\sigma_1} \mu_1(\langle \bar{\sigma} \rangle_{v_0}).$$

That quantity is called value *of $G_{\restriction v_0}$, denoted by* $\mathrm{val}_1(G_{\restriction v_0})$*;* solving *the game G means computing its value.*

The following examples illustrate the SPE existence problem in mean-payoff games.

▶ Example 15. Let G be the mean-payoff game of Figure 1b, where each edge is labelled by its weights $\pi_\bigcirc$ and $\pi_\Box$. No weight is given for the edges ac and bd since they can be used only once, and therefore do not influence the final payoff. As shown in [9], this game does not have any SPE, neither from the state a nor from the state b.

Indeed, the only NE plays from the state b are the plays where player $\Box$ eventually leaves the cycle ab and goes to d: if he stays in the cycle ab, then player $\bigcirc$ would be better off leaving it, and if she does, player $\Box$ would be better off leaving it before. From the state a, if player $\bigcirc$ knows that player $\Box$ will leave, she has no incentive to do it before: there is no NE where $\bigcirc$ leaves the cycle and $\Box$ plans to do it if ever she does not. Therefore, there is no SPE where $\bigcirc$ leaves the cycle. But then, after a history that terminates in b, player $\Box$ has actually no incentive to leave if player $\bigcirc$ never plans to do it afterwards: contradiction.

▶ Example 16. Let us now study the game of Figure 2a. Using techniques from [11], we can represent the outcomes of possible plays in that game as in Figure 2b (gray and blue areas).

Following exclusively one of the three simple cycles a, ab and b of the game graph during a play yields the outcomes $01, 10$ and 22, respectively. By combining those cycles with well chosen frequencies, one can obtain any outcome in the convex hull of those three points. Now, it is also possible to obtain the point 00 by using the properties of the limit inferior: it is for instance the outcome of the play $a^2 b^4 a^{16} b^{256} \dots a^{2^{2^n}} b^{2^{2^{n+1}}} \dots$. In fact, one can construct a play that yields any outcome in the convex hull of the four points $00, 10, 01$, and 22.

We claim that the outcomes of SPEs plays correspond to the entire blue area in Figure 2b: there exists an SPE $\bar{\sigma}$ in $G_{\restriction a}$ with $\langle \bar{\sigma} \rangle_a = \rho$ if and only if $\mu_{\Box}(\rho), \mu_{\bigcirc}(\rho) \geq 1$. That statement will be a direct consequence of the results we show in the remaining sections, but let us give a first intuition: a play with such an outcome necessarily uses infinitely often both states. It is an NE play because none of the players can get a better payoff by looping forever on their state, and they can both force each other to follow that play, by threatening them to loop for ever on their state whenever they can. But such a strategy profile is clearly not an SPE.

It can be transformed into an SPE as follows: when a player deviates, say player $\Box$, then player $\bigcirc$ can punish him by looping on a, not forever, but a great number of times, until player $\Box$'s mean-payoff gets very close to 1. Afterwards, both players follow again the play that was initially planned. Since that threat is temporary, it does not affect player $\bigcirc$'s payoff on the long term, but it really punishes player $\Box$ if that one tries to deviate infinitely often.

3 Requirements and negotiation

We will now see that SPEs are strategy profiles that respect some *requirements* about the payoffs, depending on the states it traverses. In this part, we develop the notions of *requirement* and *negotiation*.

3.1 Requirement

In the method we will develop further, we will need to analyze the players' behaviour when they have some *requirement* to satisfy. Intuitively, one can see requirements as *rationality constraints* for the players, that is, a threshold payoff value under which a player will not accept to follow a play. In all what follows, $\overline{\mathbb{R}}$ denotes the set $\mathbb{R} \cup \{\pm\infty\}$.

▶ **Definition 17** (Requirement). A *requirement* on the game G is a function $\lambda : V \to \overline{\mathbb{R}}$.

For a given state v, the quantity $\lambda(v)$ represents the minimal payoff that the player controlling v will require in a play beginning in v.

▶ **Definition 18** (λ-consistency). Let λ be a requirement on a game G. A play ρ in G is λ-*consistent* if and only if, for all $i \in \Pi$ and $n \in \mathbb{N}$ with $\rho_n \in V_i$, we have $\mu_i(\rho_n \rho_{n+1} \dots) \geq \lambda(\rho_n)$. The set of the λ-consistent plays from a state v is denoted by $\lambda\mathrm{Cons}(v)$.

▶ **Definition 19** (λ-rationality). Let λ be a requirement on a mean-payoff game G. Let $i \in \Pi$. A strategy profile $\bar{\sigma}_{-i}$ is λ-*rational* if and only if there exists a strategy σ_i such that, for every history hv compatible with $\bar{\sigma}_{-i}$, the play $\langle \bar{\sigma}_{\restriction hv} \rangle_v$ is λ-consistent. We then say that the strategy profile $\bar{\sigma}_{-i}$ is λ-rational *assuming* σ_i. The set of λ-rational strategy profiles in $G_{\restriction v}$ is denoted by $\lambda\mathrm{Rat}(v)$.

Note that λ-rationality is a property of a strategy profile for all the players but one, player i. Intuitively, their rationality is justified by the fact that they collectively assume that player i will, eventually, play according to the strategy σ_i: if player i does so, then everyone gets their payoff satisfied. Finally, let us define a particular requirement: the *vacuous requirement*, that requires nothing, and with which every play is consistent.

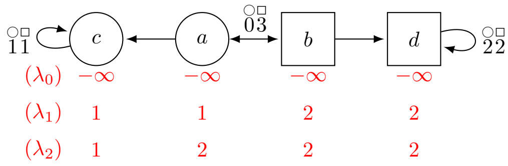


Figure 3 A game without SPE.

▶ **Definition 20** (Vacuous requirement)**.** In any game, the *vacuous requirement*, denoted by λ_0, is the requirement constantly equal to $-\infty$.

3.2 Negotiation

We will show that SPEs in prefix-independent games are characterized by the fixed points of a function on requirements. That function can be seen as a *negotiation*: when a player has a requirement to satisfy, another player can hope a better payoff than what they can secure in general, and therefore update their own requirement.

▶ **Definition 21** (Negotiation function)**.** Let G be a game. The *negotiation function* is the function that transforms any requirement λ on G into a requirement $\text{nego}(\lambda)$ on G, such that for each $i \in \Pi$ and $v \in V_i$, with the convention $\inf \emptyset = +\infty$, we have:

$$\text{nego}(\lambda)(v) = \inf_{\bar{\sigma}_{-i} \in \lambda\text{Rat}(v)} \sup_{\sigma_i} \mu_i(\langle \bar{\sigma} \rangle_v).$$

▶ Remarks. There exists a λ-rational strategy profile from v against the player controlling v if and only if $\text{nego}(\lambda)(v) \neq +\infty$. The negotiation function is monotone: if $\lambda \leq \lambda'$ (for the pointwise order, i.e. if for each v, $\lambda(v) \leq \lambda'(v)$), then $\text{nego}(\lambda) \leq \text{nego}(\lambda')$. The negotiation function is also non-decreasing: for every λ, we have $\lambda \leq \text{nego}(\lambda)$.

In the general case, the quantity $\text{nego}(\lambda)(v)$ represents the worst case value that the player controlling v can ensure, assuming that the other players play λ-rationally.

▶ Example 22. Let us consider the game of Example 15: in Figure 3, on the two first lines below the states, we present the requirements λ_0 and $\lambda_1 = \text{nego}(\lambda_0)$, which is easy to compute since any strategy profile is λ_0-rational: for each v, $\lambda_1(v)$ is the classical *worst-case value* or *antagonistic value* of v, i.e. the best value the player controlling v can enforce against a fully hostile environment. Let us now compute the requirement $\lambda_2 = \text{nego}(\lambda_1)$.

From c, there exists exactly one λ_1-rational strategy profile $\bar{\sigma}_{-\bigcirc} = \sigma_\square$, which is the empty strategy since player $\square$ has never to choose anything. Against that strategy, the best and the only payoff player $\bigcirc$ can get is 1, hence $\lambda_2(c) = 1$. For the same reasons, $\lambda_2(d) = 2$.

From b, player $\bigcirc$ can force $\square$ to get the payoff 2 or less, with the strategy profile $\sigma_\bigcirc : h \mapsto c$. Such a strategy is λ_1-rational, assuming the strategy $\sigma_\square : h \mapsto d$. Therefore, $\lambda_2(b) = 2$.

Finally, from a, player $\square$ can force $\bigcirc$ to get the payoff 2 or less, with the strategy profile $\sigma_\square : h \mapsto d$. Such a strategy is λ_1-rational, assuming the strategy $\sigma_\bigcirc : h \mapsto c$. But, he cannot force her to get less than the payoff 2, because she can force the access to the state b, and the only λ_1-consistent plays from b are the plays with the form $(ba)^k bd^\omega$. Therefore, $\lambda_2(a) = 2$.

3.3 Steady negotiation

In what follows, we will often need a game to be *with steady negotiation*, i.e. such that there always exists a worst λ-rational behaviour for the environment against a given player.

▶ **Definition 23** (Game with steady negotiation). A game G is *with steady negotiation* if and only if for every player i, for every vertex v, and for every requirement λ, the set $\left\{\sup_{\sigma_i} \ \mu_i(\langle \bar{\sigma}_{-i}, \sigma_i \rangle_v) \ \middle| \ \bar{\sigma}_{-i} \in \lambda\mathrm{Rat}(v)\right\}$ is either empty, or has a minimum.

▶ Remark. In particular, when a game is with steady negotiation, the infimum in the definition of negotiation is always reached.

It will be proved in Section 5 that mean-payoff games are with steady negotiation.

3.4 Link with Nash equilibria

Requirements and the negotiation function are able to capture Nash equilibria. Indeed, if λ_0 is the vacuous requirement, then $\mathrm{nego}(\lambda_0)$ characterizes the plays that are supported by a Nash equilibrium (abbreviated by NE plays), in the following formal sense:

▶ **Theorem 24.** *Let G be a game with steady negotiation. Then, a play ρ in G is an NE play if and only if ρ is $\mathrm{nego}(\lambda_0)$-consistent.*

▶ Example 25. Let us consider again the game of Example 15, with the requirement λ_1 given in Figure 3. The only λ_1-consistent plays in this game, starting from the state a, are ac^ω, and $(ab)^k d^\omega$ with $k \geq 1$. One can check that those plays are exactly the NE plays in that game.

In the following section, we will prove that as well as $\mathrm{nego}(\lambda_0)$ characterizes the NEs, the requirement that is the least fixed point of the negotiation function characterizes the SPEs.

4 Link between negotiation and SPEs

The notion of negotiation will enable us to find the SPEs, but also more generally the ε-SPEs, in a game. For that purpose, we need the notion of ε-fixed points of a function.

▶ **Definition 26** (ε-fixed point). Let $\varepsilon \geq 0$, let D be a finite set and let $f : \overline{\mathbb{R}}^D \to \overline{\mathbb{R}}^D$ be a mapping. A tuple $\bar{x} \in \mathbb{R}^D$ is a *ε-fixed point* of f if for each $d \in D$, for $\bar{y} = f(\bar{x})$, we have $y_d \in [x_d - \varepsilon, x_d + \varepsilon]$.

▶ Remark. A 0-fixed point is a fixed point, and conversely.

The set of requirements, equipped with the componentwise order, is a complete lattice. Since the negotiation function is monotone, Tarski's fixed point theorem states that the negotiation function has a least fixed point. That result can be generalized to ε-fixed points:

▶ **Lemma 27.** *Let $\varepsilon \geq 0$. On each game, the function* nego *has a least ε-fixed point.*

Intuitively, the ε-fixed points of the negotiation function are the requirements λ such that, from every vertex v, the player i controlling v cannot enforce a payoff greater than $\lambda(v) + \varepsilon$ against a λ-rational behaviour. Therefore, the λ-consistent plays are such that if one player tries to deviate, it is possible for the other players to prevent them improving their payoff by more than ε, while still playing rationally. Formally:

▶ **Theorem 28.** *Let $G_{\restriction v_0}$ be an initialized prefix-independent game, and let $\varepsilon \geq 0$. Let λ^* be the least ε-fixed point of the negotiation function. Let ξ be a play starting in v_0. If there exists an ε-SPE $\bar{\sigma}$ such that $\langle \bar{\sigma} \rangle_{v_0} = \xi$, then ξ is λ^*-consistent. The converse is true if the game G is with steady negotiation.*

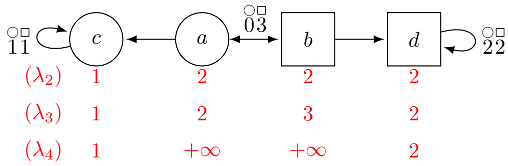


Figure 4 A game without SPE.

5 Negotiation games

We have now proved that SPEs are characterized by the requirements that are fixed points of the negotiation function; but we need to know how to compute, in practice, the quantity $\text{nego}(\lambda)$ for a given requirement λ. In other words, we need a algorithm that computes, given a state v_0 controlled by a player i in the game G, and given a requirement λ, which value player i can ensure in $G_{\restriction v_0}$ if the other players play λ-rationally.

5.1 Abstract negotiation game

We first define an *abstract negotiation game*, that is conceptually simple but not directly usable for an algorithmic purpose, because it is defined on an uncoutably infinite state space.

A similar definition was given in [14], as a tool in a general method to compute SPE plays in games whose payoff functions have finite range, which is not the case of mean-payoff games. Here, linking that game with our concepts of requirements, negotiation function and steady negotiation enables us to present an effective algorithm in the case of mean-payoff games, by constructing a finite version of the abstract negotiation game, the *concrete negotiation game*, and afterwards by analyzing the negotiation function with linear algebra tools.

The abstract negotiation game from a state v_0, with regards to a player i and a requirement λ, is denoted by $\text{Abs}_{\lambda i}(G)_{\restriction [v_0]}$ and opposes two players, *Prover* and *Challenger*, as follows:

- Prover proposes a λ-consistent play ρ from v_0 (or loses, if she has no play to propose).
- Then, either Challenger accepts the play and the game terminates; or, he chooses an edge $\rho_k\rho_{k+1}$, with $\rho_k \in V_i$, from which he can make player i deviate, using another edge $\rho_k v$ with $v \neq \rho_{k+1}$: then, the game starts again from v instead of v_0.
- In the resulting play (either eventually accepted by Challenger, or constructed by an infinity of deviations), Prover wants player i's payoff to be low, and Challenger wants it to be high.

That game gives us the basis of a method to compute $\text{nego}(\lambda)$ from λ: the maximal outcome that Challenger – or $\mathbb{C}$ for short – can ensure in $\text{Abs}_{\lambda i}(G)_{\restriction [v_0]}$, with $v_0 \in V_i$, is also the maximal payoff that player i can ensure in $G_{\restriction v_0}$, against a λ-rational environment; hence the equality $\text{val}_{\mathbb{C}}\left(\text{Abs}_{\lambda i}(G)_{\restriction [v_0]}\right) = \text{nego}(\lambda)(v_0)$. A proof of that statement, with a complete formalization of the abstract negotiation game, is presented in [3].

▶ Example 29. Let us consider again the game of Example 15: the requirement $\lambda_2 = \text{nego}(\lambda_1)$, computed in Section 3.2, is given again in Figure 4. Let us use the abstract negotiation game to compute the requirement $\lambda_3 = \text{nego}(\lambda_2)$.

From a, Prover can propose the play abd^ω, and the only deviation Challenger can do is going to c; he has of course no incentive to do it. Therefore, $\lambda_3(a) = 2$. From b, whatever Prover proposes at first, Challenger can deviate and go to a. Then, from a, Prover cannot propose the play ac^ω, which is not λ_2-consistent: she has to propose a play beginning by ab, and to let Challenger deviate once more. He can then deviate infinitely often that way, and generate the play $(ba)^\omega$: therefore, $\lambda_3(b) = 3$. The other states keep the same values. Note that there exists no λ_3-consistent play from a or b, hence $\text{nego}(\lambda_3)(a) = \text{nego}(\lambda_3)(b) = +\infty$. This proves that there is no SPE in that game.

5.2 Concrete negotiation game

In the abstract negotiation game, Prover has to propose complete plays, on which we can make the hypothesis that they are λ-consistent. In practice, there will often be an infinity of such plays, and therefore it cannot be used directly for an algorithmic purpose. Instead, those plays can be given edge by edge, in a finite state game. Its definition is more technical, but it can be shown that it is equivalent to the abstract one. In order to make the definition as clear as possible, we give it only when the original game is a mean-payoff game. However, one could easily adapt this definition to other classes of prefix-independent games.

▶ **Definition 30** (Concrete negotiation game). Let $G_{\restriction v_0}$ be an initialized mean-payoff game, and let λ be a requirement on G, with either $\lambda(V) \subseteq \mathbb{R}$, or $\lambda = \lambda_0$.

The *concrete negotiation game* of $G_{\restriction v_0}$ for player i is the two-player zero-sum game $\text{Conc}_{\lambda i}(G)_{\restriction s_0} = (\{\mathbb{P}, \mathbb{C}\}, S, (S_\mathbb{P}, S_\mathbb{C}), \Delta, \nu)_{\restriction s_0}$, defined as follows:

- The set of states controlled by Prover is $S_\mathbb{P} = V \times 2^V$, where the state $s = (v, M)$ contains the information of the current state v on which Prover has to define the strategy profile, and the *memory* M of the states that have been traversed so far since the last deviation, and that define the requirements Prover has to satisfy. The initial state is $s_0 = (v_0, \{v_0\})$.
- The set of states controlled by Challenger is $S_\mathbb{C} = E \times 2^V$, where in the state $s = (uv, M)$, the edge uv is the edge proposed by Prover.
- The set Δ contains three types of transitions: *proposals*, *acceptations* and *deviations*.
  - The proposals are transitions in which Prover proposes an edge of the game G:

$$\text{Prop} = \left\{(v, M)(vw, M) \;\middle|\; vw \in E, M \in 2^V \right\};$$

  - the acceptations are transitions in which Challenger accepts to follow the edge proposed by Prover (it is in particular his only possibility when that edge begins on a state that is not controlled by player i) – note that the memory is updated:

$$\text{Acc} = \{(vw, M)\,(w, M \cup \{w\}) \;|\; j \in \Pi, w \in V_j\};$$

  - the deviations are transitions in which Challenger refuses to follow the edge proposed by Prover, as he can if that edge begins in a state controlled by player i – the memory is erased, and only the new state the deviating edge leads to is memorized:

$$\text{Dev} = \{(uv, M)(w, \{w\}) \;|\; u \in V_i, w \neq v, uw \in E\}.$$

- On those transitions, we define a multidimensional weight function $\hat{\pi} : \Delta \to \mathbb{R}^{\Pi \cup \{\star\}}$, with one dimension per player (*non-main* dimensions) plus one special dimension (*main* dimension) denoted by the symbol $\star$. For each non-main dimension $j \in \Pi$, we define:
  - on proposals: $\hat{\pi}_j\left((v, M)(vw, M)\right) = 0$;
  - on acceptations and deviations: $\hat{\pi}_j\left((uv, M)(w, N)\right) = 2\left(\pi_j(uw) - \max\limits_{v_j \in M \cap V_j} \lambda(v_j)\right)$;

  and on the main dimension:
  - on proposals: $\hat{\pi}_\star\left((v, M), (vw, M)\right) = 0$;
  - on acceptations and deviations: $\hat{\pi}_\star\left((uv, M), (w, N)\right) = 2\pi_i(uw)$.

For each dimension d, we write $\hat{\mu}_d$ the corresponding mean-payoff function:

$$\hat{\mu}_d(\rho) = \liminf_{n \in \mathbb{N}} \frac{1}{n} \sum_{k=0}^{n-1} \hat{\pi}_d(\rho_k \rho_{k+1}).$$

Thus, the mean-payoff along the main dimension corresponds to player i's payoff, while the mean-payoff along a non-main dimension j corresponds to player j's payoff... minus the maximal requirement player j has to satisfy.

- Then, the outcome function $\nu_{\mathbb{C}} = -\nu_{\mathbb{P}}$ measures player i's payoff, with a winning condition if the constructed strategy profile is not λ-rational, that is to say if after finitely many player i's deviations, it can generate a play which is not λ-consistent:
  - $\nu_{\mathbb{C}}(\eta) = +\infty$ if after some index $n \in \mathbb{N}$, the play $\eta_n \eta_{n+1} \dots$ contains no deviation, and if $\hat{\mu}_j(\eta) < 0$ for some $j \in \Pi$;
  - $\nu_{\mathbb{C}}(\eta) = \hat{\mu}_\star(\eta)$ otherwise.

Like in the abstract negotiation game, the goal of Challenger is to find a λ-rational strategy profile that forces the worst possible payoff for player i, and the goal of Prover is to find a possibly deviating strategy for player i that gives them the highest possible payoff.

A play or a history in the concrete negotiation game has a projection in the game on which that negotiation game has been constructed, defined as follows:

▶ **Definition 31** (Projection of a history, of a play). Let G be a prefix-independent game. Let λ be a requirement and i a player, and let $\mathrm{Conc}_{\lambda i}(G)$ be the corresponding concrete negotiation game. Let $H = (h_0, M_0)(h_0 h_0', M_0) \dots (h_n h_n', M_n)$ be a history in $\mathrm{Conc}_{\lambda i}(G)$: the *projection* of the history H is the history $\dot{H} = h_0 \dots h_n$ in the game G. That definition is naturally extended to plays.

▶ Remark. For a play η without deviations, we have $\hat{\mu}_j(\eta) \geq 0$ for each $j \in \Pi$ if and only if $\dot{\eta}$ is λ-consistent.

The concrete negotiation game is equivalent to the abstract one: the only differences are that the plays proposed by Prover are proposed edge by edge, and that their λ-consistency is not written in the rules of the game but in its outcome function.

▶ **Theorem 32.** *Let $G_{\restriction v_0}$ be an initialized mean-payoff game. Let λ be a requirement and i a player. Then, we have:*

$$\mathrm{val}_{\mathbb{C}}\left(\mathrm{Conc}_{\lambda i}(G)_{\restriction s_0}\right) = \inf_{\bar{\sigma}_{-i} \in \lambda\mathrm{Rat}(v_0)} \sup_{\sigma_i} \mu_i(\langle \bar{\sigma} \rangle_{v_0}).$$

▶ Example 33. Let us consider again the game from Example 15. Figure 5 represents the game $\mathrm{Conc}_{\lambda_1 \bigcirc}(G)$ (with $\lambda_1(a) = 1$ and $\lambda_1(b) = 2$), where the dashed states are controlled by Challenger, and the other ones by Prover. The dotted arrows indicate the deviations, and the transitions that are not labelled have either the weight 0 on the three dimensions, or meaningless weights since they cannot be used more than once. The red arrows indicate a (memoryless) optimal strategy for Challenger. Against that strategy, the lowest outcome Prover can ensure is 2. Therefore, $\mathrm{nego}(\lambda_1)(v_0) = 2$, in line with the abstract game in Example 29.

5.3 Solving the concrete negotiation game

We now know that $\mathrm{nego}(\lambda)(v)$, for a given requirement λ, a given player i and a given state $v \in V_i$, is the value of the concrete negotiation game $\mathrm{Conc}_{\lambda i}(G)_{\restriction (v, \{v\})}$. Let us now show how, in the mean-payoff case, that value can be computed.

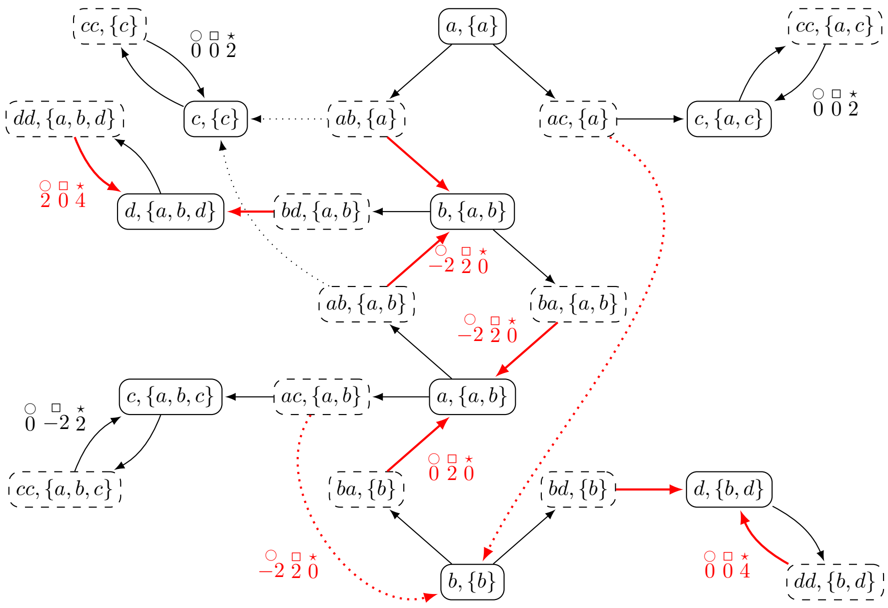


Figure 5 A concrete negotiation game.

▶ **Definition 34** (Memoryless strategy). A strategy σ_i in a game G is *memoryless* if for all vertices $v \in V_i$ and for all histories h and h', we have $\sigma_i(hv) = \sigma_i(h'v)$.

For any game G and any memoryless strategy σ_i, $G[\sigma_i]$ denotes the graph *induced* by σ_i, that is the graph (V, E'), with $E' = \{vw \in E \mid v \notin V_i \text{ or } w = \sigma_i(v)\}$. For any finite set D and any set $X \subseteq \mathbb{R}^D$, $\mathrm{Conv}X$ denotes the convex hull of X.

We can now prove that in the concrete negotiation game constructed from a mean-payoff game, Challenger has an optimal strategy that is memoryless.

▶ **Lemma 35.** *Let $G_{\upharpoonright v_0}$ be an initialized mean-payoff game, let i be a player, let λ be a requirement and let $\mathrm{Conc}_{\lambda i}(G)_{\upharpoonright s_0}$ be the corresponding concrete negotiation game. There exists a memoryless strategy $\tau_{\mathbb{C}}$ that is optimal for Challenger, i.e. such that:*

$$\inf_{\tau_{\mathbb{P}}} \nu_{\mathbb{C}}(\langle \bar{\tau} \rangle_{s_0}) = \mathrm{val}_{\mathbb{C}}\left(\mathrm{Conc}_{\lambda i}(G)_{\upharpoonright s_0}\right).$$

For every game $G_{\upharpoonright v_0}$ and each player i, $\mathrm{ML}_i\left(G_{\upharpoonright v_0}\right)$, or $\mathrm{ML}\left(G_{\upharpoonright v_0}\right)$ when the context is clear, denotes the set of memoryless strategies for player i in $G_{\upharpoonright v_0}$. When (V, E) is a graph, $\mathrm{SC}(V, E)$ denotes the set of its simple cycles, and $\mathrm{SConn}(V, E)$ the set of its strongly connected components. For any closed set $C \subseteq \mathbb{R}^{\Pi \cup \{\star\}}$, the quantity $\min^\star C = \min\{x_\star \mid \bar{x} \in C, \forall j \in \Pi, x_j \geq 0\}$ is the *$\star$-minimum* of C: it will capture, in the concrete negotiation game, the least payoff that can be imposed on player i while keeping every player's payoff above their requirements, among a set of possible outcomes.

With Lemma 35, we can now solve the concrete negotiation game.

▶ **Lemma 36.** *Let $G_{\upharpoonright v_0}$ be an initialized mean-payoff game, and let $\mathrm{Conc}_{\lambda i}(G)_{\upharpoonright s_0}$ be its concrete negotiation game for some λ and some i. Then, the value of the game $\mathrm{Conc}_{\lambda i}(G)_{\upharpoonright s_0}$ is given by the formula:*

$$\max_{\tau_{\mathbb{C}} \in \mathrm{ML}_{\mathbb{C}}(\mathrm{Conc}_{\lambda i}(G))} \; \min_{\substack{K \,\in\, \mathrm{SConn}\,(\mathrm{Conc}_{\lambda i}(G)[\tau_{\mathbb{C}}]) \\ \text{accessible from } s_0}} \mathrm{opt}(K),$$

where $\mathrm{opt}(K)$ is the minimal value $\nu_{\mathbb{C}}(\rho)$ for ρ among the infinite paths in K.

If K contains a deviation, then Prover can choose among its simple cycles the one that minimizes player i's payoff:

$$\mathrm{opt}(K) = \min_{c \in \mathrm{SC}(K)} \hat{\mu}_\star(c^\omega).$$

If K does not contain a deviation, then Prover must choose a combination of its simple cycles that minimizes the main dimension while keeping the other dimensions above 0:

$$\mathrm{opt}(K) = \min{}^\star \operatorname*{Conv}_{c \in \mathrm{SC}(K)} \hat{\mu}(c^\omega).$$

▶ **Corollary 37.** *For each player i and every state $v \in V_i$, the value $\mathrm{nego}(\lambda)(v)$ can be computed with the formula given in Lemma 36 applied to the game $\mathrm{Conc}_{\lambda i}(G)_{\restriction(v,\{v\})}$*

Another corollary of that result is that there always exists a best play that Prover can choose, i.e. Prover has an optimal strategy; by Theorem 32, this is equivalent to saying that:

▶ **Corollary 38.** *Mean-payoff games are games with steady negotiation.*

6 Analysis of the negotiation function in mean-payoff games

When one wants to compute the least fixed point of a function, the usual method is to iterate it on the minimal element of the considered set, to go until that fixed point. That approach is valid if the negotiation function is *Scott-continuous*, i.e. such that for every non-decreasing sequence $(\lambda_n)_n$ of requirements on G, we have $\mathrm{nego}\left(\sup_n \lambda_n\right) = \sup_n \mathrm{nego}(\lambda_n)$.

▶ **Proposition 39.** *In mean-payoff games, the negotiation function is Scott-continuous.*

By Kleene-Tarski fixed-point theorem, the least fixed point of the negotiation function is, then, the limit of the *negotiation sequence*, defined as the sequence $(\lambda_n)_{n\in\mathbb{N}} = (\mathrm{nego}^n(\lambda_0))_n$.

In many cases, the negotiation sequence is stationary, and in such a case, it is possible to compute its limit: whenever a term is equal to the previous one, we know that we reached it. But actually, the negotiation sequence is not always stationary.

▶ Example 40. Let us consider the game of Figure 6. Since all player ◇'s weights are equal to 0, for all $n > 0$, we have $\lambda_n(d) = \lambda_n(f) = 0$. It comes that for all $n > 0$, we also have $\lambda_n(c) = \lambda_n(e) = 0$. Moreover, by symmetry of the game, we always have $\lambda_n(a) = \lambda_n(b)$. Therefore, to compute the negotiation sequence, it suffices to compute $\lambda_{n+1}(a)$ as a function of $\lambda_n(b)$, knowing that $\lambda_1(a) = \lambda_1(b) = 1$, and therefore that for all $n > 0$, $\lambda_n(a) = \lambda_n(b) \geq 1$.

From a, the worst play that player □ could propose to player ○ would be a combination of the cycles cd and d giving her exactly 1. But then, player ○ will deviate to go to b, from which if player □ proposes plays in the strongly connected component containing c and d, then player ○ will always deviate and generate the play $(ab)^\omega$, and then get the payoff 2.

Then, in order to give her a payoff lower than 2, player □ has to go to the state e. Since player ○ does not control any state in that strongly connected component, the play he will propose will be accepted: he will, then, propose the worst possible combination of the cycles ef and f for player ○, such that he gets at least his requirement $\lambda_n(b)$. The payoff $\lambda_{n+1}(a)$ is then the minimal solution of the system:

$$\begin{cases} \lambda_{n+1}(a) = x + 2(1-x) \\ 2(1-x) \geq \lambda_n(b) \\ 0 \leq x \leq 1 \end{cases}$$

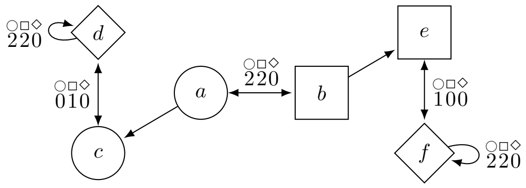
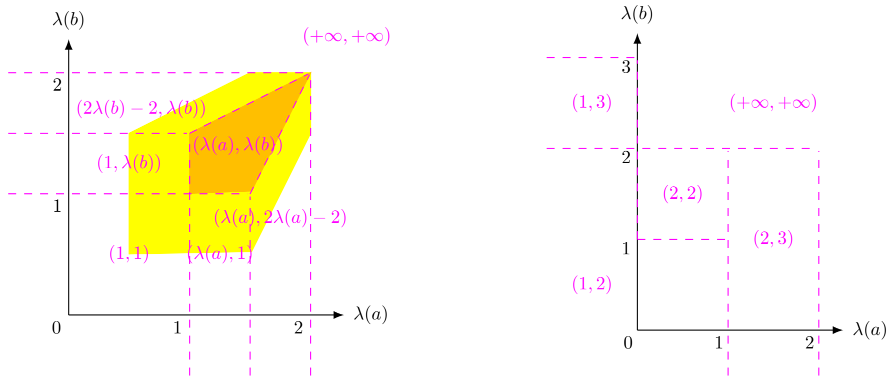


Figure 6 A game where the negotiation sequence is not stationary.



(a) Example 16. **(b)** Example 15.

Figure 7 The negotiation function on the games of Examples 16 and 15.

that is to say $\lambda_{n+1}(a) = 1 + \frac{\lambda_n(b)}{2} = 1 + \frac{\lambda_n(a)}{2}$, and by induction, for all $n > 0$:

$$\lambda_n(a) = \lambda_n(b) = 2 - \frac{1}{2^{n-1}},$$

which converges to 2 but never reaches it.

Therefore, we need a different approach to compute that least fixed point. We will now show that, in the case of mean-payoff games, the negotiation function is a piecewise linear function from the vector space of requirements into itself, which can therefore be computed and analyzed using classical linear algebra techniques. Then, it becomes possible to search for the fixed points or the ε-fixed points of such a function, and to decide the existence or not of SPEs or ε-SPEs in the game studied.

▶ **Theorem 41.** *Let G be a mean-payoff game. Let us assimilate any requirement λ on G with finite values to the tuple $\bar{\lambda} = (\lambda(v))_{v \in V}$, element of the vector space $\mathbb{R}^V$. Then, for each player i and every vertex $v_0 \in V_i$, the quantity $\mathrm{nego}(\lambda)(v_0)$ is a piecewise linear function of $\bar{\lambda}$, and an effective expression of that function can be computed in 2-ExpTime.*

▶ Example 42. Let us consider the game of Example 16. If a requirement λ is represented by the tuple $(\lambda(a), \lambda(b))$, the function $\mathrm{nego} : \mathbb{R}^2 \to \mathbb{R}^2$ can be represented by Figure 7a, where in any one of the regions delimited by the dashed lines, we wrote a formula for the couple $(\mathrm{nego}(\lambda)(a), \mathrm{nego}(\lambda)(b))$. The orange area indicates the fixed points of the function, and the yellow area the other $\frac{1}{2}$-fixed points.

▶ Example 43. Now, let us consider the game of Example 15. If we fix $\lambda(c) = 1$ and $\lambda(d) = 2$, and represent the requirements λ by the tuples $(\lambda(a), \lambda(b))$, as in the previous example. Then, the negotiation function can be represented as in Figure 7b. One can check that there is no fixed point here, and even no $\frac{1}{2}$-fixed point – except $(+\infty, +\infty)$.

7 Conclusion: algorithm and complexity

Thanks to all the previous results, we are now able to compute the least fixed point, or the least ε-fixed point, of the negotiation function, on every mean-payoff game, and to use it as a characterization of all the SPEs or all the ε-SPEs. A direct application is an algorithm that solves the *ε-SPE constrained existence problem*, i.e. that decides, given an initialized mean-payoff game $G_{\restriction v_0}$, two thresholds $\bar{x}, \bar{y} \in \mathbb{Q}^\Pi$, and a rational number $\varepsilon \geq 0$, whether there exists an SPE $\bar{\sigma}$ such that $\bar{x} \leq \mu(\langle \bar{\sigma} \rangle_{v_0}) \leq \bar{y}$.

We leave for future work the optimal complexity of that problem. However, we can easily prove that it cannot be solved in polynomial time, unless $\mathbf{P} = \mathbf{NP}$.

▶ **Theorem 44.** *The ε-SPE constrained existence problem is NP-hard.*

Given $G_{\restriction v_0}$, by Theorem 41, computing a general expression of the negotiation function as a piecewise linear function can be done in time double exponential in the size of G. Then, for each linear piece of nego, computing its set of ε-fixed points is a polynomial problem. Since the number of pieces is at most double exponential in the size of G, computing its entire set of fixed points, and thus its least ε-fixed point λ, can be done in double exponential time.

Then, from the requirement λ and the thresholds $\bar{x}$ and $\bar{y}$, we can construct a multi-mean-payoff automaton $\mathcal{A}_\lambda$ of exponential size, that accepts an infinite word $\rho \in V^\omega$, if and only if ρ is a λ-consistent play of $G_{\restriction v_0}$, and $\bar{x} \leq \mu(\rho) \leq \bar{y}$ – see [3] for the construction of $\mathcal{A}_\lambda$.

Finally, by Theorem 28, there exists an SPE $\bar{\sigma}$ in $G_{\restriction v_0}$ with $\bar{x} \leq \mu(\langle \bar{\sigma} \rangle_{v_0}) \leq \bar{y}$ if and only if the language of the automaton $\mathcal{A}_\lambda$ is nonempty, which can be known in a time polynomial in the size of $\mathcal{A}_\lambda$ (see for example [1]), i.e. in a time exponential in the size of G. We can therefore conclude on the following result:

▶ **Theorem 45.** *The ε-SPE constrained existence problem is decidable and 2-ExpTime-easy.*

References

1 Rajeev Alur, Aldric Degorre, Oded Maler, and Gera Weiss. On omega-languages defined by mean-payoff conditions. In Luca de Alfaro, editor, *Foundations of Software Science and Computational Structures, 12th International Conference, FOSSACS 2009, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009, York, UK, March 22-29, 2009. Proceedings*, volume 5504 of *Lecture Notes in Computer Science*, pages 333–347. Springer, 2009. doi:10.1007/978-3-642-00596-1_24.

2 Romain Brenguier, Lorenzo Clemente, Paul Hunter, Guillermo A. Pérez, Mickael Randour, Jean-François Raskin, Ocan Sankur, and Mathieu Sassolas. Non-zero sum games for reactive synthesis. In *Language and Automata Theory and Applications - 10th International Conference, LATA 2016, Prague, Czech Republic, March 14-18, 2016, Proceedings*, volume 9618 of *Lecture Notes in Computer Science*, pages 3–23. Springer, 2016.

3 Léonard Brice, Jean-François Raskin, and Marie van den Bogaard. Subgame-perfect equilibria in mean-payoff games. *CoRR*, abs/2101.10685, 2021. arXiv:2101.10685.

4 Thomas Brihaye, Véronique Bruyère, Aline Goeminne, Jean-François Raskin, and Marie van den Bogaard. The complexity of subgame perfect equilibria in quantitative reachability games. In *CONCUR*, volume 140 of *LIPIcs*, pages 13:1–13:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019.

5 Thomas Brihaye, Véronique Bruyère, Noémie Meunier, and Jean-François Raskin. Weak subgame perfect equilibria and their application to quantitative reachability. In *24th EACSL Annual Conference on Computer Science Logic, CSL 2015, September 7-10, 2015, Berlin, Germany*, volume 41 of *LIPIcs*, pages 504–518. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015.

6 Thomas Brihaye, Julie De Pril, and Sven Schewe. Multiplayer cost games with simple nash equilibria. In *Logical Foundations of Computer Science, International Symposium, LFCS 2013, San Diego, CA, USA, January 6-8, 2013. Proceedings*, volume 7734 of *Lecture Notes in Computer Science*, pages 59–73. Springer, 2013.

7 Véronique Bruyère. Computer aided synthesis: A game-theoretic approach. In *Developments in Language Theory - 21st International Conference, DLT 2017, Liège, Belgium, August 7-11, 2017, Proceedings*, volume 10396 of *Lecture Notes in Computer Science*, pages 3–35. Springer, 2017.

8 Véronique Bruyère, Noémie Meunier, and Jean-François Raskin. Secure equilibria in weighted games. In *CSL-LICS*, pages 26:1–26:26. ACM, 2014.

9 Véronique Bruyère, Stéphane Le Roux, Arno Pauly, and Jean-François Raskin. On the existence of weak subgame perfect equilibria. *CoRR*, abs/1612.01402, 2016.

10 Véronique Bruyère, Stéphane Le Roux, Arno Pauly, and Jean-François Raskin. On the existence of weak subgame perfect equilibria. In *Foundations of Software Science and Computation Structures - 20th International Conference, FOSSACS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings*, volume 10203 of *Lecture Notes in Computer Science*, pages 145–161, 2017.

11 Krishnendu Chatterjee, Laurent Doyen, Herbert Edelsbrunner, Thomas A. Henzinger, and Philippe Rannou. Mean-payoff automaton expressions. In Paul Gastin and François Laroussinie, editors, *CONCUR 2010 - Concurrency Theory, 21th International Conference, CONCUR 2010, Paris, France, August 31-September 3, 2010. Proceedings*, volume 6269 of *Lecture Notes in Computer Science*, pages 269–283. Springer, 2010. `doi:10.1007/978-3-642-15375-4_19`.

12 Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. Strategy logic. *Inf. Comput.*, 208(6):677–693, 2010. `doi:10.1016/j.ic.2009.07.004`.

13 János Flesch, Jeroen Kuipers, Ayala Mashiah-Yaakovi, Gijs Schoenmakers, Eilon Solan, and Koos Vrieze. Perfect-information games with lower-semicontinuous payoffs. *Math. Oper. Res.*, 35(4):742–755, 2010.

14 János Flesch and Arkadi Predtetchinski. A characterization of subgame-perfect equilibrium plays in borel games of perfect information. *Math. Oper. Res.*, 42(4):1162–1179, 2017. `doi:10.1287/moor.2016.0843`.

15 Orna Kupferman, Giuseppe Perelli, and Moshe Y. Vardi. Synthesis with rational environments. *Ann. Math. Artif. Intell.*, 78(1):3–20, 2016. `doi:10.1007/s10472-016-9508-8`.

16 Donald A. Martin. Borel determinacy. *Annals of Mathematics*, pages 363–371, 1975.

17 Noémie Meunier. *Multi-Player Quantitative Games: Equilibria and Algorithms*. PhD thesis, Université de Mons, 2016.

18 Martin J. Osborne. *An introduction to game theory*. Oxford Univ. Press, 2004.

19 Eilon Solan and Nicolas Vieille. Deterministic multi-player dynkin games. *Journal of Mathematical Economics*, 39(8):911–929, 2003.

20 Michael Ummels. Rational behaviour and strategy construction in infinite multiplayer games. In *FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science, 26th International Conference, Kolkata, India, December 13-15, 2006, Proceedings*, volume 4337 of *Lecture Notes in Computer Science*, pages 212–223. Springer, 2006.

21 Nicolas Vieille and Eilon Solan. Deterministic multi-player Dynkin games. *Journal of Mathematical Economics*, Vol.39,num. 8:pp.911–929, 2003. `doi:10.1016/S0304-4068(03)00021-1`.