arXiv:1604.03773v1 [cs.GT] 13 Apr 2016

Rational verification in Iterated Electric Boolean Games

Youssouf Oualhadj* Nicolas Troquard†

LACL, U-PEC
Paris, France

Abstract

Electric boolean games are compact representations of games where the players have qualitative objectives described by LTL formulae and have limited resources. We study the complexity of several decision problems related to the analysis of rationality in electric boolean games with LTL objectives. In particular, we report that the problem of deciding whether a profile is a Nash equilibrium in an iterated electric boolean game is no harder than in iterated boolean games without resource bounds. We show that it is a PSPACE-complete problem. As a corollary, we obtain that both rational elimination and rational construction of Nash equilibria by a supervising authority are PSPACE-complete problems.

1 Introduction

We study multiagent systems populated with self-interested agents who interact repeatedly and are limited in their actions by a limited amount of energy. We investigate the computational aspects of deciding whether a collective, non-cooperative, behaviour is rational.

Electric boolean games The formalism under consideration was introduced in the second part of [19] but the decision problems were left open. They extend naturally the models of multi-player boolean games [7], one-shot electric games [19], and iterated boolean games [17]. Boolean games have occupied an important position in the recent formal AI literature. This line of work is an effort in formalisation of game theoretical situations with boolean games (see previously cited work and e.g., [24, 15]).

Strategically, the players in Iterated Electric Boolean Games (Sec. 2) are intricately mixing qualitative and quantitative considerations. Not only do they need to find a strategy that helps them satisfy their qualitative objective over time, they need to do so, seeking to keep the interaction alive so as not to run out of energy and fail to be able to perform a single action. This can be illustrated by the next simple example.

Example 1. *Isabella and Jules are two demanding kids. Isabella's objective towards happiness is to be granted a new comic book on a regular basis, and Jules' objective is to be granted a new jigsaw puzzle just as often. Their mom's objective is naturally to have all requests eventually fulfilled. Whether they ask for a new item or not, it costs zero to the kids either way. They never incur any costs. Buying a new comic book however, will cost* \$4 *to their mother, and getting a new jigsaw puzzle will cost her* \$6. *Each day, each item that is not bought will earn Mom* \$1.

*youssouf.oualhadj@lacl.fr
†nicolas.troquard@lacl.fr

Isabella and Jules, being what they are, decide that their behaviour to satisfy their objective is to ask a new item all the time. Fortunately, Mom is going to cope with it by waiting 5 *days, buying a new comic book and a new jigsaw puzzle on the* 6*-th day, and repeating. It results in a collective behaviour which is rational as we shall explain later on.*

Boolean games as compact game representations Solving problems on an input only makes sense when the input is reasonable. Possible worlds and relational semantics are commonly used to model multiagent systems. However, describing a complex system in terms of possible worlds is often unpractical. In fact, the size of the description of a system as a transition system typically grows exponentially in the number of variables in the system. For instance, model checkers for Alternating-time Temporal Logic make use of Reactive Modules [2] or Interpreted Systems [21] to overcome the difficulty. The powers of agents and coalitions are derived from the ability to control the value of some variables, thus bringing about some change to the system. Boolean games [18, 8] are such compact representations which in addition also integrate agents' preferences. They recently have been widely used to study various phenomena relevant to artificial intelligence [15, 6, 5, 16, 24].

Boolean Games are multi-player games where each player controls a set of propositional variables and has a qualitative preference represented by a propositional formula over the set of variables in the system. An action for a player is to assign a valuation to the propositional variables she controls. Iterated Boolean Games [17] are a variant of Boolean games where the players repeat the interaction infinitely often, and where their qualitative objectives are represented as LTL formulas over the set of variables in the system.

Electric Boolean Games [19] are an extension of Boolean Games where agents are assigned an initial energy endowment and taking actions has a cost, positive or negative. Already in [19], the authors define an iterated version of Electric Boolean Games, but they do not investigate their strategic aspects.

Design of safe computer systems In theoretical computer science, and particularly in the design and verification of computer systems, two-player zero-sum games have been extensively studied and used with great success [3, 22]. Recently, researchers have brought their attention to introducing quantitative restrictions for the players. For instance games where the system has to accomplish a task while maintaining its resource level above zero was modeled using Mean payoff Parity games [13], or Energy Parity Games [12]. This line of work was naturally extended by the study of the so-called multi-objective games with actual implementation [9]. In a multi objective game, a protagonist player wants to achieve a conjunction of goals, and the antagonist player wants to achieve the exact opposite. Nevertheless, the pessimistic assumption that a system and its environment always have opposite interests is not always realistic. Therefore, multiplayer games seem to be a more suitable formalism [10]. Indeed, the environment is considered to be another player with her own goal. In order to study those games, the solution concept of choice was Nash equilibria as it is a sensible formalisation of rationality [11]. In an electric boolean game, each agent has to partake in a cooperation that keeps the system alive. Namely, every single player has to make sure that none of the other players is running out of resource. This approach can be seen as an intermediate setting between non-cooperative and cooperative games. Actually, this can also be seen as a new definition of multi-objective games in the setting of multi-player games; Every player has a personal goal with no incentive to cooperate and second goal where it is best for her to cooperate.

Engineering multiagent systems Some plays of a game may appear better than others by some supervising authority. Some strategic equilibria in a game may be undesirable, while play which are not equilibria might be seen as desirable. A supervising authority could have the power to redistribute the resources available in the system so as to achieve better equilibria from their point of view. Dealing with resources such as energy, it then becomes interesting to study how much different the game would be, were the endowments of the players be different. As in [19], it is very natural to consider resource redistributions that allow one to eliminate 'bad' equilibria and/or construct 'good' equilibria.

Apart from [24] and [19], looking into ways of engineering a game's outcome has also been considered in [1]. The authors propose a framework where the winning conditions can be modified at a cost, thus changing the strategic equilibria of the game.

Contributions Our main result is the PSPACE membership for rational verification i.e., given a strategy profile decide whether it is a Nash equilibrium (Sec. 3). Note that the computational complexity in the electric case matches the one in the non-electric case. Our proof differs from the one in [17] for the non-electric case. Indeed, a straightforward adaptation of their proof would fail for it relies on a translation of the input into a well chosen LTL formula. In the electric case, one has to pay particular attention to the electric constraints (c.f., Ex. 7) which of course, are not expressible in LTL. We overcome this difficulty as follows. We construct a one-player game played on a weighted graph. This allows us to encode the behaviour of the possible deviator together with the electric constraints in an existing formalism, viz., *Energy Büchi games* [14]. We prove that a rational deviation exists iff this one-player game contains a winning strategy. The size of the constructed one-player game may be exponential in the size of the input. However, on-the-fly automata-theoretic techniques allow one to maintain a PSPACE upper-bound for the problem of finding a winning strategy. Finally, to decide in PSPACE whether a strategy profile is a Nash equilibrium, it suffices to guess a deviator and check whether she has a winning strategy in her one-player game.

Solving rational verification facilitates the access to more problems. We show (Sec. 4) that the problems of resource redistribution come out as corollaries. We leave open the more challenging problem of rational synthesis for which rational verification is a stepping stone; Rational verification is to model checking what rational synthesis is to model synthesis.

2 Iterated Electric Boolean Games

Definition 2 (Electric Boolean Games)**.** *An* electric boolean game *(EBG for short) is a tuple* $\mathcal{B} = (N, A, \Phi, c, e)$ *where:* $N = \{1, \cdots, n\}$ *is a finite set of players.* $A = \cup_{i=1}^{n} A_i$ *with* A_i *are the atoms controlled by player* i *and* $(A_1, \cdots, A_n)$ *forms a partition of* A. $\Phi = \{\phi_1, \cdots, \phi_n\}$ *where* ϕ_i *is the objective of player* i. $c : A \times \{\bot, \top\} \to \mathbb{Z}$ *is a cost function.* $e : N \to \mathbb{N}$ *is an endowment function.*

We denote $\mathcal{T}$ the set $\{\bot, \top\}$ and for any set E, $\mathcal{T}^E$ the set of mappings from E to $\mathcal{T}$, the set of all the finite sequences over E is E^* , and E^ω is the set of all the infinite sequences over E.

Let X be a set of atomic propositions, a valuation of X is a total function $v \in \mathcal{T}^X$. The cost of a valuation v is given by $\mathsf{cst}(v) = \sum_{p \in X} c(p, v(p))$. An *action* of player i is to assign a valuation to each variable in the set A_i of the atoms she controls.

We consider the setting of concurrent and infinitely repeated electric boolean games, where players choose their actions simultaneously and for an infinite duration. We consider objectives in Φ which are specified by LTL formulas over the atoms of A ([4, Chap. 5]). Formulas of LTL are defined by the following grammar: $\phi ::= p \mid \phi \wedge \phi \mid \neg\phi \mid \mathsf{X}\phi \mid \phi \mathsf{U} \phi$ where $p \in A$. The other propositional operands and temporal operators (F, G) can be defined as usual.

We need to introduce some useful terminology to talk about repeated games and define the semantics of LTL formulas over $\left(\mathcal{T}^A\right)^\omega$.

A *history* in a repeated electric boolean game is a word in $\left(\mathcal{T}^A\right)^*$. That is, a finite sequence of valuations for the set A of boolean variables. A *play* is an infinite sequence in $\left(\mathcal{T}^A\right)^\omega$. Given a play ρ, we note $\rho[t]$ the t-th valuation function in ρ. We note $\rho[t\ldots]$ the suffix of ρ starting at $\rho[t]$, and $\rho[\ldots t]$ the prefix of ρ ending at $\rho[t]$ which is a history of size $t+1$.

LTL objectives are evaluated over a play ρ of the game. For $p \in A$, and for ϕ and ψ two LTL formulas:

$\rho \models p$ iff $\rho[0](p) = \top$

$\rho \models \neg\phi$ iff $\rho \not\models \phi$

$\rho \models \mathsf{X}\phi$ iff $\rho[1\ldots] \models \phi$

$\rho \models \phi \wedge \psi$ iff $\rho \models \phi$ and $\rho \models \psi$

$\rho \models \phi \mathsf{U} \psi$ iff $\exists i \geq 0,\ \rho[i\ldots] \models \psi$ and $\forall 0 \leq j < i,\ \rho[j\ldots] \models \phi$

The formula $\mathsf{X}\phi$ holds true on ρ if ϕ is true next. The formula $\phi\mathsf{U}\psi$ holds true on ρ if ϕ is true at least until ψ is true.

In order to play, the players choose their actions according to a strategy. A *strategy* for player i is a mapping that takes as input a history and outputs a valuation for each atom controlled by player i. Formally a strategy σ_i for player i is a mapping $\sigma_i : \left(\mathcal{T}^A\right)^* \to \mathcal{T}^{A_i}$. We note Σ_i the set of strategies of player i.

A *strategy profile* σ is a vector $(\sigma_1, \cdots, \sigma_n)$ specifying one strategy σ_i for each player $i \in N$. Given a strategy profile $\sigma = (\sigma_1, \cdots, \sigma_n)$ and a strategy τ_i for player i, we note (τ_i, σ_{-i}) the strategy profile $(\sigma_1, \cdots, \tau_i, \cdots, \sigma_n)$. Each strategy profile induces a play, and since we consider pure strategies, there is one and only one such play consistent with σ. We denote $\langle\sigma\rangle$ the play induced by the profile σ. It is defined inductively as follows: if $p \in A_i$ then $\langle\sigma\rangle[0](p) = \sigma_i(\epsilon)(p)$, and for $t \geq 0$, $\langle\sigma\rangle[t+1](p) = \sigma_i(\langle\sigma\rangle[\ldots t])(p)$.

The endowment $e(i)$ of each player i specified in the definition of an electric boolean game, represents the initial resources of the player. While playing the game following a strategy, this endowment grows as the player takes an action of negative cost and shrinks as the player takes an action of positive cost.

We will say that the strategy profile σ is feasible in an iterated EBG if it does not over-consume the endowed resources, in the sense that, every player's strategy σ_i can be infinitely executed without ever causing the player's compound endowment to go under 0. We make it more formal.

Consider an EBG (N, A, Φ, c, e) and a strategy profile σ. The *compound endowment* of player i at the t-th step of the play $\langle\sigma\rangle$ is defined with $\mathsf{E}_i^\sigma(0) = e(i)$, and

$$\mathsf{E}_i^\sigma(t+1) = \mathsf{E}_i^\sigma(t) - \mathsf{cst}(\sigma_i(\langle\sigma\rangle[\ldots t]))$$

Thus, the strategy profile σ is *feasible* iff for each player $i \in N$, and for all $t \geq 0$ we have $\mathsf{E}_i^\sigma(t) \geq 0$. In the strategy profile σ, we say that τ_i is a *feasible deviation* for player i iff (τ_i, σ_{-i}) is a feasible strategy profile.

Once an objective ϕ_i and a strategy profile σ are fixed, the payoff of σ for player i is defined as follows:

$$\mathsf{Payoff}_i(\sigma) = \begin{cases} 1 & \text{if } \sigma \text{ is feasible, and } \langle\sigma\rangle \models \phi_i \ , \\ 0 & \text{otherwise.} \end{cases}$$

In the strategy profile σ, we say that τ_i is a *rational deviation* for player i iff $\mathsf{Payoff}_i((\tau_i, \sigma_{-i})) > \mathsf{Payoff}_i(\sigma)$.

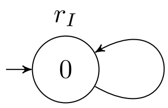
Example 3. *We formalise the game of Example 1 and model a strategy for the three participants. Let $\mathcal{B}^{c,e}$ be an EBG (N, A, Φ, c, e) where $N = \{I, J, M\}$, $A_I = \{r_I\}$, $A_I = \{r_J\}$, $A_M = \{g_I, g_J\}$.*

Evaluated to $\top$, the atoms r_I, r_J, g_I, g_J, respectively represent the facts that Isabella asks for a comic book, Jules asks for a jigsaw puzzle, Mom buys a comic book, and Mom buys a jigsaw puzzle. The costs are given by $c(r_I, \top) = c(r_I, \bot) = c(r_J, \top) = c(r_J, \bot) = 0$, and $c(g_I, \bot) = c(g_J, \bot) = -1$, $c(g_I, \top) = 4$, and $c(g_J, \top) = 6$. We suppose that $e(I) = e(J) = e(M) = 0$. The objectives are given as $\Phi_M = \mathsf{G}((r_I \to \mathsf{F}(g_I)) \wedge (r_J \to \mathsf{F}(g_J)))$, $\Phi_I = \mathsf{GF}(g_I)$, and $\Phi_J = \mathsf{GF}(g_J)$. The strategies of the kids continuously asking a new item and of the Mom buying one comic book and one jigsaw puzzle every 6 days result in a strategy profile whose payoff is 1 for everyone.

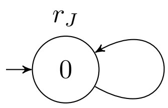
The strategies suggested at the end of Example 3 are depicted in Figure 1. They are instances of what we call *finite memory* strategies. We formalise the class of finite memory strategies next.

Definition 4 (Finite memory strategy)**.** *Let $i \in N$ be a player, a finite memory strategy σ_i for player i consists of a finite set M called the memory, an initial memory state m^{in} in M, a mapping $\sigma_i^{\mathfrak{U}} : M \times \mathcal{T}^A \to M$ called the update function, and a mapping $\sigma_i^{\mathfrak{C}} : M \to \mathcal{T}^A$ called the choice function.*

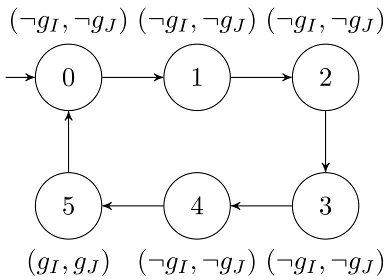
We say that $(\sigma_1, \cdots, \sigma_n)$ is a *finite memory* profile if for every $i \in N$, σ_i is a finite memory strategy. For instance, in the strategy of Figure 1c, the set M is $\{0, 1, 2, 3, 4, 5\}$, the initial memory state is 0, the update function is the edge relation and the choice function is illustrated by labels next to vertices[1].



(a) Isabella's strategy.



(b) Jules' strategy.



(c) Mom's strategy.

Figure 1: A finite memory profile seen as finite graphs.

3 Nash Equilibria in Electric Boolean Games

In [19], the authors introduced iterated electric boolean games but did not study their strategic aspects. Hence no solution concept was defined. However, the concept of Nash equilibria is one of most natural concept in multiplayer games.

Definition 5 (Nash equilibrium)**.** *Let $\mathcal{B}^{c,e}$ be an EBG and σ be a strategy profile. We say that σ is a* Nash equilibrium *iff the following holds:*

1. $\forall t \geq 0,\ \forall i \in N,\ \mathsf{E}_i^\sigma(t) \geq 0$,
2. $\forall i \in N,\ \forall \tau_i \in \Sigma_i\ , \mathsf{Payoff}_i((\tau_i, \sigma_{-i})) \leq \mathsf{Payoff}_i(\sigma)$.

Using our terminology, σ is a Nash equilibrium in $\mathcal{B}^{c,e}$ if and only if it is feasible and there is no rational deviation for any player. We note $\mathsf{NE}(\mathcal{B}^{c,e})$ the set of Nash equilibria in the game $\mathcal{B}^{c,e}$. For instance, the strategy profile depicted in Figure 1 is a Nash equilibrium in the game of Examples 1 and 3

[1]We omit the labels on the edges to highlight that for each player the update function depends only on the current memory state.

Definition 6 (Nash Equilibrium Membership)**.** *Let $\mathcal{B}^{c,e}$ be an electric boolean game, and σ be a finite memory strategy profile. The Nash Equilibrium Membership (NEM) problem asks whether $\sigma \in \mathsf{NE}(\mathcal{B}^{c,e})$.*

In order to build intuition regarding deviations, consider the following example

Example 7. *Let $\mathcal{B}^{c,e}$ be the following two-player game,*

$$\begin{aligned}
&A_1 = \{p\},\ A_2 = \{q\} \ ,\\
&\phi_1 \equiv \mathsf{G}\left((q \to \mathsf{X}p) \wedge (\neg q \to \mathsf{X}\neg p)\right),\ \phi_2 \equiv \mathsf{G}q \ ,\\
&c(p,\top) = 1,\ c(p,\bot) = -1,\ c(q,\top) = c(q,\bot) = 0,\ e(1) = e(2) = 0 \ .
\end{aligned}$$

Consider the following strategy σ_1 for player 1 that assings $\top$ to p iff $\top$ was assigned to q the previous round. We also consider the strategy σ_2 for player 2 that always assigns $\bot$ to q.

We argue that the profile (σ_1, σ_2) is a Nash equilibrium. Clearly (σ_1, σ_2) is feasible. Let us show that player 2 does not have a rational deviation. In order to increase her payoff, player 2 has to always assign $\top$ to q, call this new strategy τ. However, the deviation τ is not feasible. Indeed, player 1 is still following σ_1, we obtain

$$\begin{aligned}
&\sigma_1(\epsilon)(p) = \bot \textit{ with } \mathsf{E}_1^{(\sigma_1,\tau)}(1) = 1 \ ,\\
&\sigma_1(\{(p,\bot),(q,\top)\})(p) = \top \textit{ with } \mathsf{E}_1^{(\sigma_1,\tau)}(2) = 0 \ ,\\
&\sigma_1(\{(p,\bot),(q,\top)\}\{(p,\top),(q,\top)\})(p) = \top \textit{ with } \mathsf{E}_1^{(\sigma_1,\tau)}(3) = -1 \ ,
\end{aligned}$$

showing that the compound endowment drops below 0 after the third round. The plays induced by the two profiles are depicted in Figure 2.

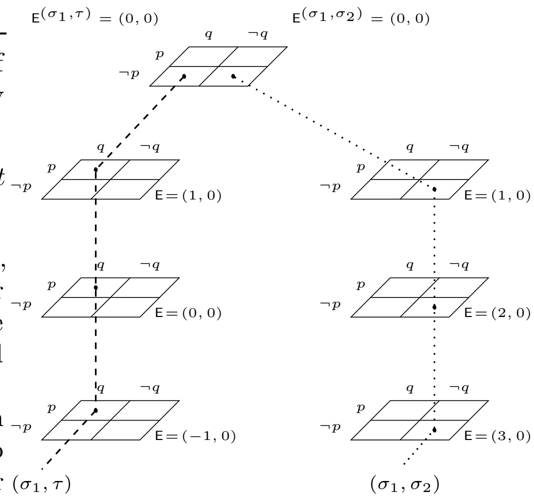


Figure 2: plays induced by the profiles (σ_1, σ_2) and (σ_1, τ).

This example shows that in order to perform a rational deviation, a player has to check the endowment of all the players and not only her own. We are now ready to state the main theorem of this paper.

Theorem 8. *NEM is a* PSPACE*-complete problem. It is* PSPACE*-hard even when there is only one player.*

To prove the theorem, we exhibit two constructions, c.f. Construction 1, and Construction 2. The former allows one to check the feasibility of a profile, while the latter allows one to check the existence of a rational deviation.

In Section 3.1, and Section 3.2 we let $\mathcal{B}^{c,e}$ be an EBG, and σ be a finite memory profile. Let also $(M_i, m_i^{in}, \sigma_i^{\mathfrak{U}}, \sigma_i^{\mathfrak{C}})$ be the finite memory strategy of player i in the profile σ.

3.1 Checking feasibility in PSPACE

We say that G is a *d-weighted graph* if G is associated with a weight function $w : E \to \mathbb{Z}^d$. For a vertex u and a vector w_0 in $\mathbb{N}^d$, a subset C of V is a nonnegative reachable cycle from u if the following holds. (i) There exists v in $C = \{u_j \mid l \leq j \leq k\}$, and a path $u_0, \cdots, u_l, \cdots u_k$ such that $u_0 = u$, $u_l = v$, and $u_k = v$. (ii) For all $0 \leq t \leq k-1$ we have $w_0 - \sum_{j=0}^{t} w(u_j, u_{j+1}) \geq \{0\}^d$, and $\sum_{j=l}^{k-1} w(u_j, u_{j+1}) \leq \{0\}^d$. Positive cycles are defined as expected.

In order to prove Proposition 10 we use the results of [20]. In particular, given a d-weighted graph G, we can detect a *nonnegative reachable cycle* in polynomial time in the size of G.[2]

Our approach consists in constructing a n-weighted graph $G[\sigma]$ from the finite memory profile σ. This is achieved by Construction 1. We show that $G[\sigma]$ contains such a cycle iff σ is feasible.

We start first by giving the details of how $G[\sigma]$ is obtained.

Construction 1. *$G[\sigma]$ consists of a finite set of vertices V, an edge relation $E \subseteq V \times V$, and weight function $w : E \to \mathbb{Z}^n$. $G[\sigma]$ is obtained as follows:*

- *The vertices are $V = \prod_{i \in N} M_i$.*
- *For $v \in V$ we denote v_i the i-th component of v. Let $(u, v) \in V \times V$ be a couple of vertices, (u, v) is an edge in E if for each $i \in N$ we have $\sigma_i^{\mathfrak{U}}(u_i, X) = v_i$ where $X = \bigcup_{j \in N} \sigma_j^{\mathfrak{C}}(u_j)$ is the complete valuation over A prescribed by the profile σ.*
- *Finally, for $(u, v) \in E$,*

$$w(u, v) = \big(\mathsf{cst}(\sigma_1^{\mathfrak{C}}(u_1)), \cdots, \mathsf{cst}(\sigma_n^{\mathfrak{C}}(u_n))\big) \quad .$$

The following lemma states the key property of Construction 1.

Lemma 9. *The finite memory strategy profile σ is feasible iff $G[\sigma]$ has a nonnegative reachable cycle from $u^0 = (m_1^{in}, \ldots, m_n^{in})$ with initial credit e.*

A consequence of the above lemma is

Proposition 10. *We can check in* PSPACE *whether σ is feasible.*

3.2 Checking the existence of rational deviation in PSPACE

Now that we can check whether a profile is feasible, we need to show how to check the existence of rational deviation for a player.

We recall that $\mathcal{B}^{c,e}$, σ, and $\sigma_i = (M_i, m_i^{in}, \sigma_i^{\mathfrak{U}}, \sigma_i^{\mathfrak{C}})$ are still fixed.

We need to introduce some technical material. A Büchi automaton $\mathcal{A}$ is a tuple $\mathcal{A} = (Q, q_0, A, \Delta, F)$ where the Q is a finite set of states, q_0 is an initial state, A is a finite alphabet, Δ is relation in $Q \times A \times Q$, and F is a subset of states called accepting. We say that an infinite word w is recognised by $\mathcal{A}$ if there exists an infinite path ρ in $\mathcal{A}$ labelled by w such that ρ visits states in F infinitely many times. We also say that ρ is a run induced by w on $\mathcal{A}$. We define $\mathcal{L}_{\mathcal{A}}$ as the set of words recognised by $\mathcal{A}$. The reason we need Büchi automata is their strong link with LTL. Indeed, any LTL formula ϕ, can be associated to a Büchi automaton accepting all its models. The following theorem formalises this idea.

Theorem 11. *Let ϕ be a LTL formula, there exists a Büchi automaton $\mathcal{A}_\phi$ accepting the language $\mathcal{L}_\phi$ consisting of all the models of ϕ.*

The other formalism is *one-player games*. Let $G = (V, E, W)$ be a graph with a set of vertices V, a set of edges $E \subseteq V \times V$, and winning objective $W \subseteq V^\omega$. Strategies for these games are formalised by the following mapping $V^*V \to V$. Let σ be a strategy for the player, and u_0 a vertex in V. The play ρ starting in u_0 and consistent with σ is obtained as follows: $\rho[0] = u_0$, and for all $i > 0$, $\sigma(\rho[\ldots i])$. The player wins if the play ρ is in W. A strategy σ is winning for

[2] The result of [20] is to find 0-cycles. To find nonnegative cycles, it suffices to transform a weighted graph G into G' by adding a reflexive edge of weight -1 to every vertice. This is a polynomial transformation. G has a nonnegative cycle iff G' has a zero-cycle.

the player from u_0 if the play consistent with σ is in W. Finite memory strategies can be defined in a similar fashion as for EBGs. In this paper, we use the so-called multi-objective games. Those are games where the player has to fulfil a combination of objectives at once.

Büchi objectives. We choose a set $F \subseteq V$ of accepting vertices. The winning objective W is $(V^*F)^\omega$. We denote this winning objective Buchi.

Energy objectives. Let $d > 0$ be a natural, $w_0 \in \mathbb{N}^d$ be an initial vector, and $w : E \to \mathbb{Z}^d$ be an energy function. The winning objective is the set $\{u_0 u_1 \cdots \in V^\omega \mid \forall k \geq i,\ w_0 - \sum_{i=0}^{k} w(u_i, u_{i+1}) \geq \{0\}^d\}$. We denote this winning objective Energy.

The winning objective we are interested in is $\mathsf{EnergyBuchi}$ defined by $\mathsf{Buchi} \cap \mathsf{Energy}$.

Roughly speaking, given a profile σ and a player i, we construct an $\mathsf{EnergyBuchi}$ game $G[\sigma_{-i}]$. The purpose of this latter, is that it will contain a winning strategy iff a rational deviation exists. Moreover, the winning strategy in $G[\sigma_{-i}]$ will be the deviation player i uses to increase her payoff. Let us explain how to construct the one player game $G[\sigma_{-i}]$.

Construction 2. *We note V the set of vertices in $G[\sigma_{-i}]$, E the edge relation defined over $V \times \mathcal{T}^A \times V$, and the weight function w is a mapping from $V \times \mathcal{T}^A \to \mathbb{Z}^n$.*

Let $\mathcal{A}_i = (Q, \mathcal{T}^{A_{\phi_i}}, q_0, \Delta, F)$ be an automaton accepting the language $\mathcal{L}_{\phi_i}$.

The graph $G[\sigma_{-i}]$ is obtained as follows:

- *The vertices are $V = Q \times \prod_{j \in N \setminus \{i\}} M_j$.*
- *Let v be a vertex in V, for $j \in N \setminus \{i\}, v_j$ refers to the j-th component of v and v_i is the projection over Q. For $(u, v) \in V \times V$, and for every valuation $X \in \mathcal{T}^A$ we have (u, X, v) in E if*
  - *i) there exists $Y \in \mathcal{T}^{A_{\phi_i}}$ such that $(u_i, Y, v_i) \in \Delta$ and $Y \subseteq X$,*
  - *ii) the set $Z = Y \cup \bigcup_{j \in N \setminus \{i\}} \sigma_j^{\mathfrak{C}}(u_j) \subseteq X$ and is consistent over A_{ϕ_i} i.e.*

    $$\forall p \in A_{\phi_i},\ (p, \top) \in Z \implies (p, \bot) \notin Z \ ,$$

  - *iii) for each $j \in N \setminus \{i\}$ we have $\sigma_j^{\mathfrak{U}}(u_j, X) = v_j$.*
- *The weight function is given by $\mathsf{cst}(\sigma_j^{\mathfrak{C}}(u_j))$ for every dimension $j \in N \setminus \{i\}$ and by $\sum_{p \in A_i} c(p, X(p))$ for dimension i.*
- *Finally, a vertex $v \in V$ is accepting if $v_i \in F$.*

The intuition behind this construction is as follows. If player i can deviate rationally, then necessarily the new profile satisfies ϕ_i. This is why we use automaton $\mathcal{A}_{\phi_i}$ whose language is exactly those words that satisfy ϕ_i. Also, since we consider only unilateral deviations, the actions leading to the satisfaction of ϕ_i have to be compatible with the choices of other players, that is σ_{-i}. This is ensured by *ii)*. Item *iii)* is a synchronisation between the action of the other player and the deviation of player i.

Thanks to the following lemma, we show that Construction 2 meets the desired intuition.

Lemma 12. *Let σ be a finite memory profile, and i be a player such that $\mathsf{Payoff}_i(\sigma) = 0$ then, i has a rational deviation iff there exists a winning strategy in $G[\sigma_{-i}]$.*

As a consequence we obtain the core property for the existence of our PSPACE algorithm.

Proposition 13. *Let σ be a finite memory profile, and i be a player such that $\mathsf{Payoff}_i(\sigma) = 0$. We can check whether i has a rational deviation in PSPACE.*

3.3 Proof of Theorem 8

We recall Theorem 8

Theorem 8. *NEM is a* PSPACE*-complete problem. It is* PSPACE*-hard even when there is only one player.*

Proof. If the profile is not feasible, return "no". Otherwise, guess a possible deviator i (among the players with null payoff) and check whether she has a winning strategy in $G[\sigma_{-i}]$. Return "no" iff she has a winning strategy. Lemma 9 and Lemma 12 justify the correctness. Proposition 10 and Proposition 13 justify the upper-bound complexity.

To establish the hardness, one needs to notice that any BG is an EBG with endowment $\{0\}^N$ and $c : A \times \mathcal{T} \to \{0\}$. Thus the PSPACE lower bound established in [17, Prop. 2] holds for EBGs with LTL specifications. Since the proof is a reduction from LTL satisfiability to one-player iterated boolean games, NEM is hard even when there is only one player. $\square$

4 Resource redistributions

Having characterized the complexity of the problem of deciding whether a strategy profile of an iterated EBG is a Nash equilibrium, we will see how we can easily tackle derived decision problems for engineering Electric Boolean Games.

A resource redistribution for an EBG $\mathcal{B} = (N, \Sigma, \Phi, c, e)$ is an endowment function $e' : N \to \mathbb{N}$ such that

$$\sum_{i \in N} e(i) = \sum_{i \in N} e'(i).$$

Remark 14. *Let an EBG* $\mathcal{B} = (N, \Sigma, \Phi, c, e)$*. There is finite number of resource redistributions for* $\mathcal{B}$*.*

In [19], the authors studied the problems of determining whether there is a resource redistribution such that a strategy profile is a Nash Equilibrium (rational construction), and of determining whether there is a resource redistribution such that a strategy profile is not a Nash Equilibrium (rational elimination). For the iterated setting we propose the following decision problems.

Definition 15 (Construction and elimination)**.** *Let* $\mathcal{B}$ *be an electric boolean game, and* σ *be a finite memory strategy profile. The Rational Construction (RC) problem asks whether there is a resource redistribution such that* σ *is a Nash equilibrium.The Rational Elimination (RE) problem asks whether there is a resource redistribution such that* σ *is not a Nash equilibrium.*

Theorem 16. *The RC problem and the RE problem are* PSPACE*-complete.*

The non-deterministic procedures outlined in the proof of Theorem 16 are sufficient to characterize an optimal upper-bound of the problems. In the case of RE, there exists a more practical deterministic algorithm. Indeed, the result of [19, Corr. 4] carries over in the iterated setting.

Proposition 17. *Let an endowment e be given. The endowment* e^i *is the resource redistribution of e such that all resources are allocated to player* i*. The strategy profile* σ *is eliminable in* $\mathcal{B}^{c,e}$ *iff for some player* i*,* $\sigma \notin \mathsf{NE}(\mathcal{B}^{c,e^i})$*.*

This hints at a "more practical" algorithm to solve RE: for each player i, test whether $\sigma \notin \mathsf{NE}(\mathcal{B}^{c,e^i})$. Return "yes" as soon as a test succeeds. Return "no" when all $|N|$ tests failed.

5 Conclusion

In this paper we presented a preliminary result on the Electric Boolean Games introduced in [19]. We considered the iterated setting where the objectives are specified as LTL formulas. We showed the PSPACE-completness of Nash equilibrium membership, thus matching the complexity bounds of [17] for the non quantitative setting of iterated Boolean Games. In order to establish this result, we extended existing techniques for plain LTL to an extension of LTL with electric constraints. This result is used to characterise the complexity of two problems of resource redistribution that can serve at social-welfare engineering.

As future research direction, we plan to investigate the Nash equilibrium non-emptyness and Nash equilibrium synthesis. We believe that Construction 2 can be extended in order to construct a concurrent game with the property that it contains a pure Nash equilibrium iff the electric boolean game does. To the best of our knowledge, the obtained class of concurrent games is rather novel and has yet to be studied.

References

[1] Almagor, S., Avni, G., Kupferman, O.: Repairing multi-player games. In: CONCUR 2015. Volume 42 of LIPIcs., Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2015) 325–339

[2] Alur, R., Henzinger, T.A., Mang, F.Y.C., Qadeer, S., Rajamani, S.K., Tasiran, S.: MOCHA: modularity in model checking. In: CAV 1998, Springer (1998) 521–525

[3] Asarin, E., Maler, O., Pnueli, A.: Symbolic controller synthesis for discrete and timed systems. In: Hybrid Systems II. (1994) 1–20

[4] Baier, C., Katoen, J.P.: Principles of Model Checking. The MIT Press (2008)

[5] Bonzon, E., Devred, C., Lagasquie-Schiex, M.: Argumentation and CP-Boolean Games. International Journal on Artificial Intelligence Tools **19**(4) (2010) 487–510

[6] Bonzon, E., Lagasquie-Schiex, M., Lang, J.: Dependencies between players in boolean games. International Journal of Approximate Reasoning **50**(6) (2009) 899–914

[7] Bonzon, E., Lagasquie-Schiex, M., Lang, J., Zanuttini, B.: Boolean games revisited. In: ECAI 2006. Volume 141 of Frontiers in Artificial Intelligence and Applications., IOS Press (2006) 265–269

[8] Bonzon, E., Lagasquie-Schiex, M., Lang, J., Zanuttini, B.: Compact preference representation and boolean games. Autonomous Agents and Multi-Agent Systems **18**(1) (2009) 1–35

[9] Brázdil, T., Chatterjee, K., Forejt, V., Kucera, A.: Multigain: A controller synthesis tool for mdps with multiple mean-payoff objectives. In: TACAS 2015. (2015) 181–187

[10] Brenguier, R., Clemente, L., Hunter, P., Pérez, G.A., Randour, M., Raskin, J., Sankur, O., Sassolas, M.: Non-zero sum games for reactive synthesis. In: LATA 2016. (2016) 3–23

[11] Brihaye, T., De Pril, J., Schewe, S.: Multiplayer cost games with simple nash equilibria. In: LFCS 2013. (2013) 59–73

[12] Chatterjee, K., Doyen, L.: Energy parity games. Theor. Comput. Sci. **458** (2012) 49–60

[13] Chatterjee, K., Henzinger, T.A., Jurdzinski, M.: Mean-payoff parity games. In: (LICS 2005). (2005) 178–187

[14] Chatterjee, K., Randour, M., Raskin, J.F.: Strategy synthesis for multi-dimensional quantitative objectives. Acta Informatica **51**(3-4) (2014) 129–163

[15] Dunne, P.E., van der Hoek, W., Kraus, S., Wooldridge, M.: Cooperative boolean games. In: AAMAS 2008, IFAAMAS (2008) 1015–1022

[16] Grant, J., Kraus, S., Wooldridge, M., Zuckerman, I.: Manipulating boolean games through communication. In: IJCAI 2011, IJCAI/AAAI (2011) 210–215

[17] Gutierrez, J., Harrenstein, P., Wooldridge, M.: Iterated boolean games. Information and Computation **242** (2015) 53–79

[18] Harrenstein, P.: Logic in conflict. PhD thesis, Utrecht University (2004)

[19] Harrenstein, P., Turrini, P., Wooldridge, M.: Electric Boolean Games: Redistribution Schemes for Resource-Bounded Agents. In: AAMAS 2015, ACM (2015) 655–663

[20] Kosaraju, S.R., Sullivan, G.F.: Detecting cycles in dynamic graphs in polynomial time (preliminary version). In: STOC 1988, ACM (1988) 398–406

[21] Lomuscio, A., Qu, H., Raimondi, F.: MCMAS: A model checker for the verification of multi-agent systems. In: CAV 2009, Springer (2009) 682–688

[22] Tripakis, S., Altisen, K.: On-the-fly controller synthesis for discrete and dense-time systems. In: FM'99. (1999) 233–252

[23] Vardi, M.Y.: An automata-theoretic approach to linear temporal logic. In: Logics for Concurrency. Volume 1043 of Lecture Notes in Computer Science., Springer (1995) 238–266

[24] Wooldridge, M., Endriss, U., Kraus, S., Lang, J.: Incentive engineering for boolean games. Artificial Intelligence **195** (2013) 418 – 439

A Proofs of Section 3

A.1 Lemma 9

We state technical yet useful remarks.

Let $h_1 \cdots h_l \in \left(\mathcal{T}^A\right)^+$ be a finite history. We denote $\mathcal{M}_i : \left(\mathcal{T}^A\right)^+ \to M_i$ the operator that gives the memory state of player i after h has occurred. We define $\mathcal{M}_i$ inductively by:

$$\mathcal{M}_i(h_1) = \sigma_i^{\mathfrak{U}}(m_i^{in}, h_1) \ ,$$
$$\mathcal{M}_i(h_1 \cdots h_l) = \sigma_i^{\mathfrak{U}}(\mathcal{M}_i(h_1 \cdots h_{l-1}), h_l) \ .$$

The action played after h by player i is:

$$\sigma_i(h) = \sigma_i^{\mathfrak{C}}(\mathcal{M}_i(h)) \ .$$

Remark 18. *Let σ be a finite memory profile, then for history h of size l*

$$\langle \sigma \rangle[\dots l] = \left(\sigma_1^{\mathfrak{C}}(\mathcal{M}_1(h)) \cup \cdots \cup \sigma_n^{\mathfrak{C}}(\mathcal{M}_n(h))\right) \ .$$

Remark 19. *By definition of finite memory strategies, the following holds*

$$((u, v) \in E) \wedge ((u, v') \in E) \implies v = v' \ .$$

The above remark implies that the edge relation in $G[\sigma]$ is functional, hence from every vertex there exists a unique infinite path in $G[\sigma]$.

We can now carry on with the proof. We recall Lemma 9

Lemma 9. *The finite memory strategy profile σ is feasible iff $G[\sigma]$ has a nonnegative reachable cycle from $u^0 = (m_1^{in}, \dots, m_n^{in})$ with initial credit e.*

Proof. We show the direct implication. Suppose σ is a finite memory strategy profile in $\mathcal{B}^{c,e}$. We can construct the path $(u^t)_{t \geq 0}$ in $G[\sigma]$ inductively as

$$u^{in} = (m_1^{in}, \cdots, m_n^{in}) \ ,$$

and for every $t > 0$

$$u^t = (\sigma_1^{\mathfrak{U}}(u_1^{t-1}, \langle \sigma \rangle[t-1]), \cdots, \sigma_n^{\mathfrak{U}}(u_n^{t-1}, \langle \sigma \rangle[t-1])) \ .$$

Clearly, $(u^t)_{t \geq 0}$ is the unique infinite path in $G[\sigma]$, and since V is finite there exist $k \geq 0$ and $l > k$ such that $u^l = u^k$ and for every $k < j < l$ we have $u^j \neq u^{j+1}$.

Since the strategy profile σ is feasible in $\mathcal{B}^{c,e}$, it follows that that $u^k \cdots u^l$ is a cycle that satisfies the proposition. This is a consequence of the following facts. Feasibility of σ means that

$$\forall t \geq 0, \ \forall i \in N, \ \mathsf{E}_i^{\sigma}(t) \geq 0 \ . \tag{1}$$

Equation (1) implies that the cumulative weight of the path in $G[\sigma]$ from $u^0 \cdots u^k \cdots u^l$ is nonnegative on all the dimensions. Moreover, the cumulative weight of the cycle $u^k \cdots u^l$ is necessarily nonnegative on all the dimensions, otherwise it would indicate that the profile σ eventually depletes one player's endowment, and contradict that σ is feasible. In other words, $u^k \cdots u^l$ is a nonnegative reachable cycle in $G[\sigma]$ with initial credit e.

We show the converse implication. We will show that for any path ρ of length k in $G[\sigma]$ we have:

$$(\mathsf{E}_1^\sigma(k), \cdots, \mathsf{E}_n^\sigma(k)) = e - W(k) \ , \tag{2}$$

where W is the cumulative weight of ρ at step t inductively defined as follows:

$$W(t) = \begin{cases} 0 \text{ if } t = 0 \\ W(t-1) + w(\rho[t-1], \rho[t]) \text{ if } 0 < t < |\rho| \end{cases}$$

Equation 2 is enough to prove the desired implication. Indeed, since $G[\sigma]$ has a unique infinite path (c.f. Remark 19, it follows that there exists exactly one reachable cycle and by assumption this reachable cycle is nonnegative thus the cumulative weight is always nonnegative for any given length k thus the same will hold for every compound endowment finishing the proof.

In order to establish Equation (2), we use the following equation

$$w(\rho[k], \rho[k+1]) = (\mathsf{cst}(\sigma_1(\langle\sigma\rangle[\dots k]), \cdots, \mathsf{cst}(\sigma_n(\langle\sigma\rangle[\dots k])) \ . \tag{3}$$

Equation (3) can be derived inductively for any $k \geq 0$ using Remark 18.

We prove Equation 2 by induction over k.
For $k = 0$,

$$(\mathsf{E}_1^\sigma(0), \cdots, \mathsf{E}_n^\sigma(0)) = e - W(0) = e - 0 \ ,$$

where the second equality is by definition, hence the property holds.

Now assume that for $k \geq 0$ we have $(\mathsf{E}_1^\sigma(k), \cdots, \mathsf{E}_n^\sigma(k)) = e - W(k)$.

Let us show that property hold for any path of length $k + 1$.

$$\begin{aligned} e - W(k+1) &= e - W(k) - w(\rho[k], \rho[k+1]) \\ &= (\mathsf{E}_1^\sigma(k), \cdots, \mathsf{E}_n^\sigma(k)) - w(\rho[k], \rho[k+1]) \\ &= (\mathsf{E}_1^\sigma(k), \cdots, \mathsf{E}_n^\sigma(k)) - (\mathsf{cst}(\sigma_1(\langle\sigma\rangle[\dots k]), \cdots, \mathsf{cst}(\sigma_n(\langle\sigma\rangle[\dots k])) \\ &= (\mathsf{E}_1^\sigma(k+1), \cdots, \mathsf{E}_n^\sigma(k+1)) \ , \end{aligned}$$

where the first equality is by definition of W, the second equality is by induction hypothesis, the third equality by Equation (3), and the last equality from the definition of the compound endowment. ☐

A.2 Proof of Proposition 10

Proposition 10. *We can check in* PSPACE *whether σ is feasible.*

Proof. From Lemma 9, we know that a profile is feasible if and only if the graph $G[\sigma]$ has a nonnegative reachable cycle. Thus, we show how to detect the latter using only polynomial space in the size of the description of $\mathcal{B}^{c,e}$ and σ. Notice that $G[\sigma]$ is of size exponential in the description of the input. Fortunately, checking the reachability in a graph is known to be in NLOGSPACE. Moreover, it can be performed using on-the-fly techniques. (The complete argument would be analogous to the proofs of complexity for LTL [23, 4].) This allows one to detect a cycle without storing the entire representation of $G[\sigma]$. For the weight vectors, we need to keep track of the accumulation of the weights which is at most $|\sigma|^n C$ on each dimension where $C = \max(\{z \in \mathbb{Z} \mid \exists p \in A,\ z = |c(p, \bot)|\} \cup \{z \in \mathbb{Z} \mid \exists p \in A,\ z = |c(p, \top)|\})$. This quantity needs a memory of size $O(n^2 \log(|\sigma|C))$, establishing the PSPACE upper-bound. ☐

A.3 Proof of Lemma 12

We will need the following result about EnergyBuchi one-player games.

Theorem 20 ([14]). *Let G be a* EnergyBuchi *one-player game, u_0 be a vertex, it is decidable wether the player has a winning strategy from u_0. Moreover, winning strategies can be implemented using finite memory.*

Lemma 12. *Let σ be a finite memory profile, and i be a player such that* $\mathsf{Payoff}_i(\sigma) = 0$ *then, i has a rational deviation iff there exists a winning strategy in $G[\sigma_{-i}]$.*

Proof. We start with the direct implication. We note τ the rational deviation of player i. We will show that τ is a winning strategy in $G[\sigma_{-i}]$.

Let us show that τ is winning for Buchi. By definition of a rational deviation we know that $\langle(\sigma_{-i}, \tau)\rangle \models \phi_i$. Second, by construction of the edge relation E (c.f. Construction 2, *i*), *ii*), and *iii*)) we know that there exists an infinite path labelled by $\langle(\sigma_{-i}, \tau)\rangle$ in $G[\sigma_{-i}]$. By Theorem 11 we know that this path visits states in F infinitely often. Thus τ satisfies the objective Buchi in $G[\sigma_{-i}]$.

Second let us show that τ is winning for Energy. Again by definition of rationality, (σ_{-i}, τ) is feasible. Now, notice that we can apply Construction 1 to obtain the d-weighted graph $G[(\sigma_{-i}, \tau)]$. By Lemma 9, since (σ_{-i}, τ) is feasible, it will contain a nonnegative reachable cycle. By definition of nonnegative reachable cycles it follows that τ is winning for Energy.

Let us prove the converse implication. Let τ be a winning strategy in $G[\sigma_{-i}]$. We show that τ is a rational deviation i.e.

- (σ_{-i}, τ) is feasible.
- $\langle(\sigma_{-i}, \tau)\rangle \models \phi_i$.

By Theorem 20, we can assume without loss of generality that τ is a finite memory strategy. Since it is winning for Buchi, it follows that the play consistent with τ visits states in F infinitely often. By Theorem 11 it follows that this play is a model for ϕ_i. Thus $\langle(\sigma_{-i}, \tau)\rangle \models \phi_i$. Since τ is winning for Energy, we know that $G[(\sigma_{-i}, \tau)]$ contains a nonnegative reachable cycle (c.f. Construction 1). By Lemma 9 it follows that (σ_{-i}, τ) is feasible. It means that $\mathsf{Payoff}_i((\sigma_{-i}, \tau)) = 1$, thus τ is a rational deviation from σ. ☐

A.4 Proof of Proposition 13

Proposition 13. *Let σ be a finite memory profile, and i be a player such that* $\mathsf{Payoff}_i(\sigma) = 0$*. We can check whether i has a rational deviation in* PSPACE*.*

Proof. Thanks to Lemma 12 we know that the existence of a winning strategy in $G[\sigma_{-i}]$ is a necessary and sufficient condition for the existence of rational deviation for player i. Therefore, we only need to explain how to check the existence of a such a strategy in PSPACE. A careful analysis of the proof in [14] shows that the existence of a winning strategy in a one-player EnergyBuchi amounts to finding a very specific pattern. Namely, one has to find a reachable cycle C that is either *i*) positive, and from a state in C one can start a new cycle that contains a state in F, or *ii*) the cost of cycle C is nonnegative and it contains a state in F. In both cases, it is nothing but checking the reachability in a finite graph while keeping track of the accumulated cost. We have already seen how to perform all those steps in PSPACE by taking advantage of the fact the reachability problem is NLOGSPACE and that a memory of size $O(n^2 \log(|\sigma|C))$ is needed to keep track of cost's accumulation. ☐

B Proofs of Section 4

B.1 Proof of Theorem 16

Theorem 16. *The RC problem and the RE problem are* PSPACE*-complete.*

Proof. Let an EBG $\mathcal{B}^{c,e} = (N, \Sigma, \Phi, c, e)$ and a finite memory strategy profile σ.

To solve RC, by Remark 14, we can guess a resource redistribution e' and check whether $\sigma \in \mathsf{NE}(\mathcal{B}^{c,e'})$. To solve RE, we can guess a resource redistribution e' and check whether $\sigma \notin \mathsf{NE}(\mathcal{B}^{c,e'})$.

By Theorem 8, $\sigma \in \mathsf{NE}(\mathcal{B}^{c,e'})$ is a PSPACE predicate. Since PSPACE is closed under complement, $\sigma \notin \mathsf{NE}(\mathcal{B}^{c,e'})$ is also a PSPACE predicate. Furthermore, by Savitch's theorem $\mathsf{NPSPACE} = \mathsf{PSPACE}$. So both non-deterministic procedures outlined before indicate the existence of deterministic algorithms to solve RC and RE with polynomial space complexity.

To establish a lower bound, it suffices to remark that the NEM problem is PSPACE-hard even for one-player games. In a one-player game $\mathcal{B}^{c,e}$, there is only one resource redistribution which is e. In this case, σ can be rationally constructed iff $\sigma \in \mathsf{NE}(\mathcal{B}^{c,e})$ iff σ cannot be rationally eliminated. $\square$

B.2 Proof of Proposition 17

Proposition 17. *Let an endowment e be given. The endowment e^i is the resource redistribution of e such that all resources are allocated to player i. The strategy profile σ is eliminable in $\mathcal{B}^{c,e}$ iff for some player i, $\sigma \notin \mathsf{NE}(\mathcal{B}^{c,e^i})$.*

Proof. Right-to-left is immediate. Now, assume σ is eliminable in $\mathcal{B}^{c,e}$. So there is a resource redistribution e' such that $\sigma \notin \mathsf{NE}(\mathcal{B}^{c,e'})$. It means that there is $i \in N$ and a strategy τ_i s.t. $\mathsf{Payoff}_i((\tau_i, \sigma_{-i})) > \mathsf{Payoff}_i(\sigma)$. Observe that necessarily, (τ_i, σ_{-i}) is feasible in $\mathcal{B}^{c,e}$, and thus that player i has enough resources to execute τ_i with an endowment of $e(i)$.

Now consider the game $\mathcal{B}^{c,e^i}$. If σ is not feasible in $\mathcal{B}^{c,e^i}$ then $\sigma \notin \mathsf{NE}(\mathcal{B}^{c,e^i})$ and σ is eliminable. If on the other hand, σ is feasible in $\mathcal{B}^{c,e^i}$, each player $j \neq i$ can execute their σ_j with an endowment $e^i(j) = 0$. Moreover, since player i had enough resources to execute τ_i with an endowment of $e(i)$, she still can execute τ_i with an endowment of $e^i(i) \geq e(i)$. Hence, (τ_i, σ_{-i}) is feasible in $\mathcal{B}^{c,e^i}$. So it is still the case that $\mathsf{Payoff}_i((\tau_i, \sigma_{-i})) > \mathsf{Payoff}_i(\sigma)$ in the game $\mathcal{B}^{c,e^i}$. Again, $\sigma \notin \mathsf{NE}(\mathcal{B}^{c,e^i})$ and σ is eliminable. $\square$