Nash Equilibria in Concurrent Games with Lexicographic Preferences

Julian Gutierrez[1], **Aniello Murano**[2], **Giuseppe Perelli**[1], **Sasha Rubin**[2], **Michael Wooldridge**[1]
[1] Department of Computer Science, Oxford University, Oxford, UK
[2] Università degli Studi di Napoli Federico II, Naples, Italy

Abstract

We study concurrent games with finite-memory strategies where players are given a Büchi and a mean-payoff objective, which are related by a lexicographic order: a player first prefers to satisfy its Büchi objective, and then prefers to minimise costs, which are given by a mean-payoff function. In particular, we show that deciding the existence of a strict Nash equilibrium in such games is decidable, even if players' deviations are implemented as infinite memory strategies.

1 Introduction

The last twenty years has seen considerable research directed at the use of game theoretic techniques in the analysis and verification of multi-agent systems. From this standpoint, processes in a multi-agent systems can be understood as players in a game, acting strategically and rationally in pursuit of delegated preferences; possible behaviours of agents correspond to the strategies of players. One important strand of work in this tradition is the development of techniques for reasoning about what properties players (or coalitions of players) can bring about (i.e., whether they have "winning strategies" for certain conditions) [Alur *et al.*, 2002]. More recently, attention has shifted to the analysis of the equilibrium properties of such systems. A typical question is whether a temporal property holds under the assumption that players select strategies that form a Nash equilibrium. In this work, a key question is how the preferences of agents are represented, and one widely-adopted answer to this question is to associate with each player a qualitative goal/objective, usually given either by a temporal logic property or by a winning (acceptance) condition, such as reachability, safety, Büchi, LTL, etc [Pnueli and Rosner, 1989; Gutierrez *et al.*, 2015; Bouyer *et al.*, 2015]. This setting fits very naturally with the verification of discrete (multi-agent) systems. However, the preference structures that are induced in this way have a rather simple (dichotomous) structure: a player is simply either satisfied or unsatisfied; no distinction is made between outcomes that satisfy the player's objective, nor is any made between outcomes that do not satisfy the objective. This limits the applicability of such representations for modelling many situations of interest. Another alternative setting is given by games where instead of having a qualitative objective, players have quantitative goals—for instance, to minimize a given cost, or to maximise some reward [Ehrenfeucht and Mycielski, 1979; Ummels and Wojtczak, 2011; Kupferman *et al.*, 2016]. A third possibility is to use preference models that *combine* qualitative and quantitative objectives [Chatterjee *et al.*, 2005; Bloem *et al.*, 2009]. To date, most models that combine qualitative and quantitative preferences in game-based multi-agent settings have only been studied in restricted settings (although see [Bulling and Goranko, 2013]).

In this paper, we study this scenario and investigate a combination of quantitative and qualitative objectives not found in the literature. We study goals given by a lexicographic order, where a player first desires to satisfy its qualitative objective (given by a Büchi condition on the states of the system), and secondarily desires to minimise its costs (where costs are given by a quantitative mean-payoff objective). These games are studied in the context of settings with finite memory strategies, while the solution concept under consideration is strict Nash equilibrium. The focus on finite memory strategies is motivated by the fact that, in games with quantitative objectives, finite memory strategies can render decidable settings that would be undecidable otherwise [Ummels and Wojtczak, 2011]. Moreover, the combination of qualitative and quantitative objectives is natural for situations in which agents aim to satisfy some goal while minimising costs. For example, consider a robot whose task is to deliver packages around a factory environment: the primary goal of the robot is to deliver the packages (a qualitative objective), while secondarily minimising fuel consumption (a quantitative objective).

Our contributions are as follows. We show that, in a concurrent game with a lexicographic order of goals given by a Büchi (the primary goal) and a mean-payoff (the secondary goal) condition, deciding the existence of a finite-state strict Nash equilibrium is decidable in NP. Our results also show how to solve the rational synthesis problem [Fisman *et al.*, 2010] and the rational verification problem [Wooldridge *et al.*, 2016]. These two problems relate to establishing which properties (e.g., temporal, ω-regular, etc.) hold in a game, under the assumption that players in the game (i.e., system components) choose strategies in equilibrium.

2 Concurrent Graph-Games

In this section we introduce our game model. We make use of multi-player games played on finite graphs (rather than games in extensive-form or normal-form). Agents move concurrently (which includes the special sequential case), play deterministic (rather than randomised) and finite-state (instead of simply memoryless or infinite-memory) strategies, while trying to maximise their payoffs (equivalently, minimise costs), which are given as a lexicographic combination of a qualitative liveness property (of the form "visit some designated set of states infinitely often") and a quantitative long-term average of the rewards of its actions.

We fix some notation. If X is a set, then X^ω is the set of all infinite sequences over X. If X and Y are sets, then X^Y is the set of all functions $\alpha : Y \to X$. We will often use uppercase Greek letters $\alpha, \beta, \kappa, \cdots$ to name functions. Also, we will use tuple notation: we write $\alpha_y \in X$ instead of $\alpha(y)$.

An *arena* is a tuple

$$A = \langle \mathrm{Ag}, \mathrm{Act}, \mathrm{St}, \iota, \tau \rangle$$

where Ag, Act, and St are finite non-empty sets of *agents* (write $N = |\mathrm{Ag}|$), *actions*, and *states*, respectively; ι is the *initial state*; $\tau : \mathrm{St} \times \mathrm{Act}^{\mathrm{Ag}} \to \mathrm{St}$ is a *transition function* mapping each pair, consisting of a state and an action for each agent, namely a decision $\delta \in \mathrm{Act}^{\mathrm{Ag}}$, to a successor state. A *weighted arena* is a tuple $W = \langle A, \kappa, \beta \rangle$ where A is an arena; $\kappa : \mathrm{Ag} \to (\mathrm{Act} \to \mathbb{Z})$ is a *weight function* associating an integer *weight* to each action of each agent; $\beta : \mathrm{Ag} \to 2^{\mathrm{St}}$ is a *Büchi function* associating a set of *Büchi states* to each agent.

Executions: A *path* $\pi = s_0\delta_0s_1\delta_1 \cdots$ is an infinite sequence over $\mathrm{St} \times \mathrm{Act}^{\mathrm{Ag}}$ such that $\tau(s_i, \delta_i) = s_{i+1}$ for all i. In particular, $\delta_i(a)$ is the action of agent a in step i. An *execution* is a path with $s_0 = \iota$.

Let Exec denote the set of all executions. For each execution $\pi = s_0\delta_0s_1\delta_1 \cdots$ and each agent a, define

1. the sequence $\kappa_a(\pi) = \kappa_a(\delta_0(a))\kappa_a(\delta_1(a)) \cdots$,
2. the set $\mathsf{INF}(\pi) = \{\mathsf{s} \in \mathrm{St} : \exists^\infty \mathsf{m}\ \mathsf{s} = \mathsf{s_m}\}$.

Payoffs: For a sequence $\alpha \in \mathbb{R}^\omega$, let $\mathsf{mp}(\alpha)$ be the *mean-payoff* of α, that is, for every $n \in \mathbb{N}$ define

$$Avg_n(\alpha) = \frac{\Sigma_{j=0}^{n-1} \alpha_j}{n}$$

and then

$$\mathsf{mp}(\alpha) = \lim \inf_{n\to\infty} Avg_n(\alpha).$$

This definition naturally extends to executions, i.e., define $\mathsf{mp}_a(\pi) = \mathsf{mp}(\kappa_a(\pi))$.

For an execution $\pi \in \mathsf{Exec}$ and $B \in 2^{\mathrm{St}}$ define

$$\mathsf{Buchi}_B(\pi) = \begin{cases} \top & \text{if } \mathsf{INF}(\pi) \cap \mathsf{B} \neq \emptyset \\ \bot & \text{otherwise.} \end{cases}$$

Define $\mathsf{Buchi}_a(\pi) = \mathsf{Buchi}_{\beta(a)}$.

Let $\Omega = \{\bot, \top\} \times \mathbb{R}$ denote the set of *payoffs*. We define a total ordering on the set Ω of payoffs: $(x, y) \prec_{lex} (x', y')$ iff, either ($x = \bot$ and $x' = \top$) or ($x = x'$ and $y < y'$).

Lexicographic games: A $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$ *game* is a tuple $G = \langle W, pay \rangle$ where A is a weighted arena and $pay : \mathrm{Ag} \to (\mathsf{Exec} \to \Omega)$ is the *payoff function* defined by $pay_a(\pi) = (\mathsf{Buchi}_a(\pi), \mathsf{mp}_a(\pi))$.

Each agent is trying to maximise its payoff. In other words, agent a's primary goal is to see infinitely some state from the set β_a infinitely often, and its secondary goal is to maximise it's mp-reward (which can be seen as minimising its cost).

Remark 1. *We consider weights as rewards to be maximised. One may, instead, consider them as costs to be minimised. All our results hold for such cost-games. Indeed, given a weighted arena A with weights $(\kappa_a)_{a\in\mathrm{Ag}}$, define a weighted arena A' in which all weights are replaced by their negation. Then agent a that maximises its payoff in A' has the primary goal of making* $\mathsf{Buchi}_a(\cdot)$ *true and the secondary goal of maximising* $\mathsf{mp}(-\kappa_a(\cdot))$. *But maximising* $\mathsf{mp}(-\kappa_a(\cdot)) = -\mathsf{mp}(\kappa_a(\cdot))$ *is the same as minimising* $\mathsf{mp}(\kappa_a(\cdot))$.

Strategies: A *history* is a finite sequence $s_0\delta_0 \cdots s_{n-1}\delta_{n-1}s_n$ such that $s_0 = \iota$ and $\tau(s_i, \delta_i) = s_{i+1}$ for $i < n$. The set of all histories is denoted Hst. A *strategy* for agent $a \in \mathrm{Ag}$ is a function $\mathsf{Hst} \to \mathrm{Act}$. A *strategy profile* is a function $\sigma : \mathrm{Ag} \to (\mathsf{Hst} \to \mathrm{Act})$. A strategy profile σ induces a unique execution π_σ, i.e., the execution $\pi_\sigma = s_0\delta_0s_1\delta_1 \cdots$ such that $\delta_i(a) = \sigma(a)(s_0s_1 \cdots s_i)$ for $i \geq 0$.

Finite-state strategies: A strategy σ is *finite-state* if it is generated by a deterministic finite automaton M with output, i.e., $M = \langle Q, q_0, \Delta, \lambda \rangle$ where Q is a finite set of states, $q_0 \in Q$ is the initial state, $\Delta : Q \times (\mathrm{St} \cup \mathrm{Act}^{\mathrm{Ag}}) \to Q$ is the transition function, and $\lambda : Q \to \mathrm{Act}$ is the output function, so that, on input $h \in \mathsf{Hst}$, the automaton M reaches a state q such that $\lambda(q) = \Delta(h)$. A strategy profile is finite-state if every strategy $\sigma(a)$ is finite-state. Observe that in this case, the unique execution π_σ is ultimately periodic.

Strict ϵ Nash-equilibria: The solution concept we work with is *strict ϵ Nash-equilibrium*. This is a natural refinement of ϵ Nash-equilibrium, and includes strict Nash equilibrium as a special case. For $\epsilon \geq 0$ and $(x, y) \in \Omega$, let $(x, y) + \epsilon$ denote $(x, y + \epsilon) \in \Omega$. A strategy profile σ is a *strict ϵ Nash-equilibrium* if for every agent $a \in \mathrm{Ag}$, and every strategy profile σ' that disagrees with σ on the a-th component (i.e., $\sigma'(a) \neq \sigma(a)$ and $\sigma'(b) = \sigma(b)$ for all $b \neq a$), we have that $pay_a(\pi_{\sigma'}) \prec_{lex} pay_a(\pi_\sigma) + \epsilon$. If $\epsilon = 0$ then we call this a *strict Nash equilibrium*. We remark that an (ordinary) Nash equilibrium uses $\preceq_{lex}$ instead of $\prec_{lex}$. By $\mathsf{FSNE}^\epsilon(G)$ we denote the set of finite-state strict ϵ Nash equilibria in G. We emphasise that, in the definition of a finite-state strict ϵ Nash-equilibrium σ, the deviating strategies $\sigma'(a)$ need not be finite-state. This captures worst-case behaviour of the deviators.

Remark 2. *Consider the function that maps an execution $\pi = s_0\delta_0s_1\delta_1 \cdots$ to $\delta_0\delta_1 \cdots$. Note that, since the transition function is deterministic and there is a unique intial state, this map is a bijection between the set of executions* Exec *and the set of sequences of decisions* $(\mathrm{Act}^{\mathrm{Ag}})^\omega$. *Clearly, this natural mapping applies to histories and finite sequences of decisions too. Thus, in the following, we might refer to histories as sequences of decisions, and to strategies as functions from sequences of decisions to actions. This slight but equivalent way of looking at strategies is a useful technical convenience.*

Decision Problems: The central decision problem of this work, called Rational Synthesis or Verification [Fisman *et al.*, 2010; Gutierrez *et al.*, 2015; Kupferman *et al.*, 2016], asks if there exists a FSNE^ϵ so that the induced play π_σ satisfies a given Büchi condition. Note that in case $B = \mathrm{St}$, this amounts to the deciding the existence of a FSNE^ϵ.

Formally, for a rational $\epsilon \geq 0$ we consider the following decision problems for the class of $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$-games:

1. FSNE^ϵ*-emptiness* is the problem of deciding, given a game G, if $\mathsf{FSNE}^\epsilon(G) \neq \emptyset$.
2. $\mathsf{E}-\mathsf{FSNE}^\epsilon$ is the problem of deciding, given a game G and a set $B \subseteq \mathrm{St}$, if there exists $\sigma \in \mathsf{FSNE}^\epsilon(G)$ such that $\mathsf{INF}(\pi_\sigma) \cap \mathsf{B} \neq \emptyset$.

3 $\mathsf{E}-\mathsf{FSNE}^\epsilon$ and FSNE^ϵ-emptiness are decidable, and in NP

In this section we establish our main technical result, i.e., that $\mathsf{E}-\mathsf{FSNE}^\epsilon$ is in NP. This immediately gives that FSNE^ϵ-emptiness is in NP.

Theorem 1. *There is an* NP *algorithm that, given a rational* $\epsilon \geq 0$*, a* $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$ *game* G*, and a set* $B \subseteq \mathrm{St}$*, decides whether there is* $\sigma \in \mathsf{FSNE}^\epsilon(G)$ *such that* $\mathsf{INF}(\pi_\sigma) \cap \mathsf{B} \neq \emptyset$*.*

We first apply a pre-processing step that pushes the weights from the actions into the states, i.e., we can assume, without loss of generality, that the weight function is of the form $\kappa : \mathrm{Ag} \to (\mathrm{St} \to \mathbb{Z})$. Since this step is standard and of polynomial complexity, we omit it.

We split the remainder of the proof into three steps.

1. We study two-agent zero-sum games with a $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$ objective (played on the same arena as G). We prove that every such game has a value *opt*, this value is computable in NP, and for every $c > 0$ there exists a finite-state strategy for the first agent that guarantees a payoff of at least $opt - c$, and similarly a finite-state strategy for the second agent that guarantees a payoff of at most $opt + c$.
2. We reduce the problem of whether $\mathsf{FSNE}^\epsilon(G) \neq \emptyset$ to the problem of finding thresholds $\overline{z} \in \Omega^{\mathrm{Ag}}$ and an ultimately periodic path in a certain graph $G[\overline{z}]$ such that $z_a \prec_{lex} pay_a(\pi) + \epsilon$. More precisely, each z_a is a so-called "punishing value", i.e., the value of a two-agent zero-sum game with a $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$ objective played on the same arena as G, starting at some state $s \in St$, but with a trying to maximise its payoff and the the rest of the opponents (viewed as a single player) trying to minimise a's payoff. Such values can be computed by the previous step. Moreover, the proof of this characterisation makes use of the approximation result from the previous step.
3. We show how to find ultimately periodic paths π such that $z_a \prec_{lex} pay_a(\pi) + \epsilon$ in graphs of the form $G[\bar{z}]$. We do this in Section 3.3 by adapting the linear programming approach for computing zero-cycles in mean-payoff graphs [Kosaraju and Sullivan, 1988].

In the rest of this section we sketch the technicalities of these steps, and then show how to establish the theorem.

3.1 Two-agent zero-sum $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$-games

We begin with a study of two-agent zero-sum $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$ games $H = \langle A, \kappa, B \rangle$ where A is an arena with $\mathrm{Ag} = \{1, 2\}$, and $\kappa : \mathrm{St} \to \mathbb{Z}$ and $B \subseteq \mathrm{St}$. Agent 1 is called "maximizer" and agent 2 is called "minimizer". Define $pay(\pi) = (\mathsf{Buchi}_B(\pi), \mathsf{mp}(\kappa(\pi)))$. The *value* of H is defined as $val(H) = \sup_\sigma \inf_\delta pay(\pi_{\langle \sigma, \delta \rangle})$ where σ ranges over strategies of the maximizer, δ ranges over strategies of the minimizer, and $\pi_{\langle \sigma, \delta \rangle}$ is the unique execution determined by the profile $\langle \sigma, \delta \rangle$.

Proposition 1. *Every two-agent zero-sum* $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$ *game* H *has a value, denoted* $val(H) \in \Omega$*. Moreover, this value can be computed in* NP*.*

Proof. Without loss of generality, we can consider H to be turn-based (see Remark 3). We compute $val(H)$ by reducing to solving two-agent turn-based zero-sum games J with mean-payoff parity objectives [Chatterjee *et al.*, 2005], which are know to be in NP [Bloem *et al.*, 2009]. The games $J = (W, \mathbb{R}', pay^+)$ are played on state-weighted arenas $W = (A, \kappa, \pi)$ where $\kappa : \mathrm{St} \to \mathbb{Z}$ assigns weights and $\pi : \mathrm{St} \to \mathbb{Z}$ assigns *priorities*, with payoff set $\mathbb{R}' = \{-\infty\} \cup \mathbb{R}$ with its usual ordering $<$, and payoff function $pay^+ : \mathsf{Exec} \to \mathbb{R}'$ defined as follows: $pay^+(\pi)$ equals $-\infty$ if $\mathsf{parity}(\pi) = \bot$, and $\mathsf{mp}(\kappa(\pi))$ otherwise. Here $parity(\pi)$ is defined to be $\top$ if the largest priority occuring infinitely often on π is even, and $\bot$ otherwise. In words, the first player is trying to simultaneously make the parity condition hold and maximise its mean-payoff. The *value* $val(J)$ is defined to be the maximum payoff that the first player can achieve. It follows from [Chatterjee *et al.*, 2005] that values exist and can be computed. Note that the parity condition generalises the Büchi condition (i.e., states in B get priority 2 and all other states get priority 1), and the co-Büchi condition (i.e., states in B get priority 1 and all other states get priority 0).

Since mean-payoff parity games are asymmetric, we introduce their counterparts in which the first player is trying to ensure the largest priority occuring infinitely often is odd (instead of even) and, at the same time, trying to minimise (instead of maximise) the mean-payoff. Formally, these are games $K = (W, \mathbb{R}', pay^-)$ where $pay^-(\pi)$ equals $-\infty$ if $\mathsf{parity}(\pi) = \top$, and $\mathsf{mp}(\kappa(\pi))$ otherwise. The value $val(K)$ is defined to be the minimum payoff that the first player can achieve. Such values exist and can be computed by reducing to mean-payoff parity games: $val(K) = val(K^\partial)$ with $K^\partial = (W^\partial, \mathbb{R}', pay^+)$ and $W^\partial = (A, \kappa^\partial, \pi^\partial)$ with $\kappa^\partial(s) = -\kappa(s)$ and $\pi^\partial(s) = \pi(s) + 1$.

We use these facts to characterise and compute the value of two-agent zero-sum $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$ games as follows. Let $J = (W, \mathbb{R}', pay^+)$ be the mean-payoff Büchi game on the same state-weighted arena as H. If $val(J) \neq -\infty$, then $val(J)$ is the largest mean-payoff value that the first player can enforce while ensuring the Büchi condition holds. In this case $val(H) = (\top, val(J))$. Indeed, if σ_1 ensures $val(J)$ then it also ensures that $(\top, val(J)) \preceq_{lex} val(H)$. On the other hand, if $(\top, val(J)) \prec_{lex} val(H)$ then there is some strategy σ'_1 that ensures the Büchi condition holds and achieves mean-payoff higher than $val(J)$, which contradicts the definition of $val(J)$.

On the other hand, if $val(J) = -\infty$ then the second player can enforce the failure of the Büchi condition. Yet, this strategy may not minimise the mean-payoff as required in the definition of $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$ games. Thus, consider the game $K = (W, \mathbb{R}', pay^-)$ on the same state-weighted arena as H. As $val(J) = -\infty$ we have that $val(K) \neq -\infty$. Thus, $val(H) = (\bot, val(K))$ using similar reasoning as before.

Regarding the complexity, observe that the construction of the games J and K is linear in the size of H, and that we employ an NP procedure once per game. Thus the overall complexity computing $val(H)$ is NP. $\square$

Remark 3. *It is not hard to transform our two-player zero-sum concurrent game H into a two-player zero-sum turn-based H' such that $val(H) = val(H')$.*

To give an intuition, we replace every transition $s \xrightarrow{(c_1,c_2)} s'$ by two transitions $s \xrightarrow{c_1} s_{c_1} \xrightarrow{c_2} s'$, in a way that all the original states belong to Player 1, while every extra state s_c belongs to Player 2, and has the same weight and priority as s (and we do not change the Büchi sets). Note that such a construction depends on the ordering of players, i.e., in order to compute the value for Player 2, we need to employ a construction of a game H'' that replaces $s \xrightarrow{(c_1,c_2)} s'$ by $s \xrightarrow{c_2} s_{c_2} \xrightarrow{c_1} s'$.

It is not hard to see that in two-player zero-sum $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$-games H, a player may need infinite memory to achieve the optimal value $val(H)$ (Cfr. [Chatterjee *et al.*, 2005, Figure 1]). However, as proven in [Bloem *et al.*, 2009], for every mean-payoff parity game J and every $\epsilon > 0$ there exists a finite-memory strategy σ (that depends on ϵ) such that for every strategy δ, it holds that $pay(\pi_{\sigma,\delta}) \geq val(J) - \epsilon$. Thus, using the same argument as in Proposition 1, we get:

Proposition 2. *For every two-agent zero-sum $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$ game H and every $\epsilon > 0$ there exists a finite-state strategy σ for maximizer such that for every strategy δ of the minimizer (not neccessarily finite-state), it holds that $val(H) - \epsilon \preceq_{lex} pay(\pi_{\sigma,\delta})$.*

3.2 Reducing equilibrium finding to path finding

In this section we reduce the problem of the existence of $\sigma \in \mathsf{FSNE}^{\epsilon}(G)$ with $\mathsf{INF}(\pi_\sigma) \cap \mathsf{B} \neq \emptyset$ in a $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$ game G to the existence of an ultimately periodic path in a certain subgraph of the weighted arena of G. To do this, we adapt the proof in [Ummels and Wojtczak, 2011, Section 6] that shows how to decide the existence of a (not neccessarily finite-state) Nash equilibrium for mean-payoff games.

We first need the notion of punishing values and strategies. For $a \in \mathrm{Ag}$ and $s \in \mathrm{St}$ define the *punishing value* $p_a(s)$ to be the $\preceq_{lex}$-largest (x, y) that player a can achieve from state s by "going it alone", i.e., by playing against the coalition $\mathrm{Ag} \setminus \{a\}$. These values can be computed by Proposition 1. Moreover, for every $\epsilon' > 0$, fix $\sigma^{\epsilon'}_{s,a}$ from Proposition 2. We view $\sigma^{\epsilon'}_{s,a}$ as a profile, i.e., $\sigma^{\epsilon'}_{s,a} : \mathrm{Ag} \setminus \{a\} \to (\mathsf{Hst} \to \mathrm{Act})$, and call $\sigma^{\epsilon'}_{s,a}(b)$ a *ϵ'-punishing strategy* for agent b. Note that these ϵ'-punishing strategies are finite-state.

For an agent $a \in \mathrm{Ag}$ and $z \in \Omega$, a pair $(s, \delta) \in \mathrm{St} \times \mathrm{Act}^{\mathrm{Ag}}$ is *z-secure for a* if $p_a(tr(s, \delta')) \preceq_{lex} z$ for every $\delta' \in \mathrm{Act}^{\mathrm{Ag}}$ that agrees with δ except possibly at a.

Proposition 3. *For every $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$ game G, set $B \subseteq \mathrm{St}$, and $\epsilon \geq 0$, the following are equivalent:*

1. *There exists $\sigma \in \mathsf{FSNE}^{\epsilon}(G)$ with $\mathsf{INF}(\pi_\sigma) \cap \mathsf{B} \neq \emptyset$.*
2. *There exists $\bar{z} \in \Omega^{|\mathrm{Ag}|}$ where $z_a \in \{p_a(s) : s \in \mathrm{St}\}$ and there exists an ultimately-periodic execution $\pi = s_0\delta_0 s_1 \delta_1 \cdots$ in G with $\mathsf{INF}(\pi) \cap \mathsf{B} \neq \emptyset$, such that for every agent a,*
   (a) $z_a \prec_{lex} pay_a(\pi) + \epsilon$ and
   (b) for all $i \in \mathbb{N}$, the pair (s_i, δ_i) is z_a-secure for a.

Proof. For (1) implies (2), take $\sigma \in \mathsf{FSNE}^{\epsilon}(G)$ and let $\pi = \pi_\sigma$ be the execution resulting from σ. Since σ is finite-state, π is ultimately periodic. Define $\bar{z} \in \Omega^{|\mathrm{Ag}|}$ by $z_a = \max\{p_a(\delta(s_n, \delta'_n)) : n \in \mathbb{N}, \wedge_{b \neq a} \delta'_n(b) = \delta_n(b)\}$, i.e., z_a is the largest value player a can get by deviating from π. For every $n \in \mathbb{N}$, (s_n, δ_n) is z_a-secure for a (by definition of z_a and z_a-secure). Moreover, $z_a \prec_{lex} pay_a(\pi) + \epsilon$: indeed, let n be such that $z_a = p_a(\delta(s_n, \delta'_n))$, and suppose that $pay_a(\pi) + \epsilon \preceq_{lex} z_a$; then player a would deviate at step n by playing $\delta'_n(a)$ and following a strategy that achieves at least z_a from this point. Note that such a (possibly infinite-state) strategy exists by Proposition 1. But, due to prefix-independence of the payoff function, this is also the payoff of the whole play, contradicting the choice of π as the execution of a strict ϵ Nash-equilibrium.

For (2) implies (1), let $\bar{z} \in \Omega^{|\mathrm{Ag}|}$ and $\pi = s_0\delta_0 s_1 \delta_1 \ldots$ be given with the stated properties. We build a strict ϵ Nash-equilibrium σ such that $\pi_\sigma = \pi$. For $b \in \mathrm{Ag}$, we define $\sigma(b)$ as follows. For every history $h = s_0\delta_0 \ldots s_{n-1}\delta_{n-1}s_n$ (i.e., a prefix of π), define $\sigma_b(h) = \delta_n(b)$, i.e., follow π as long as no-one has deviated from π. For every agent $a \neq b$ and every history of the form $h = s_0\delta_0 \ldots s_{n-2}\delta_{n-2}s_{n-1}\delta'_{n-1}s'_n v$ where δ'_{n-1} equals δ_{n-1} except for the action of a, define $\sigma_b(h) = \sigma^{\epsilon'}_{s,a}(b)(s'_n v)$ where $\epsilon' > 0$ is a constant with the property that $p_a(s'_n) + \epsilon' \prec_{lex} pay_a(\pi) + \epsilon$ (such a constant exists since, by assumptions 2(a) and 2(b), $p_a(s'_n) \preceq_{lex} z_a \prec_{lex} pay_a(\pi) + \epsilon$), i.e., punish a for deviating. For any other history h, define the value $\sigma_b(h)$ to be some arbitrary but fixed action. Note that σ is finite-state since π is ultimately periodic. Finite memory is needed to remember if an agent has deviated and, if so, which agent has deviated. Moreover, $\sigma^{\epsilon'}_{s,a}(b)$ is finite-state.

We show that $\sigma \in \mathsf{FSNE}^{\epsilon}(G)$. Observe that $\pi_\sigma = \pi$, and so $pay_a(\pi_\sigma) = pay_a(\pi)$. Suppose, for a contradiction, that there is an agent a and a strategy profile σ' such that $\sigma'(b) = \sigma(b)$ for $b \neq a$, and (*) $pay_a(\pi) + \epsilon \preceq_{lex} pay_a(\pi')$ where π' is the execution determined by σ'. Let h be the longest common prefix of π and π', and suppose $h\delta s$ is the prefix of π and $h\delta' s'$ is the prefix of π'. Then $\delta_a \neq \delta'_a$. Thus, from the history $h\delta' s'$, each agent $b \neq a$ plays the strategy $\sigma^{\epsilon}_{s,a}(b)$, and thus $pay_a(\pi') \preceq_{lex} p_a(s') + \epsilon'$. Combining this with (*)

we get $pay_a(\pi) + \epsilon \preceq_{lex} p_a(s') + \epsilon'$. By the choice of ϵ', we get $pay_a(\pi) + \epsilon \prec_{lex} pay_a(\pi) + \epsilon$, a contradiction. $\square$

3.3 Path finding in multi-weighted graphs with $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$ payoffs

The following theorem, of interest in its own right, will be used to decide the existence of ultimately periodic paths in Proposition 3. A *multi-weighted graph* is a structure of the form $(V, E, (w_a)_{a\in A}, (\beta_a)_{a\in A})$ where V is a finite set of states, $E \subseteq V^2$ a set of edges, A is a finite index set, $w_a : V \to \mathbb{Z}$ and $\beta_a \subseteq V$.

Theorem 2. *Given a multi-weighted graph $W = (V, E, (w_a)_{a\in A}, (\beta_a)_{a\in A})$, a starting vertex $\iota \in V$, and a vector of payoffs $f \in \Omega^A$, one can decide in polynomial time whether there exists an ultimately periodic $\pi = v_0 v_1 \cdots$ in the graph with $\iota = v_0$ and $f_a \prec_{lex} pay_a(\pi)$ for every $a \in A$.*

Proof. Without loss of generality we may assume that $f_a \in \{\top, \bot\} \times \{0\}$ (to see this, redefine $w_a(s)$ to be $w_a(s) - f_a$). Also, we may assume that every state in V is reachable from ι (to see this, restrict V to the states reachable from ι, computable in linear time). We now reduce the problem to finding certain cycles in W. A *cycle* is a finite path of the form $s_0 s_1 \cdots s_n$ (for some $n \geq 1$) such that $s_0 = s_n$ (note that cycles are not necessarily simple). Write $s \in C$ to mean that $s = s_i$ for some $i \leq n$; similarly, for $B \subseteq V$, write $C \cap B$ for the set of $s \in C$ such that $c \in B$. Define $\mathsf{avg}_a(C) = \frac{1}{n+1}\Sigma_{j=0}^{n} w_a(s_j)$, and $\mathsf{Buchi}_a(C) = \top$ iff $C \cap \beta_a \neq \emptyset$. Given W and $f \in (\{\top, \bot\} \times \{0\})^A$, the stated problem is equivalent to deciding if there exists a cycle C in W such that, for every $a \in A$, $f_a \prec_{lex} \langle \mathsf{Buchi}_a(C), \mathsf{avg}_a(C)\rangle$. However, writing $\mathsf{sum}_a(C) = \Sigma_{j=0}^{n+1} w_a(s_j)$, we can replace $\mathsf{avg}_a(C)$ by $\mathsf{sum}_a(C)$ in this problem (since $\mathsf{avg}_a(C) > 0$ iff $\mathsf{sum}_a(C) > 0$). In order to decide the existence of such a cycle, we adapt the proof from [Kosaraju and Sullivan, 1988] that shows how to decide if there is a cycle C such that for every $a \in A$, $\mathsf{sum}_a(C) = 0$.

A *multicycle* $\mathcal{M}$ is a non-empty multiset of cycles. Thus a cycle is a multicycle $\mathcal{M}$ with $|\mathcal{M}| = 1$. A β-*multicycle* is a multicycle $\mathcal{M}$ such that for every $a \in A$, i) $\mathsf{Buchi}_a(C) = \top$ for some $C \in \mathcal{M}$, and ii) $\Sigma_{C\in\mathcal{M}} \mathsf{sum}_a(C) > 0$. We want to decide if there exists a β-cycle.

Now, one can decide if there exists a β-multicycle in polynomial time using linear programming. For every edge e introduce a variable x_e. Informally, the value x_e is the number of times that the edge e is used on a β-multicycle. Formally, let $\mathsf{src}(e) = \{v \in V : \exists w\, e = (v, w) \in E\}$; $\mathsf{trg}(e) = \{v \in V : \exists w\, e = (w, v) \in E\}$; $\mathsf{out}(v) = \{e \in E : \mathsf{src}(e) = v\}$ and $\mathsf{in}(v) = \{e \in E : \mathsf{trg}(e) = v\}$.

The linear program LP has the following inequalities and equations:

- Eq1: $x_e \geq 0$ for each edge e — this is a basic consistency criterion;
- Eq2: $\Sigma_{e\in E} x_e \geq 1$ — this ensures that at least one edge is chosen;
- Eq3: for each $a \in A$, $\Sigma_{e\in E} \kappa_a(\mathsf{src}(v)) x_e > 0$ — this enforces that the total sum is positive;
- Eq4: for each $a \in A$ such that $f_a = \langle \top, 0\rangle$, $\Sigma_{\mathsf{src}(e)\cap B_a \neq \emptyset} x_e \geq 1$ — this ensures that the Büchi condition is satisfied;
- Eq5: for each $v \in V$, $\Sigma_{e\in \mathsf{out}(v)} x_e = \Sigma_{e\in \mathsf{in}(v)} x_e$ — this "preservation" condition says that the number of times one enters a vertex is equal to the number of times one leaves that vertex.

Clearly every β-multicycle determines an integer solution of the LP. Also, an integer solution of the LP determines a β-multicycle. This is due to Eq5 and Euler's Theorem, i.e., a directed graph is Eulerian if and only if it is connected and for every vertex v, the indegree of v is equal to the outdegree of v [Bang-Jensen and Gutin, 2008, Theorem 1.6.3].

Moreover, the program LP has a solution in the reals iff it has a solution in the rationals [Matousek and Gärtner, 2007]. Moreover, if $(x_e)_{e\in E}$ is a solution to LP and $k \in \mathbb{N}\setminus\{0\}$, then $(kx_e)_{e\in E}$ is also a solution to LP. Thus, the LP has a solution iff it has an integer solution. Thus, the LP has a solution iff the graph has a β-multicycle.

Recall that we want to decide if there is a β-multicycle $\mathcal{M}$ such that $|\mathcal{M}| = 1$, i.e., contains a single cycle. Define a relation on V: $v \equiv w$ iff $v = w$ or there exists a β-multicycle $\mathcal{M}$ and $C \in \mathcal{M}$ such that $v, w \in C$. Note that $\equiv$ is an equivalence relation: indeed, if $u, v \in C$ for $C \in \mathcal{M}$, and $v, w \in C'$ for $C' \in \mathcal{M}'$ then $u, w \in C''$ for $C'' \in \mathcal{M}''$ where C'' is formed by tracing C from w to w and then tracing C' from w to w, and $\mathcal{M}''$ is $\mathcal{M} \cup \mathcal{M}' \setminus \{C, C'\}$.

Suppose $\equiv$ has index 1, i.e., for all $v, w \in V$, $v \equiv w$. We claim that there exists a β-cycle. Indeed: for every $v, v' \in V$ let $\mathcal{M}_{v,v'}$ be a β-multicycle containing a cycle C that visits v and v'. Then $\mathcal{M} = \cup_{v,v'\in V} \mathcal{M}_{v,v'}$ is a β-multicycle such that (*): for every $v, v' \in V$ there exists $C \in \mathcal{M}$ such that $v, v' \in C$. We now define two transformations of multicycles $\mathcal{M} \mapsto \mathcal{M}'$ that maintain the following invariants: a) $\mathsf{sum}(\mathcal{M}) = \mathsf{sum}(\mathcal{M}')$, b) if $\mathcal{M}$ satisfies (*) then so does $\mathcal{M}'$, c) $|\mathcal{M}'| < |\mathcal{M}|$ (i.e., the number of cycles decreases). Thus, repeatedly applying these transformation results in a β-cycle.

First, if C occurs more than once in $\mathcal{M}$, say n times, then remove all occurences of C from $\mathcal{M}$ and add the single C^n formed by tracing C n-many times. Thus, we have that $\mathcal{M}$ is a *set* of cycles (i.e., not a proper multiset). Second, if $\mathcal{M}$ is not a single cycle, take $C \neq C' \in \mathcal{M}$, $v \in C, v' \in C'$ and by (*) pick $D \in \mathcal{M}$ such that $v, v' \in D$. There are three cases: if $D \neq C, D \neq C'$ then form the cycle F by tracing C from v to v, then tracing "half" of D from v to v', then tracing C' from v' to v', and then tracing the "other half" of D from v' to v and let $\mathcal{M}'$ be $\mathcal{M} \cup \{F\} \setminus \{C, C', D\}$; if $D = C$ (the case $D = C'$ is symmetric), then $v' \in C$ and thus form F by tracing C from v' to v' and then tracing C' from v' to v', and let $\mathcal{M}'$ be $\mathcal{M} \cup \{F\} \setminus \{C, C'\}$. Both transformations satisfy the invariants.

Thus, the following algorithm decides if there is a β-cycle: if $|V| = 1$, then output "yes" if $\equiv$ has index 1, and "no" otherwise; else, compute $\equiv$; if it has index 1 then output "yes"; else, for each $\equiv$-class X, recurse on the subgraph induced by X. The algorithm is clearly sound, i.e., if it outputs "yes" then there is indeed a β-cycle. To see that it is complete, note if that C is a β-cycle, then for all $v, w \in C$, $v \equiv w$; and thus C is contained in an $\equiv$-class.

Regarding the complexity, observe that the size of the LP is polynomial in the size of the graph. Moreover, computing the equivalence relation $\equiv$ can be done with $|E|$ linear programs, and hence is also polynomial in the size of the graph (Cfr. [Kosaraju and Sullivan, 1988]). Hence, the overall complexity of the problem is polynomial. $\square$

3.4 Putting the steps together

We can now finish the proof of Theorem 1. Consider a rational $\epsilon \geq 0$ and a $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$ game G with state-weighted arena $W = \langle A, \kappa, \beta \rangle$ where $A = \langle \mathrm{Ag}, \mathrm{Act}, \mathrm{St}, \iota, \tau \rangle$, $\kappa : \mathrm{Ag} \to (\mathrm{St} \to \mathbb{Z})$, and $\beta : \mathrm{Ag} \to 2^{\mathrm{St}}$. For every agent $a \in \mathrm{Ag}$ nondeterministically guess a state $s \in \mathrm{St}$ and, by Proposition 1 compute, in nondeterministic polynomial time, the punishing value $p_a(s)$. Let $z_a = p_a(s)$. Next, we can think of W as a multi-weighted graph $(\mathrm{St}, E, (\kappa_a)_{a \in A}, (\beta_a)_{a \in A})$ where $E(s, s')$ iff $\tau(s, \delta) = s'$ for some δ. For $\overline{z} \in \Omega^{|\mathrm{Ag}|}$, a pair (s, δ) is *$\overline{z}$-secure* if it is z_a-secure for every agent a. Let $G[\overline{z}]$ be the multi-weighted graph formed by restricting W to the set of edges (s, s') for which there exists δ such that $\tau(s, \delta) = s'$ and (s, δ) is $\overline{z}$-secure. Thus, by Proposition 3, to decide if there exists $\sigma \in \mathsf{FSNE}^{\epsilon}(G)$ such that $\mathsf{INF}(\pi_\sigma) \cap \mathsf{B} \neq \emptyset$, we run the deterministic polynomial time decision procedure in Theorem 2 on the multi-weighted graph $(\mathrm{St}, E, (\kappa'_a)_{a \in A'}, (\beta_a)_{a \in A'})$ and threshold $f' \in \Omega^{|\mathrm{Ag}|+1}$ defined as follows: $A' = A \cup \{b\}$ (for a fresh symbol b); $\kappa'_a = \kappa_a$ for all $a \in A$, and $\kappa'_b(s) = 0$ for all $s \in \mathrm{St}$; $\beta'_a = \beta_a$ for all $a \in A$, and $\beta'_b = B$; and $f'(a) = z_a - \epsilon$ for $a \in \mathrm{Ag}$, and $f'(b) = (\top, -1)$. Clearly, the overall procedure can be implemented with a NP algorithm.

4 Related Work

The main decision problem in this work concerns the existence of an equilibrium satisfying a system property in concurrent multiplayer games; this problem is called "rational synthesis" [Fisman *et al.*, 2010] or "rational verification" (cf., E-NASH in [Wooldridge *et al.*, 2016]) and includes NE-emptiness as a special case. We studied this verification problem, specifically, for $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$-games and finite-state strict Nash equilibria.

Most other work in rational synthesis concerns ordinary (not neccessarily finite-state, nor strict) NE-emptiness. In particular, NE-emptiness has been studied for other objectives, notably mean-payoff (NP-complete) [Ummels and Wojtczak, 2011], Büchi (PTIME-complete) [Bouyer *et al.*, 2015], and lexicographic order on Büchi objectives (in NP, but not known to be NP-complete) [Bouyer *et al.*, 2015].

E-NASH for finite-state strategies has been studied on iterated Boolean Games (a simple form of infinite-duration multiplayer concurrent games) as follows: with LTL objectives, E-NASH is 2EXPTIME-complete [Gutierrez *et al.*, 2015]; with objective-LTL, i.e., each agent has to optimise a reward based on the truth value of a finite number of fixed LTL formulas, E-NASH is 2EXPTIME-complete [Kupferman *et al.*, 2016]. A special case of objective-LTL is the lexicographic order on a finite (but unbounded) number of components, each consisting of an LTL formula, also 2EXPTIME-complete. Actually, these lower-bounds are inherited from the fact that solving two-player zero-sum games with LTL objectives is already 2EXPTIME-complete [Rosner, 1991].

We remark that all these works (except objective-LTL) concern equilibria concepts in multiplayer games with either qualitative or quantitative objectives, but not a combination, as we do. Objective-LTL combines Boolean objectives (given as LTL formulas) in a weak way, i.e., there are only finitely many possible payoffs. In contrast, $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$ combines Boolean objectives (given as Büchi sets) with quantitative objectives (mean-payoff), and thus result in infinitely many possible payoffs.

Combinations of qualitative and quantitative objectives have been studied for two-player turn-based games, i.e., in the zero-sum case: mean-payoff parity games [Chatterjee *et al.*, 2005; Bloem *et al.*, 2009], energy parity games [Chatterjee and Doyen, 2012]; and in the non zero-sum case secure-equilibria (in which each player tries to maximise their own payoff and then minimise their opponent's payoff) for a host of quantitative objectives including mean-payoff [Bruyère *et al.*, 2014]. In contrast, our work considers equilibria in multiplayer games.

Finally, we mention logics that combine qualitative and quantitative aspects, i.e., variants of resource-bounded alternating temporal-time logics [Alechina *et al.*, 2015; Bulling and Goranko, 2013]. Although these works deal with multiple-players, they do not deal with Nash equilibria (except in so far as winning strategies form equilibria).

5 Conclusion

In the last twenty years large efforts have been devoted to analyze qualitative and, most recently, quantitative aspects of multi-agent systems. However, these two settings have often been investigated separately. As this is not appropriate in many natural scenarios, researchers have started looking at the combination of these two worlds. The achievements in this direction, however, are far from satisfactory, either because the settings are too weak, e.g., they cannot model important solution concepts such as Nash Equilibria [Bulling and Goranko, 2013], or because they are too expensive in terms of complexity, e.g., between ExpTime-Hard and undecidable [Alechina *et al.*, 2015].

In this paper we introduce a model of multi-agent systems in which each agent's payoff is a lexicographic combination of qualitative (Büchi) and quantitive (mean-payoff) payoffs. We call these $\mathsf{Lex}(\mathsf{Buchi}, \mathsf{mp})$ games. The solution concept we focus on is finite-state strict ϵ Nash equilibria (for $\epsilon \geq 0$). In this setting, we prove the rational-synthesis problem (a generalisation of the equilibrium existence problem) is decidable, and moreover is in NP. The proof characterises the equilibrium executions as certain ultimately periodic paths in a multi-weighted graph. To compute this graph we solve two-player zero-sum games with lexicographic objectives, and to find paths in such graphs we use linear-programming.

References

[Alechina *et al.*, 2015] N. Alechina, B. Logan, N.H. Nga, and F. Raimondi. Model-Checking for Resource-Bounded ATL with Production and Consumption of Resources. *CoRR*, abs/1504.06766, 2015.

[Alur *et al.*, 2002] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-Time Temporal Logic. *Journal of the ACM*, 49(5):672–713, 2002.

[Bang-Jensen and Gutin, 2008] J. Bang-Jensen and G. Gutin. *Digraphs: theory, algorithms and applications*. Springer, 2008.

[Bloem *et al.*, 2009] R. Bloem, K. Chatterjee, T. A. Henzinger, and B. Jobstmann. Better Quality in Synthesis through Quantitative Objectives. In *CAV'09*, pages 140–156, 2009.

[Bouyer *et al.*, 2015] P. Bouyer, R. Brenguier, N. Markey, and M. Ummels. Pure Nash Equilibria in Concurrent Deterministic Games. *Logical Methods in Computer Science*, 11(2), 2015.

[Bruyère *et al.*, 2014] V. Bruyère, N. Meunier, and J.-F. Raskin. Secure equilibria in weighted games. In *CSL-LICS'14*, pages 26:1–26:26, 2014.

[Bulling and Goranko, 2013] N. Bulling and V. Goranko. How to Be Both Rich and Happy: Combining Quantitative and Qualitative Strategic Reasoning about Multi-Player Games (Extended Abstract). In *SR'13*, pages 33–41, 2013.

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

[Chatterjee *et al.*, 2005] K. Chatterjee, T. A. Henzinger, and M. Jurdzinski. Mean-Payoff Parity Games. In *LICS'05*, pages 178–187, 2005.

[Ehrenfeucht and Mycielski, 1979] A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8(2):109–113, 1979.

[Fisman *et al.*, 2010] D. Fisman, O. Kupferman, and Y. Lustig. Rational Synthesis. In *10*, LNCS 6015, pages 190–204, 2010.

[Gutierrez *et al.*, 2015] J. Gutierrez, P. Harrenstein, and M. Wooldridge. Iterated Boolean games. *Inf. Comput.*, 242:53–79, 2015.

[Kosaraju and Sullivan, 1988] S. R. Kosaraju and G. Sullivan. Detecting Cycles in Dynamic Draphs in Polynomial Time. In *STOC'88*, pages 398–406. ACM, 1988.

[Kupferman *et al.*, 2016] O. Kupferman, G. Perelli, and M.Y. Vardi. Synthesis with Rational Environments. *Ann. Math. Artif. Intell.*, 78(1):3–20, 2016.

[Matousek and Gärtner, 2007] J. Matousek and B. Gärtner. *Understanding and Using Linear Programming*. Springer Science & Business Media, 2007.

[Pnueli and Rosner, 1989] A. Pnueli and R. Rosner. On the Synthesis of a Reactive Module. In *Proc. POPL'89*, pages 179–190, 1989.

[Rosner, 1991] R. Rosner. *Modular Synthesis of Reactive Systems*. PhD thesis, Weizmann, 1991.

[Ummels and Wojtczak, 2011] M. Ummels and D. Wojtczak. The Complexity of Nash Equilibria in Limit-Average Games. In *CONCUR'11*, pages 482–496, 2011.

[Wooldridge *et al.*, 2016] M. Wooldridge, J. Gutierrez, P. Harrenstein, E. Marchioni, G. Perelli, and A. Toumi. Rational Verification: From Model Checking to Equilibrium Checking. In *AAAI'16*, pages 4184–4191, 2016.