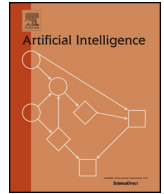




Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artintCooperative concurrent games [☆]Julian Gutierrez ^{a,*}, Szymon Kowara ^c, Sarit Kraus ^b, Thomas Steeples ^c,
Michael Wooldridge ^c^a Faculty of Information Technology, Monash University, Australia^b Department of Computer Science, Bar-Ilan University, Israel^c Department of Computer Science, University of Oxford, United Kingdom

ARTICLE INFO

Article history:

Received 24 October 2021

Received in revised form 24 August 2022

Accepted 11 October 2022

Available online 28 October 2022

Keywords:

Concurrent games

Cooperative games

Multi-agent systems

Logic

Formal verification

ABSTRACT

In *rational verification*, the aim is to verify which temporal logic properties will obtain in a multi-agent system, under the assumption that agents (“players”) in the system choose strategies for acting that form a game theoretic equilibrium. Preferences are typically defined by assuming that agents act in pursuit of individual goals, specified as temporal logic formulae. To date, rational verification has been studied using *non-cooperative* solution concepts—Nash equilibrium and refinements thereof. Such non-cooperative solution concepts assume that there is no possibility of agents forming binding agreements to cooperate, and as such they are restricted in their applicability. In this article, we extend rational verification to *cooperative* solution concepts, as studied in the field of cooperative game theory. We focus on the *core*, as this is the most fundamental (and most widely studied) cooperative solution concept. We begin by presenting a variant of the core that seems well-suited to the concurrent game setting, and we show that this version of the core can be characterised using ATL*. We then study the computational complexity of key decision problems associated with the core, which range from problems in PSPACE to problems in 3EXPTIME. We also investigate conditions that are sufficient to ensure that the core is non-empty, and explore when it is invariant under bisimilarity. We then introduce and study a number of variants of the main definition of the core, leading to the issue of credible deviations, and to stronger notions of collective stable behaviour. Finally, we study cooperative rational verification using an alternative model of preferences, in which players seek to maximise the mean-payoff they obtain over an infinite play in games where quantitative information is allowed.

© 2022 Elsevier B.V. All rights reserved.

1. Introduction

Intelligent agents such as Siri, Alexa, and Cortana are now used by millions of people every day. Originally proposed in the early 1990s, the vision for such agents was that they would be pro-active assistants, working in pursuit of goals delegated to them by their human users. Current intelligent agents make use of an ever-increasing array of AI technologies, starting with natural language understanding in their interface, used to communicate with human users. The next step in

[☆] This paper is an extended and revised version of [1].

* Corresponding author.

E-mail addresses: julian.gutierrez@monash.edu (J. Gutierrez), sarit@cs.biu.ac.il (S. Kraus), thomas.steeple@cs.ox.ac.uk (T. Steeples), michael.wooldridge@cs.ox.ac.uk (M. Wooldridge).

the development of such agents is expected to be the ability of such agents to interact not just with human users but *with each other* in pursuit of their delegated goals. Such multi-agent systems raise a raft of research challenges, which have been the subject of considerable research over the past thirty years. The issue to which we address ourselves in this article is that of understanding the dynamics of such systems. In particular, collections of interacting agents may exhibit undesirable and unpredictable dynamics. The question of *verifying* the possible behaviours of multi-agent systems thus naturally arises.

One increasingly popular approach to the problem of verifying multi-agent systems involves viewing a system as a game (in the sense of game theory), in which agents act rationally and strategically in pursuit of delegated goals. Given this, it is natural to ask what behaviours a system might exhibit under the assumption that agents act rationally—in accordance with game-theoretic solution concepts. This is the key idea that underpins the *rational verification* paradigm [2–5].

Previous work on rational verification has used *concurrent games* as a semantic model of multi-agent systems. Since they were first introduced [6], concurrent games have been very widely adopted in both the AI community and the verification/computer science community [2–4]. A concurrent game [6] is a finite-state environment, populated by a collection of independent, self-interested agents. A game takes place over an infinite sequence of rounds, where at each round, each agent chooses an action to perform. Preferences in concurrent games are typically modelled by assuming that each agent is associated with a temporal logic goal formula [7], which it desires to see satisfied. The infinite plays generated by a game will either satisfy or fail to satisfy each player's goal, and players act rationally in an attempt to achieve their goal. Since the satisfaction of a player's goal may be dependent on the choices made by other players, then players must make choices strategically in order to play optimally.

Now, in all previous studies that we are aware of, concurrent games are assumed to be *non-cooperative*: players act alone, and binding agreements between players are ruled out. The game theoretic solution concepts used in previous studies of concurrent games have therefore been those studied in non-cooperative game theory—primarily Nash equilibrium and refinements thereof. In such a non-cooperative setting, the basic questions that we ask of a concurrent game are, for example, whether a particular temporal logic property holds on *some* computation of the system that could arise through players selecting strategies that form a Nash equilibrium (the E-NASH problem) or whether a property holds on *all* such computations (the A-NASH problem). These problems can be understood as game-theoretic counterparts of the conventional model checking problem [8]: in model checking, we are typically interested in whether a particular property could hold on some or all possible computations of a system, whereas in rational verification, we are interested in whether a property holds on some or all computations *that could arise through rational choices on the part of the players*. The complexity of the corresponding decision problems in concurrent games has been extensively studied, and there now exist a small number of software tools that support rational verification [9–12].

Non-cooperative game theory, however, represents just one branch of game theory. *Cooperative game theory* is a widely studied, albeit less well-known branch of game theory, which distinguishes itself from its non-cooperative counterpart in that it allows for the possibility that *agents can make binding agreements with each other*. The possibility of binding agreements makes it possible for agents to work in teams, and to enjoy the benefits of cooperation. In the real world, we use contracts and other mechanisms (both formal and informal) to enable binding agreements.

The aim of the present paper, therefore, is to extend the study of rational verification to include *cooperative* solution concepts [13–15]. Thus, we assume there is some (exogenous) mechanism through which players in a concurrent game can reach binding agreements and form teams (“coalitions”) in order to collectively achieve goals (although we emphasise that the nature of such a mechanism is beyond the scope of the present work). The possibility of binding cooperation and coalition formation eliminates some undesirable equilibria that arise in non-cooperative settings, and makes available a range of outcomes that cannot be achieved without cooperation. We focus on the *core* as our key solution concept. The basic idea behind the core is that it consists of those strategy profiles from which no subset of players could benefit by collectively deviating. Now, in conventional cooperative games (characteristic function games with transferable utility [15]), this intuition can be given a simple and natural formal definition, and as a consequence, the core is probably the most widely-studied solution concept for cooperative games. However, the conventional definition of the core does not naturally map into our concurrent game setting, because in our games, coalitions are subject to *externalities*: whether a coalition has a beneficial deviation depends not just on the makeup of that coalition, *but also on the behaviour of the remaining players*.

We begin by introducing the framework of concurrent games, and then proceed to define two variations of the core for such settings. In the first, a coalition of players is assumed to have a beneficial deviation if they have some course of action available to them which they would benefit from *no matter what the remaining players did* (cf., the concept of the α -core in the classic game theory literature [16]). However, this “worst case” (maximin) formulation of the core requires a deviation to be beneficial against *all* courses of action by the remaining players—even those that the remaining agents would not rationally choose. This motivates a second definition, where a deviation is only required to be beneficial against all courses of action by remaining players that are *credible*, in the sense that those players would then be no worse off than they were originally. We also consider games where the agents have *quantitative* preferences, modelled as some kind of reward in every round of the game. In each case, we formally define the relevant solution concept, identify some of its key computational properties, give logical characterisations, and where possible, provide complexity results, which range from properties that can be checked in PSPACE to properties that can be checked in 3EXPTIME. We also study model theoretic properties related to the core: in particular, whether it is guaranteed to be non-empty, and whether temporal logic properties hold across bisimilar systems over plays (computation runs) induced by elements in the core of the game (a highly desirable property from a formal verification perspective).

Structure of the paper The remainder of this article is organised as follows:

- In the following section, we summarise the key concepts from logic and concurrent games that are used throughout the paper.
- In Section 3, we define the core and the main computational properties associated with it. In Section 3.4 we study the issue of credible coalition formations, with associated complexity results.
- In Section 4, we study the core in the quantitative setting of concurrent mean-payoff games.
- Concluding remarks and related work are given in Section 5, including a discussion around the implementation of our concepts using model checkers.

2. Preliminaries

2.1. Set and sequences

Given any set S , we use S^* , S^ω , and S^+ for, respectively, the sets of finite, infinite, and non-empty finite sequences of elements in S . If $w_1 \in S^*$ and w_2 is any other (finite or infinite) sequence, we write $w_1 w_2$ for their concatenation. For $Q \subseteq S$, we write S_{-Q} for $S \setminus Q$ and S_{-i} if $Q = \{i\}$. We extend this notation to tuples $u = (s_1, \dots, s_k, \dots, s_n)$ in $S_1 \times \dots \times S_n$, and write u_{-k} for $(s_1, \dots, s_{k-1}, s_{k+1}, \dots, s_n)$, and similarly for sets of elements, that is, by u_{-Q} we mean u without each s_k , for $k \in Q$. Given a sequence w , we write $w[t]$ for the element in position $t + 1$ in the sequence; for instance, $w[0]$ is the first element of w . We also use *slice notation*: we write $w[l \dots m]$ for the sequence $w[l] \dots w[m - 1]$, $w[l \dots]$ for $w[l] w[l + 1] \dots$, and $w[\dots m]$ for $w[0] \dots w[m - 1]$; if $m = 0$, we let $w[l \dots m]$ be the empty sequence, denoted ϵ .

2.2. Games

We begin by introducing the model of multi-agent systems that we use throughout the remainder of the paper: *concurrent game structures* [6]. Informally, a concurrent game consists of a set of players, a set of actions for each of those players, a set of system states, and a transition function which describes how the state of the game changes, given a current state and an action for each of the players.

Formally, a concurrent game structure, M , is given by a tuple,

$$M = (\text{Ag}, \text{St}, \{\text{Ac}_i\}_{i \in \text{Ag}}, s^0, \text{tr}),$$

where:

- Ag and St are finite, non-empty sets of **agents** and **states**, respectively—we usually identify Ag with the set $\{1, \dots, n\}$;
- For each $i \in \text{Ag}$, Ac_i is a finite, non-empty set of **actions** available to agent i . We associate each state s with a set of actions available at that state, $\text{Ac}_i(s) \subseteq \text{Ac}_i$, and we write Ac for $\text{Ac}_1 \times \dots \times \text{Ac}_n$;
- $s^0 \in \text{St}$ is the **initial/start** state; and finally,
- $\text{tr} : \text{St} \times \text{Ac} \rightarrow \text{St}$ is the **transition function** of the game.

As all the other components of the game can be derived from the transition function (the start state can be specified by listing its transitions first, say), the *size* of M is the size of its transition function, given by $|\text{St}| \times |\text{Ac}|^{|\text{Ag}|}$.

Given a concurrent game structure M , we can *play* a game on it as follows: the game starts in state s^0 , and each player $i \in \text{Ag}$ chooses an action available to them, $\text{ac}_i^0 \in \text{Ac}_i(s^0)$. The game then moves to a new state,

$$s^1 = \text{tr}(s^0, \text{ac}_1^0, \dots, \text{ac}_n^0).$$

This process is then repeated. We typically write s^i for the i^{th} state in the sequence, and $\text{ac}^i = (\text{ac}_1^i, \dots, \text{ac}_n^i)$ for the i^{th} vector of actions played in the sequence. Thus for all $t \in \mathbb{N}$, we have

$$s^{t+1} = \text{tr}(s^t, \text{ac}^t)$$

A **run**, ρ is a infinite sequence $\rho = s^0 s^1 s^2 \dots$ such that for every $t \in \mathbb{N}$, there exists some $\text{ac} \in \text{Ac}$ such that $s^{t+1} = \text{tr}(s^t, \text{ac})$. A **path**, π is a finite prefix of a run.

A **strategy** for a player i defines how player i chooses actions at each round of the game, as a function of the previous history of the game. Formally, a strategy is a function $\sigma_i : \text{St}^+ \rightarrow \text{Ac}_i$ such that $\sigma_i(\pi s) \in \text{Ac}_i(s)$ for every $\pi \in \text{St}^*$ and $s \in \text{St}$. Thus, for every path π , a strategy for a player i gives an action available to i from the last state of that path. The set of strategies for player i is denoted by Σ_i .¹

A **strategy profile** $\vec{\sigma}$ is a tuple of strategies, one for each player: $\vec{\sigma} = (\sigma_1, \dots, \sigma_n) \in \Sigma_1 \times \dots \times \Sigma_n$. Observe that a strategy profile, $\vec{\sigma}$, together with a state s , induces a unique run, $\rho(\vec{\sigma}, s)$, where:

¹ Here, we define strategies with respect to finite sequences of states. One can also define them in terms of finite sequences of action profiles. As the sequence of states can be derived from the action profiles, but not the other way around, these strategies are more powerful than the ones we use, and

- $\rho(\vec{\sigma}, s)[0] = s$; and
- $\rho(\vec{\sigma}, s)[t + 1] = \text{tr}(\rho(\vec{\sigma}, s)[t], \sigma_1(\rho(\vec{\sigma}, s)[\dots t]), \dots, \sigma_n(\rho(\vec{\sigma}, s)[\dots t]))$, for all $t \in \mathbb{N}$.

We write $\rho(\vec{\sigma})$ if $s = s^0$.

Note that viewing strategies as functions $\sigma_i: \text{St}^+ \rightarrow \text{Ac}_i$ is problematic with respect to computational analysis, because the domain of such a function is infinite. To be able to answer questions relating to (for example) computational complexity, we need a finite representation for strategies that must operate over an infinite number of rounds. For this purpose, we follow standard practice in the concurrent game's literature, and model strategies as **finite state machines with output** (transducers) [2,3]. We note that, for players with LTL goals, such strategies are sufficient: no more powerful model of strategies is necessary [2,3]. Formally, a strategy for player i is a structure

$$\sigma_i = (Q_i, q_i^0, \delta_i, \tau_i)$$

where:

- Q_i is a finite, non-empty set of **strategy states**;
- $q_i^0 \in Q_i$ is the **initial** strategy state;
- $\delta_i: Q_i \times \text{St} \rightarrow Q_i$ is a **transition function**; and
- $\tau_i: Q_i \rightarrow \text{Ac}_i$ is an **output function**.

A machine strategy works as follows: it begins in the initial state, q_i^0 , and chooses an action based on this, $\tau(q_i^0)$. The state of the game then follows the transition function into a new state. Based on this new (game) state, and the state of the strategy, the strategy then also moves into a new state, based on the strategy transition function. This process repeats, yielding a new action at each timestep. It is easy to see that a strategy profile consisting solely of machine strategies will be eventually periodic (i.e., it will eventually enter a configuration that it was in previously, at which point it will start to repeat its behaviour).

Sometimes, we find we can use an even simpler model of strategies: **memoryless strategies**. A memoryless strategy $\sigma_i: \text{St} \rightarrow \text{Ac}_i$ simply chooses an action based on the current state of the environment. Whilst memoryless strategies are not nearly as expressive as finite-memory strategies, they are still of great importance, owing to their conceptual simplicity, their ease of implementation, and the fact that in many types of game (such as two-player mean-payoff games [18], and parity games [19]), memoryless strategies are sufficient to act optimally.²

2.3. Logics

Broadly, we need to appeal to certain logics for two main reasons: expressing the preferences of agents, and reasoning about those agents and their preferences. For this, we will use three logics throughout this paper: *Linear Temporal Logic* (LTL) [22], *Alternating-Time Temporal Logic* (ATL^{*}) [6], and *Strategy Logic* (SL) [23]. We use LTL for modelling agent preferences, and ATL^{*} and SL for forming logical characterisations of the game-theoretic concepts we will study. Whilst it can be seen more formally from the syntax and semantics presented below, we note here that LTL can be seen as a subset of ATL^{*}, which can be seen as a subset of SL.

2.3.1. Linear temporal logic

LTL is a widely used logic for reasoning about the behaviours of concurrent systems, and while we present the key concepts here, we refer the reader to any standard temporal logic textbook for details (e.g., [24]).

Let AP be a set of propositional variables. Then the syntax of an LTL formula φ is given by the following grammar:

$$\varphi := p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{X}\varphi \mid \varphi\mathbf{U}\varphi,$$

where $p \in \text{AP}$. The set of all LTL formulae over a set of propositional variables AP is denoted $\mathcal{L}(\text{AP})$. If AP is clear from the context, we may instead just write \mathcal{L} . We also introduce the traditional propositional abbreviations, $\cdot \wedge \cdot$, $\cdot \rightarrow \cdot$, $\cdot \leftrightarrow \cdot$, defined in the usual way, as well as the abbreviations $\mathbf{F}\varphi$ for $\top\mathbf{U}\varphi$, and $\mathbf{G}\varphi$ for $\neg\mathbf{F}\neg\varphi$.

Typically, in the context of model checking, the semantics of LTL formulae are defined relative to labelled transition systems [25], or Kripke structures [26], but for our purposes, we define them with respect to the infinite runs generated by concurrent game structures. Formally, let M be a concurrent game structure, and let $\lambda: \text{St} \rightarrow \mathcal{P}(\text{AP})$ be a labelling function, mapping states to sets of propositional variables. Then given an infinite run $\rho \in \text{St}^\omega$ and an LTL formula φ , we say that (M, λ, ρ) *models* φ and write $(M, \lambda, \rho) \models \varphi$ according to the following inductive definition:

indeed, provide desirable properties such as the invariance of Nash equilibria under bisimilarity [17]. However, strategies defined relative to states are standard in the concurrent game structures [6] and rational verification [3] literature, so we use this model for consistency. For more details and further discussion, refer to [17].

² For a characterisation of the types of games where memoryless strategies are sufficient for one or both players to play optimally, refer to [20,21].

- For $p \in \text{AP}$, we have $(M, \lambda, \rho) \models p$ if and only if $p \in \lambda(\rho[0])$;
- For $\varphi \in \mathcal{L}(\text{AP})$, we have $(M, \lambda, \rho) \models \neg\varphi$ if and only if it is not the case that $(M, \lambda, \rho) \models \varphi$;
- For $\varphi, \psi \in \mathcal{L}(\text{AP})$, we have $(M, \lambda, \rho) \models \varphi \wedge \psi$ if and only if we have both $(M, \lambda, \rho) \models \varphi$ and $(M, \lambda, \rho) \models \psi$;
- For $\varphi \in \mathcal{L}(\text{AP})$, we have $(M, \lambda, \rho) \models \mathbf{X}\varphi$ if and only if $(M, \lambda, \rho[1 \dots]) \models \varphi$;
- For $\varphi, \psi \in \mathcal{L}(\text{AP})$, we have $(M, \lambda, \rho) \models \varphi \mathbf{U}\psi$ if and only if there exists some $j \geq 0$ such that $(M, \lambda, \rho[j \dots]) \models \psi$ and for all $i < j$ we have $(M, \lambda, \rho[i \dots]) \models \varphi$.

For notational convenience, we will write $(G, \rho) \models \varphi$ as shorthand for $(M, \lambda, \rho) \models \varphi$, and if G is apparent from the context, we will just write $\rho \models \varphi$.

2.3.2. Alternating-time temporal logic

ATL^{*} is an extension of the branching-time temporal logic CTL^{*} [27] – we shall use this logic in our proofs to reason about coalitions of agents more effectively. The key operator in ATL^{*} is the cooperation modality $\langle\langle C \rangle\rangle\phi$, which asserts that the coalition C has the power to enforce the temporal property ϕ ; more specifically, that there is a collection of strategies for C such that if C follow these strategies, then no matter what other agents do, ϕ is guaranteed to be made true. Given a set of atomic propositions AP and a set of agents Ag, the language of ATL^{*} formulae is defined by the following grammar:

$$\begin{aligned}\varphi &::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\langle C \rangle\rangle\psi \\ \psi &::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\psi \mid \psi \mathbf{U}\psi\end{aligned}$$

where $p \in \text{AP}$ and $C \subseteq \text{Ag}$. We call the formulae produced by φ in the above grammar **ATL^{*} state formulae**, and denote the set that contains them by $\mathcal{L}_s(\text{AP}, \text{Ag})$ and those generated by ψ in the above grammar **ATL^{*} path formulae**, denoted by $\mathcal{L}_p(\text{AP}, \text{Ag})$. (They are given these names as their semantics are defined relative to states and paths respectively.) We emphasise that only ATL^{*} state formulae are well-formed ATL^{*} formulae, and thus, we write $\mathcal{L}(\text{AP}, \text{Ag})$ as shorthand for $\mathcal{L}_s(\text{AP}, \text{Ag})$. When either AP or Ag, or both, are known, we may omit them. With $\text{AP}' \subseteq \text{AP}$, we may write $\varphi|_{\text{AP}'}$ if $\varphi \in \mathcal{L}(\text{AP}', \text{Ag})$ for some set of agents Ag.

In addition to the abbreviations used for LTL formulae as described above, we also use the shorthands $\mathbf{E}\varphi$ for $\langle\langle \text{Ag} \rangle\rangle\varphi$, $\mathbf{A}\varphi$ for $\langle\langle \emptyset \rangle\rangle\varphi$, and $\llbracket C \rrbracket\varphi$ for $\neg\langle\langle C \rangle\rangle\neg\varphi$. Finally, we define the *size* of an ATL^{*} formulae φ as its number of subformulae.

To define the semantics of ATL^{*} formulae, we actually need to define two semantic relations, \models_s (for state formulae) and \models_p (for path formulae). So, let M be some concurrent game structure, along with a labelling function $\lambda : \text{St} \rightarrow \mathcal{P}(\text{AP})$. Then given a state, $s \in \text{St}$ and an ATL^{*} formula φ , we say that (M, λ, s) *models* φ and write $(M, \lambda, s) \models \varphi$ according to the following inductive definition:

- For $\varphi \in \mathcal{L}_s(\text{AP}, \text{Ag})$, we have $(M, \lambda, s) \models \varphi$ if and only if $(M, \lambda, s) \models_s \varphi$;
- For operators that lie in LTL, their inductive semantics are the same as in LTL;
- For $\varphi \in \mathcal{L}_p(\text{AP}, \text{Ag})$, we have $(M, \lambda, s) \models \langle\langle C \rangle\rangle\varphi$ if and only if there is some strategy vector $\vec{\sigma}_C$ for the coalition C , such that for all complementary strategy profiles, $\vec{\sigma}_{\text{Ag} \setminus C}$, it is the case that $(M, \lambda, \rho((\vec{\sigma}_{\text{Ag} \setminus C}, \vec{\sigma}_C), s)) \models_p \varphi$ holds;
- For $\varphi \in \mathcal{L}_s(\text{AP}, \text{Ag})$, we have $(M, \lambda, \rho) \models_p \varphi$ if and only if $(M, \lambda, \rho[0]) \models_s \varphi$;

Where the concurrent game and labelling function are clear from context, we will simply write $s \models \varphi$. Given a concurrent game structure M and a labelling function $\lambda : \text{St} \rightarrow \mathcal{P}(\text{AP})$, we say that φ is **satisfiable** if there exists some state $s \in \text{St}$ such that $(M, \lambda, s) \models \varphi$. Moreover, we say that φ is **equivalent** to ψ if for all states $s \in \text{St}$ we have $(M, \lambda, s) \models \varphi$ if and only if $(M, \lambda, s) \models \psi$.

Note that LTL can be seen as the sublogic of ATL^{*} given by all formulae $\mathbf{A}\varphi$, where the formula φ does not contain the “coalition” quantifiers $\langle\langle C \rangle\rangle$ or $\llbracket C \rrbracket$. Thus ATL^{*} is a particularly effective tool for reasoning about the LTL properties that coalitions can achieve, and this is exactly how we will use it when we come to prove the complexity bounds of our decision problems. Specifically, if we have an LTL game G , we can write $(G, s) \models \varphi$ as shorthand for $(M, \lambda, s) \models \varphi$, and furthermore, if the game G is apparent from the context, we shall drop it and simply write $s \models \varphi$ instead.

The relevant decision problem here is the **model checking problem for ATL^{*}**, which we utilise heavily in the following section:

ATL^{*} MODEL CHECKING:

Given: Concurrent game M , labelling function λ , state $s \in \text{St}$, and ATL^{*} formula φ .

Question: Is it the case that $(M, \lambda, s) \models \varphi$?

This problem is 2EXPTIME-complete [6] for games with two or more players, and PSPACE-complete for one-player games (as then, the problem reduces to LTL model checking) [28].

2.3.3. Strategy logic

Whilst ATL^{*} is a powerful tool for reasoning about coalitions, it famously cannot represent certain game-theoretic concepts, such as the Nash equilibrium [23] – explicit quantification over strategies is required to do this. For our purposes,

we will be able to make most of the necessary logical characterisations using ATL^{*}, but when we come to the *strong core* in Section 3.4, we too will also need the ability to reason explicitly about strategies. SL extends LTL with two **strategy quantifiers**, $\langle\langle x \rangle\rangle$ and $[[x]]$, and an **agent binding** operator (i, x) , where i is an agent and x is a variable. These operators can be read as “*there exists a strategy x* ”, “*for every strategy x* ”, and “*bind agent i to the strategy associated with variable x* ”, respectively. Formally, SL formulae are inductively built from a set of propositions AP, variables Var, and agents Ag, using the following grammar, where $p \in \text{AP}$, $x \in \text{Var}$, and $i \in \text{Ag}$:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{X}\varphi \mid \varphi\mathbf{U}\varphi \mid \langle\langle x \rangle\rangle\varphi \mid [[x]]\varphi \mid (i, x)\varphi.$$

We can now present the semantics of SL. First, denoting the set of all strategies by Str, we define an assignment to be a partial function that maps variables and agents to strategies, $\chi \in \text{Asg} = (\text{Var} \cup \text{Ag}) \rightarrow \text{Str}$. We then use the notation $\chi[i \mapsto f]/\chi[x \mapsto f]$ to refer to the assignment which equals f on i/x , and agrees with χ on every other input on which it is defined. Then, given a concurrent game structure M , for all SL formulae φ , states $s \in \text{St}$ in M , and assignments $\chi \in \text{Asg}$, the relation $M, \chi, s \models \varphi$ is defined as follows:

1. For LTL formulae embedded in SL, their inductive semantics are the same as in LTL;
2. For all formulae φ and variables $x \in \text{Var}$ we have:
 - (a) $M, \chi, s \models \langle\langle x \rangle\rangle\varphi$ if and only if there exists some $f \in \text{Str}$ such that $M, \chi[x \mapsto f], s \models \varphi$;
 - (b) $M, \chi, s \models [[x]]\varphi$ if and only if for all $f \in \text{Str}$ we have $M, \chi[x \mapsto f], s \models \varphi$;
3. For every agent $i \in \text{Ag}$ and variable $x \in \text{Var}$, we have $M, \chi, s \models (i, x)\varphi$ if and only if $M, \chi[i \mapsto \chi(x)], s \models \varphi$.

For a sentence φ , that is, a formula with no free variables and agents [23], we say that M satisfies φ , and write $M \models \varphi$ in that case, if $M, \emptyset, s^0 \models \varphi$, where \emptyset is the empty assignment. We use the following abbreviations: $\langle i \rangle\varphi$ for $\langle\langle x \rangle\rangle(i, x)\varphi$ and $[i]\varphi$ for $[[x]](i, x)\varphi$, which can be intuitively understood as “there is a strategy for agent i such that φ holds” and “ φ holds, for all strategies of agent i ”, respectively. We extend this notation to sets of players and write, for instance, $\langle C \rangle\varphi$ instead of $\langle i \rangle \dots \langle j \rangle\varphi$, where $C = \{i, \dots, j\}$, and similarly for the universal quantifier operator. Then, with $\langle C \rangle\varphi$ we mean that “*coalition C has a joint strategy such that φ holds.*”

2.4. LTL games

We can now define *LTL games* [2,29,30]. The key idea in an LTL game is that each player i is associated with an LTL goal formula γ_i , which it desires to see satisfied. Formally, an LTL game, G , is given by a structure

$$G = (M, \text{AP}, \lambda, (\gamma_i)_{i \in \text{Ag}}),$$

where M is a concurrent game structure, AP is a set of atomic propositions, $\lambda : \text{St} \rightarrow \mathcal{P}(\text{AP})$ is a labelling function, and for each $i \in \text{Ag}$, γ_i is an LTL formula over AP that defines that player’s preference relation over runs.

We now describe how temporal goal formulae γ_i induce **preference relations** \succeq_i over runs. First, given that under a provided run, LTL goals are either satisfied or not, we can identify a set of “winners” and a set of “losers” for that run. Formally, let $\mathcal{W}(\rho)$ denote the set of players that get their goal achieved under ρ , and let $\mathcal{L}(\rho)$ denote the set of players that do not:

$$\begin{aligned} \mathcal{W}(\rho) &= \{i \in \text{Ag} \mid \rho \models \gamma_i\} \\ \mathcal{L}(\rho) &= \text{Ag} \setminus \mathcal{W}(\rho). \end{aligned}$$

We can now use winners and losers to define the preference relations of our agents. Intuitively, agent i will always strictly prefer a run that satisfies its goal γ_i over one that does not, but is indifferent between two runs that satisfy its goal, and is indifferent between two runs that fail to satisfy its goal. Formally, for two runs ρ, ρ' , and a player i , we have $\rho \succeq_i \rho'$ if and only if i is a winner under ρ (note the lack of dependence of ρ') or i is a loser under both ρ and ρ' . Put alternatively, we have $\rho \succeq_i \rho'$ if and only if $(M, \lambda, \rho') \models \gamma_i$ implies that $(M, \lambda, \rho) \models \gamma_i$. Strict preference relations \succ_i are defined in the standard way: $\rho \succ_i \rho'$ if and only if $\rho \succeq_i \rho'$ but not $\rho' \succeq_i \rho$. We leave the reader to verify, first, that this definition matches our informal explanation of preferences above, and second, that the relations \succeq_i so defined are indeed preference relations (i.e., the binary relation \succeq_i is reflexive, complete, and transitive). Finally, we note that we use \mathcal{L} in two different senses in this paper: to denote the languages of various logics, and to denote the set of losers of a run. As these generally appear in different contexts, and take different inputs, there should be no ambiguity as to what interpretation should be taken.

With preference relations defined, we can introduce game-theoretic solution concepts. For now, we will stick with non-cooperative solution concepts. First, a strategy profile $\vec{\sigma}$ is said to be a **Nash equilibrium** if there is no player $i \in \text{Ag}$ and strategy σ'_i for i such that we have $(\vec{\sigma}_{-i}, \sigma'_i) \succ_i \vec{\sigma}$. That is, $\vec{\sigma}$ is a Nash equilibrium if no player can benefit by unilaterally changing its strategy (assuming all other players leave their strategies unchanged). Let $\text{NE}(G)$ denote the set of Nash equilibria of the game G .

We emphasise that Nash equilibrium only considers *unilateral* deviations, i.e., deviations by individual players. Compare this to the notion of a **strong Nash equilibrium** [31,32]: a strategy profile $\vec{\sigma}$ is a strong Nash equilibrium if there is no

coalition C and no strategy σ'_C such that for all $i \in C$ we have $(\vec{\sigma}_{-i}, \sigma'_i) \succ_i \vec{\sigma}$. Thus, strong Nash equilibria are those strategy profiles that are immune to multilateral deviations.

Finally, we mention in passing the notion of the *coalition-proof Nash equilibrium* [33,34]. Whilst we refrain from giving a formal definition here, informally, this can be thought of as capturing the outcomes that would arise when players have unlimited, but non-binding, pre-play communication.

3. Cooperative rational verification

3.1. Defining the core

We want to define counterparts of the rational verification problems E-NASH and A-NASH, as studied in [2,3], but for cooperative settings. For this, we need a version of the core for our concurrent game setting. The core is probably the best-known solution concept in cooperative game theory. Like Nash equilibrium in the non-cooperative setting, the core defines a notion of stability for games, but whereas Nash equilibrium only requires that an outcome is stable in the sense that it admits no *individual* beneficial deviations, the core requires that an outcome admits no beneficial deviations by *coalitions*. In the “standard” model of cooperative games, (cooperative games with transferable utility in characteristic function form [15]), this intuition is easily formalised, but in concurrent games, there is an important difficulty, as follows.

Suppose a coalition of players $C \subseteq \text{Ag}$ is contemplating participating in a strategy profile $\vec{\sigma}$, and in particular, are attempting to determine whether they have a cooperative beneficial deviation from $\vec{\sigma}$. Now, as they consider possible beneficial deviations – collective strategies $\vec{\sigma}_C \in \prod_{i \in C} \Sigma_i$ – *what assumptions should C make about the behaviour of the remaining players $\text{Ag} \setminus C$?* In particular, assuming that the remaining players will not respond, by potentially altering their strategy, is implausible in a cooperative setting.³ Rational players who can cooperate will respond to the deviation rationally and in a cooperative way against the players in C . And, crucially, whether or not C ’s putative deviation is in fact beneficial may well depend upon the behaviour of the remaining players. In game theoretic terms, our concurrent game setting is subject to *externalities*: the performance of the coalition C depends not just on the makeup of the coalition C , but on the behaviour of the remaining players.

It is well-known that cooperative solution concepts are difficult to define in the presence of externalities [15]. In particular, there is no universally accepted definition of the core for games with externalities. Our first definition of the core for concurrent games, therefore, captures *worst case* (maximin) reasoning. Thus, when coalition C is contemplating a deviation, it requires that this deviation will be beneficial *no matter what the remaining players do*. This idea has been explored in the concept of the α -core in cooperative games [16]. To make it formal, we need to formalise the notion of a beneficial deviation. Let $\vec{\sigma}$ be a strategy profile and let C be a coalition; then we say that $\vec{\sigma}'_C$ is a beneficial deviation from $\vec{\sigma}$ if both:

1. $C \subseteq \mathcal{L}(\rho(\vec{\sigma}))$; and
2. For all $\vec{\sigma}'_{-C}$, we have $C \subseteq \mathcal{W}(\rho(\vec{\sigma}'_C, \vec{\sigma}'_{-C}))$.

In other words, $\vec{\sigma}'_C$ is said to be a beneficial deviation from $\vec{\sigma}$ if the players in C would be better off playing $\vec{\sigma}'_C$, rather than their respective strategies in $\vec{\sigma}$, *no matter what strategies the players outside C play*. The core of a game G , denoted $\text{core}(G)$, is then defined to be the set of outcomes of G that no coalition has a beneficial deviation from.

Example 1. Consider the following game, which illustrates how cooperative and non-cooperative solution concepts differ: it contains a poor quality Nash equilibrium that is not in the core. The ability to cooperate makes it possible for agents to avoid the undesirable equilibrium. The game contains two players, $\text{Ag} = \{1, 2\}$ and two variables $\text{AP} = \{p, q\}$, with player 1’s action set being $\text{Ac}_1 = \{pt, pf\}$ and player 2’s action set being $\text{Ac}_2 = \{qt, qf\}$, satisfying that, for every reachable state, if player 1/2 plays pt/qt then p/q will hold, and will not hold if pf/qf is played instead (*i.e.*, player 1 “controls” the value of p and player 2 the value of q). Their goals are identical (and so the game is a coordination game): $\gamma_1 = \gamma_2 = \mathbf{G}(p \wedge q)$. Now, consider the strategy profile $\vec{\sigma}$ in which both players simply fix their respective variables to be false forever (*i.e.*, play pf and qf forever). Neither player will have their goal achieved by such a strategy profile. However, the strategy profile forms a Nash equilibrium, because unilateral deviation cannot improve the situation: neither player has an alternative strategy which would make them better off. In fact, there are infinitely many such poor-quality Nash equilibria in this game, where neither player gets their goal achieved. However, this strategy profile is *not* in the core, because there is a cooperative beneficial deviation to the strategy profile in which both players fix their variables to be true forever (*i.e.*, play pt and qt forever). And, in fact, in every strategy profile which lies in the core, both players get their goal achieved. Thus, using the core instead of Nash equilibrium eliminates poor quality equilibria from the game, leading to socially more desirable outcomes.

Before proceeding, we note that additional variations of the core have been proposed in the game theory literature on cooperative games with externalities, in particular, the β -core and γ -core. The β -core assumes that players outside a deviating coalition will choose strategies to maximally punish the deviating coalition, and that members of a deviating

³ This is the kind of behaviour that one has to assume to define strong Nash equilibrium, a non-cooperative solution concept.

coalition will then collectively best respond to these punishment strategies. Thus, the β -core assumes a *minimax* choice on behalf of the deviating coalition. In contrast, the γ -core assumes that players outside a deviating coalition will form singleton coalitions, with each non-deviator choosing individual strategies in a best response to the deviating coalition strategies. We note that these solution concepts are much less studied than the α -core, and are arguably less natural and less well-motivated. We, therefore, leave them for future study (see also comments in the Related Work Section 5).

3.2. Decision problems

In rational verification [2–4] we are mainly interested in checking which temporal logic properties are satisfied in a given solution concept of a game; typically, in the non-cooperative setting, we study what LTL formulae hold in the Nash equilibria $NE(G)$ of a game G . In the cooperative setting, as introduced here, we are instead interested in what properties hold in the core of the game. The two main decision problems in rational verification are checking whether a temporal logic formula is satisfied by some/every stable strategy profile of the game. For the core, these problems are defined as follows—cf. [2–4].

E-CORE:

Given: Game G , LTL formula φ .

Question: Is it the case that $\exists \vec{\sigma} \in \text{core}(G) : \rho(\vec{\sigma}) \models \varphi$?

A-CORE:

Given: Game G , LTL formula φ .

Question: Is it the case that $\forall \vec{\sigma} \in \text{core}(G) : \rho(\vec{\sigma}) \models \varphi$?

One decision problem considered in the non-cooperative setting is the NON-EMPTYNESS problem, which asks if the set of Nash equilibria of a given game is non-empty. However, as we shall show momentarily, the core of the game is always non-empty, so it does not make sense to consider the corresponding problem in the cooperative setting.

We will also be interested in two additional decision problems: checking whether a given strategy profile is in the core (CORE MEMBERSHIP), and checking whether a given strategy vector for a coalition is a beneficial deviation with respect to a strategy profile (BENEFICIAL DEVIATION). Formally:

CORE MEMBERSHIP:

Given: Game G , strategy profile $\vec{\sigma}$.

Question: Is it the case that $\vec{\sigma} \in \text{core}(G)$?

BENEFICIAL DEVIATION:

Given: Game G , strategy profile $\vec{\sigma}$, coalition C , and deviation $\vec{\sigma}'_C$.

Question: Is $\vec{\sigma}'_C$ a beneficial deviation from $\vec{\sigma}$?

In what follows it is helpful to use the concept of a **fulfilled coalition**: a coalition is fulfilled if they are able to achieve their goals *irrespective of what other players do*; that is, they have a collective strategy that will guarantee their goals are achieved. Formally, we say that a coalition of players C is fulfilled if there is a joint strategy $\vec{\sigma}_C$ for $C \subseteq \text{Ag}$ such that for all joint strategies $\vec{\sigma}_{-C}$ for $\text{Ag} \setminus C$ we have

$$\rho((\vec{\sigma}_C, \vec{\sigma}_{-C})) \models \bigwedge_{i \in C} \gamma_i.$$

Thus, a fulfilled coalition has a winning strategy to collectively achieve their goals. Since we are considering cooperative games, the question is whether such a coalition will form. We have the following:

Lemma 1.

1. There are games G , with strategy profiles $\vec{\sigma} \in \text{core}(G)$, containing fulfilled coalitions $C \subseteq \text{Ag}$ such that $C \not\subseteq \mathcal{W}(\rho(\vec{\sigma}))$;
2. For every game G , strategy profile $\vec{\sigma} \in \text{core}(G)$, and fulfilled coalition C , we have that $C \cap \mathcal{W}(\rho(\vec{\sigma})) \neq \emptyset$;
3. For every game G and fulfilled coalition C , then there is $\vec{\sigma} \in \text{core}(G)$ such that $C \subseteq \mathcal{W}(\rho(\vec{\sigma}))$.

Informally, the first part of the lemma says that the fact that a coalition is fulfilled does not mean that every player in such a coalition is guaranteed to get its goal achieved under an arbitrary member of the core. However, the second part of the lemma says that in any member of the core, some agents of every fulfilled coalition must get their goals achieved. And, the third part of the lemma says that for every fulfilled coalition the core contains a strategy profile in which every player of this coalition gets its goal achieved.

Proof of Lemma 1.1. Consider a game with three states, start, up, down, and three players, $\{1, 2, 3\}$. Let start be the start state of the game, and assume that players 2 and 3 have only one action available to them, and player 1 has two actions, **H**

and **T**. As such, the transition function can be defined solely in terms of player 1's action, and we let $\text{tr}(\text{start}, \mathbf{H}) = \text{up}$ and $\text{tr}(\text{start}, \mathbf{T}) = \text{down}$. The remaining two states are sink states, with $\text{tr}(\text{up}, \mathbf{H}) = \text{tr}(\text{up}, \mathbf{T}) = \text{up}$ and $\text{tr}(\text{down}, \mathbf{H}) = \text{tr}(\text{down}, \mathbf{T}) = \text{down}$. Further suppose we have $\text{AP} = \{p, q\}$, and that $\lambda(\text{start}) = \emptyset$, $\lambda(\text{up}) = \{p\}$ and $\lambda(\text{down}) = \{q\}$. Finally, let the goals of the players be as follows: $\gamma_1 = \top$, $\gamma_2 = \mathbf{XG}p$, and $\gamma_3 = \mathbf{XG}q$. As the game will always end in one of two sink states after the first action, we can identify strategies simply by the first action of player 1: **H** or **T**.

Now, note that the coalition $\{1, 3\}$ is a fulfilled coalition: if player 1 plays **T**, then the game will end up in the state down, and will permanently remain there, satisfying player 1's and player 3's goals. However, note that the strategy profile where player 1 plays **H** is a member of the core: player 1's and player 2's goals are satisfied, and as player 3 only has one action, they cannot deviate to get their goal achieved. Thus, we have a fulfilled coalition and a member of the core, such that the fulfilled coalition is not a subset of the winners of the member of the core. Note the same argument could have been applied the other way around – $\{1, 2\}$ is a fulfilled coalition, and **T** is a member of the core whose winners are not a superset of $\{1, 2\}$. \square

Proof of Lemma 1.2. Let G be a game, $\vec{\sigma}$ be a strategy profile that lies in the core, and C a fulfilled coalition. For the sake of a contradiction, further suppose that the intersection of C and $\mathcal{W}(\rho(\vec{\sigma}))$ is empty. This means that for each $i \in C$, we have $\rho(\vec{\sigma}) \not\models \gamma_i$. But as C is a fulfilled coalition, there exists a strategy $\vec{\sigma}'_C$, such that for all counterstrategies $\vec{\sigma}_{-C}$, we have $\rho((\vec{\sigma}_C, \vec{\sigma}_{-C})) \models \bigwedge_{i \in C} \gamma_i$. Naturally, this implies that $\rho((\vec{\sigma}_C, \vec{\sigma}_{-C})) \models \gamma_i$ for all $i \in C$. But then this implies that $\vec{\sigma}'_C$ is a beneficial deviation for C , contradicting the fact that $\vec{\sigma}$ is a member of the core. Thus, we conclude that C and $\mathcal{W}(\rho(\vec{\sigma}))$ have non-empty intersection. \square

Proof of Lemma 1.3. Let G be a game, and C be a fulfilled coalition. Let $\vec{\sigma}^1 = (\vec{\sigma}_C, \vec{\sigma}_{-C})$ be any strategy profile under which C are fulfilled (that is, every member of C achieves their goal regardless of what the countercoalition does). If $\vec{\sigma}^1$ is a member of the core, then we are done. If not, then there exists some coalition $D \subseteq \text{Ag} \setminus C$ with a beneficial deviation, $\vec{\sigma}'_D$. Then under the strategy $\vec{\sigma}^2 = (\vec{\sigma}_C, \vec{\sigma}'_D, \vec{\sigma}_{-(C \cup D)})$, every player in C gets their goal achieved. As before, if $\vec{\sigma}^2$ is a member of the core, then we are done. If not, we can repeat this process, yielding a sequence of strategies, $\vec{\sigma}^1, \vec{\sigma}^2, \vec{\sigma}^3 \dots$. As the losers of the successive strategy profiles get strictly smaller each time, this process can only continue a finite number of times. By construction, the final member of the sequence will model the goal of every player in C , and will have no beneficial deviations, making it a member of the core. Thus, we see that there exists some strategy profile in the core such that C is a subset of the winners of the strategy profile. \square

In fact, our proof of 1.3 can be modified to give us a stronger result, namely that the core is always non-empty, a highly desirable game-theoretic property, as it ensures the existence of stable strategy profiles for every game, making them rationally implementable in practice. This contrasts with the conventional formulation of the core in transferable utility games [15].

Theorem 1. For every game G , we have $\text{core}(G) \neq \emptyset$.

Proof. Identical to the proof of Lemma 1.3, but instead of setting $\vec{\sigma}^1$ to be a strategy profile under which some coalition can be fulfilled, let it be any arbitrary strategy profile, and continue as before. \square

As fulfilled coalitions can help us understand the coalition formation power in a game, we will also be interested in the following decision problem about coalitions.

FULFILLED COALITION:

Given: Game G , coalition $C \subseteq \text{Ag}$.

Question: Is C a fulfilled coalition of G ?

In the next section, we will investigate these decision problems, as well as some model-theoretic properties of the core.

3.3. Characterising the core

In this section we will study the complexity of the decision problems introduced in the previous section, and will establish some other properties of the core. We have already seen in Theorem 1 that the core is always non-empty, and we shall go on to prove that the satisfaction of an LTL property on some/every outcome in the core is bisimulation-invariant [35]. These two results sharply contrast with the rational verification problem in the non-cooperative setting: in these, the set of Nash equilibria of a game is not guaranteed to always be non-empty [2], nor does bisimulation-invariance hold in the general case [17].

The first of our own decision problems we will consider is FULFILLED COALITION, which we solve in the general case through a logical characterisation using ATL^* .

Theorem 2. FULFILLED COALITION is PSPACE-complete for one-player games, and it is 2EXPTIME-complete for games with more than one player.

Proof. For membership we observe that given a game $G = (M, \gamma_1, \dots, \gamma_n)$ and a coalition $C \subseteq \text{Ag}$, it is the case that C is fulfilled if and only if $s^0 \models \langle\langle C \rangle\rangle \bigwedge_{i \in C} \gamma_i$ holds. By appealing to ATL* MODEL CHECKING, the two upper bounds immediately follow. For the lower bounds, we can reduce the problem of checking for the existence of a winning strategy in a two-player game with LTL goals as defined in [36] for 2EXPTIME-hardness and existential LTL model checking for PSPACE-hardness [28]. Note that, as such, FULFILLED COALITION with more than one player is 2EXPTIME-hard even for two-player zero-sum games, i.e., for games with $\gamma_1 = \neg\gamma_2$. \square

Fulfilled coalitions give an indication of which stable coalitions may form, but are insufficient to characterise the core, and therefore, to check E-CORE and A-CORE properties of a multi-agent system. To do this, we adopt a different strategy and show that these two decision problems are, in general, also 2EXPTIME-complete.

Theorem 3. E-CORE and A-CORE are PSPACE-complete for one-player games and 2EXPTIME-complete for games with more than one player.

Proof. Let us consider E-CORE first. For membership we observe that given a game G and an LTL formula φ , it is the case that $(G, \varphi) \in \text{E-CORE}$ if and only if $s^0 \models \varphi_{\text{E-CORE}}(G, \varphi)$ holds, where $\varphi_{\text{E-CORE}}(G, \varphi)$ is the following ATL* formula:

$$\bigvee_{W \subseteq \text{Ag}} \left(\langle\langle \text{Ag} \rangle\rangle \left(\varphi \wedge \bigwedge_{i \in W} \gamma_i \wedge \bigwedge_{j \in \text{Ag} \setminus W} \neg\gamma_j \right) \wedge \bigwedge_{L \subseteq \text{Ag} \setminus W} \llbracket L \rrbracket \bigvee_{j \in L} \neg\gamma_j \right)$$

which states that there is a path in G that satisfies φ as well as the goals of a set of players W (the “winners”), and that for every subset of players L that do not get their goals achieved in such a path (the “losers”), it is not the case that those players have a beneficial deviation from the path. As before, we appeal to ATL* MODEL CHECKING to obtain the upper bounds; here we need to call a 2EXPTIME algorithm an exponential number of times—one for each coalition of winners, and accept if any single one of them accepts.

For the lower bounds, as for FULFILLED COALITION, we can reduce the problem of checking for the existence of a winning strategy in a two-player game with LTL goals as defined in [36] for 2EXPTIME-hardness and existential LTL model checking for PSPACE-hardness [28]. In fact, note that for one-player games, E-CORE and FULFILLED COALITION are equivalent when $\phi = \gamma_1$, as well as when $\gamma_1 = \phi$, $\gamma_2 = \neg\phi$, and $C = \{1\}$, which are the cases that arise in the two reductions above mentioned.

Finally, for A-CORE, note that $(G, \varphi) \notin \text{A-CORE}$ if and only if $(G, \neg\varphi) \in \text{E-CORE}$: $(G, \varphi) \notin \text{A-CORE}$ means it is not the case that for all members of the core $\vec{\sigma}$ we have $\rho(\vec{\sigma}) \models \varphi$. Pushing the negation through the quantifier, this implies that there exists a member of the core such that $\rho(\vec{\sigma}) \not\models \varphi$, or $\rho(\vec{\sigma}) \models \neg\varphi$. The same argument can then be applied for the reverse direction. Then, since both PSPACE and 2EXPTIME are deterministic complexity classes, we can conclude that A-CORE is PSPACE-complete if $|\text{Ag}| = 1$ and 2EXPTIME-complete if $|\text{Ag}| > 1$, as it is for E-CORE. \square

We now study CORE MEMBERSHIP and BENEFICIAL DEVIATION. For these two problems we first need to define how we will represent strategy profiles, as at present, strategies are defined as infinitary structures, which map finite histories to players’ actions.

Given this finite representation for strategies, we can establish the complexity of CORE MEMBERSHIP and BENEFICIAL DEVIATION.

Theorem 4. CORE MEMBERSHIP is PSPACE-complete for one-player games and 2EXPTIME-complete for games with more than one player.

Proof. For membership, we first compute the winners and losers with respect to $\vec{\sigma} = (\sigma_1, \dots, \sigma_n)$, the outcome of the game. This can be done in PSPACE (it is equivalent to LTL model checking over a “product automata” or “concurrent program” [37]). Once we have computed W , we can check, for every $L \subseteq \text{Ag} \setminus W$, whether L has a beneficial deviation. This is true if and only if L is a fulfilled coalition. Because this can be checked in PSPACE for one-player games and in 2EXPTIME for games with more than one player, the two upper bounds immediately follow. For the lower bounds, we use Lemma 1 and Theorem 2 again. Consider the following game. Let φ be a satisfiable LTL formula and $\vec{\sigma}$ an outcome that does not satisfy φ . Then, $(G, \vec{\sigma}) \in \text{CORE MEMBERSHIP}$ if and only if $(G, \{1\}) \notin \text{FULFILLED COALITION}$, whenever $\gamma_1 = \varphi$ and $\gamma_j = \neg\varphi$, for every player $j \in \text{Ag} \setminus \{1\}$. \square

Let us now consider BENEFICIAL DEVIATION. This is the only “easy” problem for multi-player games: it can be solved in PSPACE. To show this, we again need to find a different proof strategy. Consider any input instance $(G, \vec{\sigma}, \vec{\sigma}'_C)$ of the problem. We observe that, because $\vec{\sigma}'_C$ is fixed, we can make it part of the arena where the game is played, and then check

if players not in C have a joint strategy for $\bigvee_{j \in C} \neg \gamma_j$. Due to the definition of beneficial deviation, we also need to check if $\rho(\vec{\sigma}) \models \bigwedge_{j \in C} \neg \gamma_j$ holds or not.

In other words, the reason why this problem can be solved in PSPACE for multi-player games, unlike all other decision problems we have studied so far (which, in general, can be solved in doubly exponential time), is that this decision problem can be reduced to a one-player game (given by coalition $\text{Ag} \setminus C$) with an LTL goal (given by $\gamma_{\text{Ag} \setminus C} = \bigvee_{j \in C} \neg \gamma_j$) over a “product arena” (denoted by M_C) built from a concurrent game structure M and the joint strategy $\vec{\sigma}'_C$ that we want to check.

Theorem 5. BENEFICIAL DEVIATION is PSPACE-complete, even for one-player games.

Proof. Checking that $\rho(\vec{\sigma}) \models \bigwedge_{j \in C} \neg \gamma_j$ holds can be done in PSPACE. Again, this is equivalent to model checking LTL formulae over a “product automata” or “concurrent program” [37]. If the statement does not hold, then, by definition, $\vec{\sigma}'_C$ is not a beneficial deviation, as at least one player in C already has its goal satisfied by $\vec{\sigma}$. If the statement holds, then we check that $\rho(\vec{\sigma}'_{-C}, \vec{\sigma}'_C) \models \bigwedge_{j \in C} \gamma_j$ holds, for all joint strategies $\vec{\sigma}'_{-C}$ for players not in C . We do this in PSPACE by checking whether it is not the case that $(M_C, \lambda', s^0) \models \bigvee_{j \in C} \neg \gamma_j$ holds, where $M_C = (\text{Ag}', \text{Ac}', \text{St}', s^0, \text{tr}')$ is the concurrent game structure defined as follows:

- $\text{Ag}' = \{0\}$, $\text{Ac}' = \prod_{i \in \text{Ag} \setminus C} \text{Ac}_i$;
- $\text{St}' = \text{St} \times \prod_{j \in C} Q_j$;
- $s^0 = (s^0, q_x^0, \dots, q_y^0)$, such that $\sigma_z = (Q_z, q_z^0, \delta_z, \tau_z)$, $\vec{\sigma}'_C = (\sigma_x, \dots, \sigma_y)$, and $z \in \{x, \dots, y\}$;
- $\text{tr}'((s, q_x, \dots, q_y), (a, \dots, b)) = (s', q'_x, \dots, q'_y)$ such that
 - $s' = \text{tr}(s, \tau(q_x), \dots, \tau(q_y), a, \dots, b)$, and
 - $q'_z = \delta(q_z, s)$, with $z \in \{x, \dots, y\}$;

and λ' is defined as,

$$\lambda'(s, q_x, \dots, q_y) = \lambda(s).$$

In other words, M_C transitions just like M save that it is restricted to the behaviour already defined by $\vec{\sigma}'_C$.

For the lower bound we use LTL model checking. \square

In addition to the above complexity results, we also have a model-theoretic result. Before we can state it, however, we need to define the notion of bisimilarity [35,38]. Let M and M' be two concurrent game structures with the same agents Ag , actions Ac and atomic propositions AP . Moreover, let their respective sets of states and transition functions be denoted by St/St' and tr/tr' respectively. Finally, let λ be a labelling function on St and λ' a labelling function of St' . Then a *bisimulation* between $s^* \in \text{St}$ and $t^* \in \text{St}'$ is a non-empty binary relation, $\sim \subseteq \text{St} \times \text{St}'$, such that,

- We have $s^* \sim t^*$;
- For all $s \in \text{St}$ and for all $t \in \text{St}'$, if $s \sim t$ then $\lambda(s) = \lambda'(t)$;
- For all $s_1, s_2 \in \text{St}$ and for all $t_1 \in \text{St}'$, if $s_1 \sim t_1$ and $\text{tr}(s_1, \text{ac}) = s_2$ for some $\text{ac} \in \text{Ac}$, then $\text{tr}'(t_1, \text{ac}) = t_2$ for some $t_2 \in \text{St}'$ with $s_2 \sim t_2$;
- For all $s_1 \in \text{St}$ and for all $t_1, t_2 \in \text{St}'$, if $s_1 \sim t_1$ and $\text{tr}(t_1, \text{ac}) = t_2$ for some $\text{ac} \in \text{Ac}$, then $\text{tr}'(s_1, \text{ac}) = s_2$ for some $s_2 \in \text{St}$ with $s_2 \sim t_2$.

We say two concurrent game structures are bisimilar if their start states are bisimilar. For further details of bisimilarity over concurrent game structures, refer to [17]. We then say that a property P is bisimulation-invariant when if $M \sim N$, then P holds on M if and only if P holds on N .

We are now in a position to state our result: informally, we have that checking whether an LTL formula is satisfied by some outcome in the core is a bisimulation-invariant property. This result is easy, and follows directly from the membership proof of E-CORE.

Corollary 1. Let $G = (M, \gamma_1, \dots, \gamma_n)$ be a game, φ be an LTL formula, and M' be a concurrent game structure that is bisimilar to M . Then, $(G, \varphi) \in \text{E-CORE}$ if and only if $(G', \varphi) \in \text{E-CORE}$, where $G' = (M', \gamma_1, \dots, \gamma_n)$.

Proof. Follows from the fact that ATL^* is bisimulation-invariant, and that the core can be characterised in ATL^* using $\varphi_{\text{E-CORE}}$, as defined in the proof of Theorem 3. More specifically, it follows from the fact that $(M, \lambda, s^0) \models \varphi_{\text{E-CORE}}(G, \varphi)$ if and only if $(M', \lambda', s^0) \models \varphi_{\text{E-CORE}}(G', \varphi)$. \square

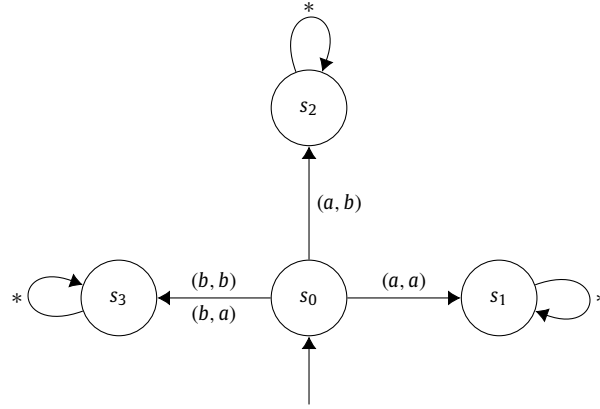


Fig. 1. A game with a non-credible strategy.

3.4. On credible coalition formation: the strong core

As we noted above, our definition of the core assumes worst-case reasoning: a deviation must be beneficial against *all* counter-responses. This definition is robust in the sense that any core-stable outcome is stable in a very strong sense, but one could argue that in some cases it is *too* strong. In particular, when a coalition C is contemplating a deviation $\vec{\sigma}_C$, it can surely assume that the remaining players will not act against their own interests. Thus, one could argue that a deviation need not be beneficial for *all* behaviours of the remaining players, but only those behaviours that are *credible*, in the sense that the remaining players might rationally choose them. To make this discussion concrete, consider the following example.

Example 2. Suppose we have a two-player game G , with a start state, s_0 and three sink states, s_1, s_2 , and s_3 . Each player has two actions available to them, a and b , and the transition function from the start state is defined as follows:

$$\begin{aligned} \text{tr}(s_0, (a, a)) &= s_1, \\ \text{tr}(s_0, (a, b)) &= s_2, \\ \text{tr}(s_0, (b, a)) &= s_3, \\ \text{tr}(s_0, (b, b)) &= s_3 \end{aligned}$$

Thus the game is as illustrated in Fig. 1. Additionally, suppose that the infinite run that ends up in s_1 is the one which satisfies player one's goal, and the run that ends up in s_2 is the one which satisfies player two's goal. Now, in this game, the run which ends up in s_1 lies in the core, but with the use of a non-credible (punishing) strategy by player 1. Notice that the only possible deviation from (a, a) for player 2 is to play b , to which player 1 could respond by also playing b . Although this behaviour would prevent player 2 from achieving its goal, such a way of playing can be regarded as not rational for player 1 given their preference relation: player 1 certainly prefers the run which ends in s_1 over the other two possible runs, but is indifferent otherwise.

Motivated by this phenomenon, we now present a stronger definition for the core. More specifically, with this new definition we require that if a coalition C wants to deviate from a given strategy profile, then the remaining players can only credibly threaten C when they have a counter-response in which both at least one player in C does not get its goal achieved and every winner in the original strategy profile remains a winner in the new one, *i.e.*, the counter-coalition act in accordance with their preference relations. In this solution concept, we are thus capturing the idea of remaining players being willing to punish deviators, but only up to a point: those left behind would prefer not to do worse than they were doing originally. We do not claim that this solution concept is always appropriate, as is the case with all solution concepts in cooperative games with externalities. But it captures another interesting variation of how a non-deviating coalition might respond to a deviation.

We then reformulate the definition of a beneficial deviation, and now say that a deviation $\vec{\sigma}'_C$ is a *strong beneficial deviation* from $\vec{\sigma}$ if the following conditions hold:

1. $C \subseteq \mathcal{L}(\vec{\sigma})$;
2. $C \subseteq \mathcal{W}(\vec{\sigma}_{-C}, \vec{\sigma}'_C)$;
3. For every joint strategy $\vec{\sigma}'_{-C}$ for $\text{Ag} \setminus C$, we have that if $\mathcal{W}(\vec{\sigma}) \subseteq \mathcal{W}(\vec{\sigma}'_{-C}, \vec{\sigma}'_C)$, then $C \subseteq \mathcal{W}(\vec{\sigma}'_{-C}, \vec{\sigma}'_C)$.

With this definition in place we say that the **strong core** of a game, denoted $core^+(G)$, is the set of outcomes of G that admit no strong beneficial deviation. Then, with respect to Example 2, we see that while $\vec{\sigma}_{a_1 a_1}$ is in $core(G)$, it is not the case that $\vec{\sigma}_{a_1 a_1}$ is in $core^+(G)$, since player 2 can now strongly beneficially deviate from $\vec{\sigma}_{a_1 a_1}$ to $\vec{\sigma}_{a_1 b_1}$.

It is worth pausing to reflect for a moment on the issue of formulating the core in the presence of externalities. The game theory literature on this topic is very large. The reason is that the existence of externalities leads to many different definitions of stable behaviour (see, e.g., [13,16,39,40] for many variants of the core). Here, we propose one definition but by no means do we claim it is the only possibility. Essentially, with our definition, we require that for a punishing joint strategy to be credible, winners must remain winners after the presenting the threat.

We will now study the complexity of the decision problems defined in previous sections, but with respect to the strong core. There are four decision problems whose definition depends on the nature of the core: E-CORE, A-CORE, CORE MEMBERSHIP, and BENEFICIAL DEVIATION. We will use the same names for these problems, with the understanding that results in this section are with respect to the strong core. As we will show next, these four problems have the same complexities as with core, but require a more complex logical characterisation, which we provide here using the two-alternation fragment of Strategy Logic (SL) [23].⁴

Theorem 6. *Given a game $G = (M, \gamma_1, \dots, \gamma_n)$ and LTL formula φ , we have $(G, \varphi) \in \text{E-CORE}$ if and only if $M \models \varphi_{\text{E-CORE}}^+(G, \varphi)$, where $\varphi_{\text{E-CORE}}^+(G, \varphi)$ is the SL formula:*

$$\varphi_{\text{E-CORE}}^+(G, \varphi) = \bigvee_{W \subseteq \text{Ag}} \langle \text{Ag} \rangle \left(\varphi \wedge \bigwedge_{i \in W} \gamma_i \wedge \bigwedge_{j \in \text{Ag} \setminus W} \neg \gamma_j \wedge \bigwedge_{C \subseteq \text{Ag} \setminus W} \varphi_{\text{NoBD}}(G, W, C) \right)$$

and the formula $\varphi_{\text{NoSBD}}(G, W, C)$ is defined as follows:

$$\varphi_{\text{NoSBD}}(G, W, C) = [C] \left(\bigwedge_{j \in C} \gamma_j \rightarrow \langle \text{Ag} \setminus C \rangle \left(\bigwedge_{i \in W} \gamma_i \wedge \bigvee_{j \in C} \neg \gamma_j \right) \right)$$

Proof. This SL formula expresses that in the concurrent game structure, there exists a path $\langle \text{Ag} \rangle (\dots)$ under which

1. The formula φ holds;
2. Some players get their goals achieved: $\bigwedge_{i \in W} \gamma_i$;
3. The remaining players do not: $\bigwedge_{j \in \text{Ag} \setminus W} \neg \gamma_j$;
4. No coalition of losers has a strong beneficial deviation: $\bigwedge_{C \subseteq \text{Ag} \setminus W} \varphi_{\text{NoSBD}}(G, W, C)$.

We express the condition of a coalition of losers C not having a strong beneficial deviation with the SL formula $\varphi_{\text{NoSBD}}(G, W, C)$; this is broken down as follows: for every joint strategy of C , if every player in C is better off $(\bigwedge_{j \in C} \gamma_j)$, then the coalition of players outside C has a joint strategy $(\langle \text{Ag} \setminus C \rangle \dots)$ such that both the winners in the original outcome remain winners after the threat is presented $(\bigwedge_{i \in W} \gamma_i)$, and at least one player in the deviating coalition, C , does not get its goal achieved $(\bigvee_{j \in C} \neg \gamma_j)$. \square

At this point, we would like to make a couple of observations. First, that the complexity of checking SL formulae is non-elementary and depends on the alternation-depth of the formula ([23]): SL formulae of alternation-depth n can be checked in $(n + 1)\text{-ExpTIME}$, and in PSPACE for formulae that are semantically equivalent to CTL* formulae. Since $\varphi_{\text{E-CORE}}^+(G, \varphi)$ is an SL formula with two alternations, it can be checked in 3ExpTIME (and in PSPACE if $|\text{Ag}| = 1$). Second, we also would like to recall that finite-state machine strategies, as those we use here, can be characterised in LTL using the technique presented in [2,3]. Using these logical characterisations, we can obtain the following complexity results.

Theorem 7. *For multi-player games, while E-CORE and A-CORE are in 3ExpTIME , CORE MEMBERSHIP is 2ExpTIME -complete and BENEFICIAL DEVIATION is PSPACE-complete. For one-player games, all problems are PSPACE-complete.*

Because we characterised the strong core using SL (which, in contrast to ATL^* , is not a bisimulation-invariant logic), we cannot conclude that the satisfaction of LTL properties by outcomes in the strong core is a bisimulation-invariant property. We believe that this is not the case.

In our first formulation of the core, we saw that the core is always non-empty. Thus, a natural question is to ask whether the strong core is always non-empty. We begin by showing that this is the case for games with three or fewer players:

⁴ We were unable to find a logical characterisation of the strong core using ATL^* . In fact, we believe that such a logical characterisation in ATL^* is not possible for multi-player games.

Theorem 8. For every two-player and three-player game, G , we have $\text{core}^+(G) \neq \emptyset$.

Proof. First, consider games with only two players. For a contradiction, let us suppose that for some game G , the set of outcomes $\text{core}^+(G)$ is empty. This means that no outcome can satisfy both goals, γ_1 and γ_2 . Let us then consider the three remaining possible cases: outcomes that only satisfy γ_1 (case 1), outcomes that only satisfy γ_2 (case 2), and outcomes that satisfy neither γ_1 nor γ_2 (case 3). Let $\vec{f} = (f_1, f_2)$ be an outcome, f'_1 be a deviation by player 1, and f'_2 be a deviation by player 2, and consider the three cases above. In case 1, only player 2 would deviate. Then, outcome (f_1, f'_2) only satisfies γ_2 . Because (f_1, f'_2) is not in the core either, from this outcome only player 1 would deviate, to another outcome (f'_1, f'_2) . Then, outcome (f'_1, f'_2) only satisfies γ_1 . But, then, we have a contradiction, since this means that (f_1, f_2) would be in $\text{core}^+(G)$. We can reason symmetrically to show that case 2 is not possible either. For case 3 we note that only single deviations would be possible. But any such deviations would be to an outcome that either only satisfies γ_1 or only satisfies γ_2 , which are no longer possible. Since no other cases are possible, we have to reject our assumption and conclude that, for two-player games, $\text{core}^+(G)$ is not empty.

For three-player games, the proof is similar, but requires a careful case-by-case analysis. For a full proof, see the appendix. \square

In contrast, for games with four or more players, we find that the strong core may be empty, as the following example illustrates.

Example 3. Consider the following 4 player game, with a start state, s^0 , and six sink states, s^1, \dots, s^6 . Let each player have two actions each, $\{0, 1\}$, and writing $a_1 a_2 a_3 a_4$ for the action (a_1, a_2, a_3, a_4) , our transition function from the start state looks like the following:

$$\begin{array}{ll} \text{tr}(0000) = s^1, & \text{tr}(0001) = s^1, \\ \text{tr}(0010) = s^2, & \text{tr}(0011) = s^2, \\ \text{tr}(0100) = s^1, & \text{tr}(0101) = s^3, \\ \text{tr}(0110) = s^2, & \text{tr}(0111) = s^5, \\ \text{tr}(1000) = s^6, & \text{tr}(1001) = s^4, \\ \text{tr}(1010) = s^4, & \text{tr}(1011) = s^4, \\ \text{tr}(1100) = s^1, & \text{tr}(1101) = s^3, \\ \text{tr}(1110) = s^4, & \text{tr}(1111) = s^3. \end{array}$$

Moreover, suppose player one prefers the runs that end up in the states s^1, s^2, s^3 , player two prefers those that end up in s^1, s^4, s^5 , player three s^2, s^4, s^6 , and player four s^3, s^5, s^6 . The game is illustrated in Fig. 2 – we label states with the players that are winners in that state, and edges with the possible action tuples.⁵ With a careful case-by-case analysis, one can verify that for every state, there exists some coalition with a strong beneficial deviation. Thus, the strong core of the game is empty. For a complete analysis of the deviations that each coalition can make, please refer to the appendix where the full case-by-case analysis is presented.

4. Mean-payoff games

Thus far, we have considered games with *qualitative* preferences—each player has a goal given by some temporal logic formula, which, under a given run, is either satisfied or unsatisfied. But this is a very coarse-grained approach to specifying preferences; it does not offer a way of expressing the *intensity* of the individual player's preferences. One possible way to obtain a richer model of preferences would be to introduce multiple LTL goals for each player, and define some mapping from the set of satisfied formulae to the real numbers [41–43]. However, an alternative approach, which has been widely studied in the literature, is to sidestep temporal logics entirely and assign weights to states, rather than atomic propositions. We then compute the **mean-payoff** of runs, with the idea that agents prefer runs which maximise their mean-payoff [18, 44,45]. In this section, we will revisit the formulation of the core in this mean-payoff setting.

Formally, a mean-payoff game, G , is a tuple,

$$G = (M, \{w_i\}_{i \in \text{Ag}}),$$

⁵ We would like to note that this (counter-)example was automatically generated using bounded exhaustive search of games of different size, which may explain why the game is so counter-intuitive, and most importantly why a game like this one is so hard to find, or be produced, by hand.

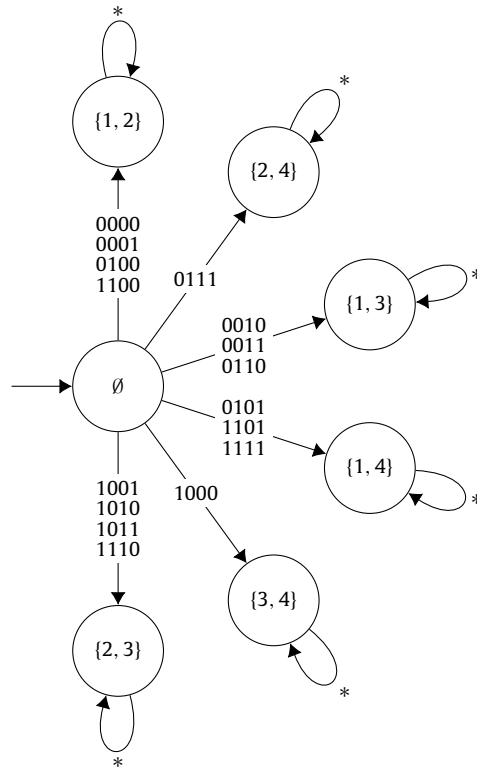


Fig. 2. A game with an empty core*.

where M is a concurrent game structure, and for each $i \in \text{Ag}$, $w_i : \text{St} \rightarrow \mathbb{Z}$ is a **weight** function, mapping states to integers. Games are played in an identical way to the LTL setting, but the agents' preference relations are defined differently here. Let $\beta \in \mathbb{R}^\omega$ be an infinite sequence of real numbers. Then the mean-payoff of β , denoted by $\text{mp}(\beta)$, is defined as follows:

$$\text{mp}(\beta) = \liminf_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \beta_i.$$

In a mean-payoff game, a run $\rho = s^0 s^1 \dots$ induces an infinite sequence of weights for each player, $w_i(s^0) w_i(s^1) \dots$ — we denote this sequence by $w_i(\rho)$ and for notational convenience, we will write $\text{pay}_i(\rho)$ for $\text{mp}(w_i(\rho))$. With this, we can define the preference relation for each player: given two runs, ρ and ρ' , we have $\rho \succeq_i \rho'$ if $\text{pay}_i(\rho) \geq \text{pay}_i(\rho')$; the strict relation \succ_i is defined in the usual way.

Now, recall that our definition of the core in the setting of LTL games relies on the notion of winners and losers of a game. In the mean-payoff setting, it clearly makes no sense to classify players as winners or losers—they can receive a wide spectrum of payoffs. Thus, we need to revisit the concept of a beneficial deviation. In the mean-payoff setting, we say that given a strategy profile $\vec{\sigma}$, a beneficial deviation by a coalition C is a strategy vector $\vec{\sigma}'_C$ such that for all complementary strategy profiles $\vec{\sigma}'_{\text{Ag} \setminus C}$, we have $\rho(\vec{\sigma}'_C, \vec{\sigma}'_{\text{Ag} \setminus C}) \succ_i \rho(\vec{\sigma})$ for all $i \in C$. We then say that $\vec{\sigma}$ is a member of the core if there exists no coalition C which has a beneficial deviation from $\vec{\sigma}$.

4.1. Non-emptiness of the core

We begin by asking whether mean-payoff games always have a non-empty core. Recall that in the LTL games case, we found the core was guaranteed to be non-empty: we find that this does not hold in general for mean-payoff games. Before proving this theorem, we need a small lemma showing that...

Lemma 2. Let $\{\vec{\sigma}^j\}_{j \in \mathbb{N}}$ be a sequence of strategy profiles, and suppose that for some player i , we have $\lim_{j \rightarrow \infty} \text{pay}_i(\rho(\vec{\sigma}^j)) = x$ for some $x \in \mathbb{R}$. Then there exists some strategy profile $\vec{\sigma}^x$ such that $\text{pay}_i(\rho(\vec{\sigma}^x)) = x$.

Proof. First note that the game will end up in a strongly-connected component. So let \mathcal{C} be the set of all simple cycles of the game graph of said strongly-connected component. Consider linear program with solution x — this gives proportion of cycles \square

Theorem 9. *In mean-payoff games, if $|Ag| \leq 2$, then the core is non-empty. For $|Ag| > 2$, there exist games with an empty core.*

Proof. If $|Ag| = 1$, it is straightforward to see that the core is always non-empty; we use Karp’s algorithm for determining the maximum cycle in a weighted graph [46] to determine the maximum payoff that one player can achieve. For two-player games, let $\vec{\sigma} = (\sigma_1, \sigma_2)$ be any strategy profile. If $\vec{\sigma}$ is not in the core, then either Player 1, or Player 2, or the coalition consisting of both players has a beneficial deviation. If the latter is true, then there is a strategy profile, $\vec{\sigma}' = (\sigma'_1, \sigma'_2)$ such that $\vec{\sigma}' \succ_i \vec{\sigma}$ for both $i \in \{1, 2\}$. We repeat this process until the coalition of both players does not have a beneficial deviation. This must eventually be the case as 1) each player’s payoff is capped by their maximum weight and 2) by Theorem 4 of [47], we see that the set of payoffs that a coalition can achieve is a closed set, so any limit point can be attained.⁶ So there must come a point when they cannot beneficially deviate together. At this point, we must either be in the core, or either player 1 or player 2 has a beneficial deviation. If player $j \in \{1, 2\}$ has a beneficial deviation, say σ_j , then any strategy profile (σ_j, σ_i) , with $i \neq j$, that maximises Player i ’s mean-payoff is in the core. Thus, for every two-player game, there exists some strategy profile that lies in the core.

However, for mean-payoff games with three or more players, the core of a game may be empty. The following example illustrates this case.

Example 4. Consider the following three-player game G , where each player has two actions, **H**, **T**, and there are four states, P, R, B, Y . The states are weighted for each player as follows:

$w_i(s)$	1	2	3
P	-1	-1	-1
R	2	1	0
B	0	2	1
Y	1	0	2

If the game is in any state other than P , then no matter what set of actions is taken, the game will remain in that state. Thus, we only specify the transitions for the state P :

Ac	St
(H , H , H)	R
(H , H , T)	R
(H , T , H)	B
(H , T , T)	P
(T , H , H)	P
(T , H , T)	Y
(T , T , H)	B
(T , T , T)	Y

Fig. 3 illustrates the structure of the game.

Note that strategies are characterised by the state that the game eventually ends up in. If the players stay in P forever, then they can all collectively change strategy to move to one of R, B, Y , and each get a better payoff. Now, if the game ends up in R , then players 2 and 3 can deviate by playing (**T**, **H**), and no matter what player 1 plays, the game will be in state B , leaving the two deviating players better off. But similarly, if the game is in B , then players 1 and 3 can deviate by playing (**T**, **T**) to enter state Y , in which they both will be better off, regardless of what player 2 does. And finally, if in Y , then players 1 and 2 can deviate by playing (**H**, **H**) to enter R and will be better off regardless of what player 3 plays. Thus, no strategy profile lies in the core. □

4.2. Decision problems

We now turn our attention to decision problems relating to the core in the mean-payoff setting. However, from a computational perspective, there is an immediate concern here—given a potential beneficial deviation, how can we verify that it is preferable to the status quo under *all* possible counter-responses? Fortunately, as we show in the following lemma, we can restrict our attention to *memoryless strategies* when thinking about potential counter-responses to players’ deviations (see Section 2 for the definition of memoryless strategies):

⁶ The result of [47] actually refers to two-player, multi-mean-payoff games, whilst we are working with multi-player mean-payoff games. We will cover why this is not an issue momentarily.

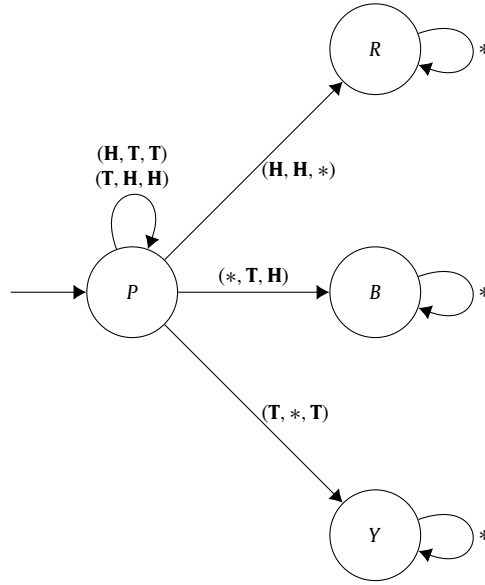


Fig. 3. A game with an empty core.

Lemma 3. Let G be a game, $C \subseteq \text{Ag}$ be a coalition and $\vec{\sigma}$ be a strategy profile. Further suppose that $\vec{\sigma}'_C$ is a strategy vector such that for all memoryless strategy vectors $\vec{\sigma}'_{\text{Ag} \setminus C}$, we have,

$$\rho(\vec{\sigma}'_C, \vec{\sigma}'_{\text{Ag} \setminus C}) \succ_i \rho(\vec{\sigma}).$$

Then, for all strategy vectors, $\vec{\sigma}'_{\text{Ag} \setminus C}$, not necessarily memoryless, we have,

$$\rho(\vec{\sigma}'_C, \vec{\sigma}'_{\text{Ag} \setminus C}) \succ_i \rho(\vec{\sigma}).$$

Before we prove this, we need to introduce an auxiliary concept of *two-player, turn-based, zero-sum, multi-mean-payoff games* [48] (we will simply refer to these as multi-mean-payoff games moving forward). Informally, these are similar to two-player, turn-based, zero-sum mean-payoff games, except player 1 has k weight functions associated with the edges, and they are trying to ensure the resulting k -vector of mean-payoffs is component-wise greater than a vector threshold. Formally, a multi-mean-payoff game is given by a structure

$$G = (V_1, V_2, v^0, E, w, z^k)$$

where V_1, V_2 are sets of states controlled by players 1 and 2 respectively, with $V = V_1 \cup V_2$ the state space, $v^0 \in V$ the start state, $E \subseteq V \times V$ a set of edges, $w : E \rightarrow \mathbb{Z}^k$ a weight function, assigning to each edge a vector of weights, and $z^k \in \mathbb{Q}^k$ is a threshold vector.

The game is played by starting in the start state, $v^0 \in V_i$, and player i choosing an edge (v^0, v^1) , and traversing it to the next state. From this new state, $v^1 \in S_j$, player j chooses an edge and so on, repeating this process forever. Runs are defined in the usual way and the payoff of a run ρ , $\text{pay}(\rho)$, is simply the vector $(\text{mp}(w_1(\rho)), \dots, \text{mp}(w_k(\rho)))$. Player 1 wins if the $\text{pay}_i(\rho) \geq z_i$ for all $i \in \{1, \dots, k\}$, and loses otherwise. The basic question associated with these games is whether player 1 can force a win:

MULTI-MEAN-PAYOFF-THRESHOLD:

Given: Multi-mean-payoff game G .

Question: Is it the case that player 1 has a winning strategy?

As shown in [48], this problem is co-NP-complete. Whilst we do not need to use this complexity result right now, we shall use this fact later. It is also worth noting that in our multi-player mean-payoff games, the weights are attached to states, whilst in multi-mean-payoff games, the weights are attached to edges. For our purposes, this difference is purely superficial—the former can be mapped into the latter simply by pushing the weights onto the outgoing edges, whilst the latter can be mapped into the former by adding more states, one for each edge. If you do this in the correct way, this gives you only a polynomial overhead. As we proceed, we shall use this mapping implicitly (and we in fact have already used it in the proof of Theorem 9).

With this decision problem introduced, we are now in a position to prove Lemma 3.

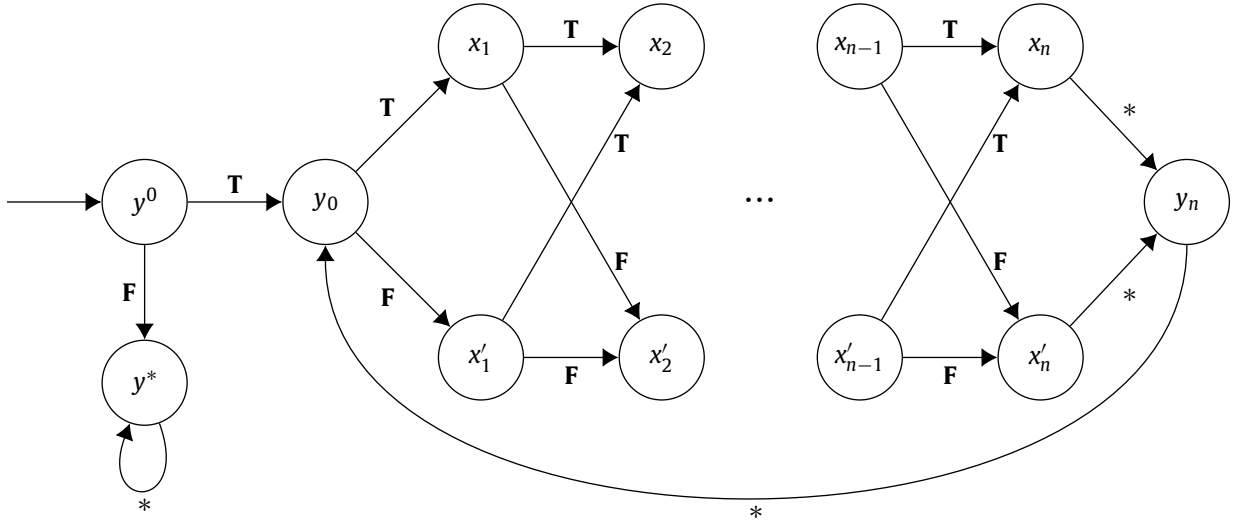


Fig. 4. Concurrent game structure for the reduction from 3SAT.

Proof of Lemma 3. Let $\vec{\sigma}'_{Ag \setminus C}$ be an arbitrary strategy and let $i \in C$ be an arbitrary agent. Suppose it is not the case that $\rho(\vec{\sigma}'_C, \vec{\sigma}'_{Ag \setminus C}) \succ_i \rho(\vec{\sigma})$. Thus, we have $\rho(\vec{\sigma}) \succeq_i \rho(\vec{\sigma}'_C, \vec{\sigma}'_{Ag \setminus C})$. Considering this as a two-player multi-mean-payoff game, where player 1's strategy is fixed and encoded into the game structure (i.e., player 1 follows $\vec{\sigma}'_C$, but has no say in the matter), and the payoff threshold is $mp(\rho(\vec{\sigma}))$, then $\vec{\sigma}'_{Ag \setminus C}$ is a winning strategy for player 2 in this game. Now, by [21,48], if player 2 has a winning strategy, then they have a memoryless winning strategy. Thus, there is a memoryless strategy $\vec{\sigma}''_{Ag \setminus C}$ such that $\rho(\vec{\sigma}) \succeq_i \rho(\vec{\sigma}'_C, \vec{\sigma}''_{Ag \setminus C})$. But this contradicts the assumptions of the lemma, and thus we must have $\rho(\vec{\sigma}'_C, \vec{\sigma}'_{Ag \setminus C}) \succ_i \rho(\vec{\sigma})$. \square

We are now in a position to look at some complexity bounds for mean-payoff games in the cooperative setting. Let us begin by considering the following decision problem relating to beneficial deviations:

BENEFICIAL DEVIATION:

Given: Game G and strategy profile $\vec{\sigma}$.

Question: Does some coalition have a beneficial deviation from $\vec{\sigma}$? That is, does there exist $C \subseteq Ag$ and $\vec{\sigma}'_C \in \Sigma_C$ such that for all $\vec{\sigma}'_{Ag \setminus C} \in \Sigma_{Ag \setminus C}$ and for all $i \in C$, we have:

$$\rho(\vec{\sigma}'_C, \vec{\sigma}'_{Ag \setminus C}) \succ_i \rho(\vec{\sigma})?$$

We have:

Theorem 10. *If the provided strategy profile $\vec{\sigma}$ is memoryless, then BENEFICIAL DEVIATION is NP-complete.*

Proof. First correctly guess a deviating coalition C and a strategy profile $\vec{\sigma}'_C$ for such a coalition of players. Then, use the following three-step algorithm. First, compute the mean-payoffs that players in C get on $\rho(\vec{\sigma})$, that is, a set of values $z_j^* = \text{pay}_j(\rho(\vec{\sigma}))$ for every $j \in C$ – this can be done in polynomial time simply by ‘running’ the strategy profile $\vec{\sigma}$. Then compute the graph $G[\vec{\sigma}'_C]$, which contains all possible behaviours (i.e., strategy profiles) for $Ag \setminus C$ with respect to $\vec{\sigma}$ – this construction is similar to the one used in the proof of Theorem 5, that is, the game when we fix $\vec{\sigma}'_C$, and can be done in polynomial time. Finally, we ask whether every path ρ in $G[\vec{\sigma}'_C]$ satisfies $\text{pay}_j(\rho) > z_j^*$, for every $j \in C$ – for this step, we can use Karp’s algorithm [46] to answer the question in polynomial time for every $j \in C$. If every path in $G[\vec{\sigma}'_C]$ has this property, then we accept; otherwise, we reject.

For hardness, we reduce from 3SAT, using a small variation of the construction in [28]. Let $P = \{x_1, \dots, x_n\}$ be a set of atomic propositions. Given a Boolean formula $\varphi = \bigwedge_{1 \leq c \leq m} C_c$ (in conjunctive normal form) over P – where each $C_c = l_{c1} \vee l_{c2} \vee l_{c3}$, and each literal $l_{ck} = x_j$ or $\neg x_j$, with $1 \leq k \leq 3$, for some $1 \leq j \leq n$ – we construct $M = (Ag, St, s^0, (Ac_i)_{i \in Ag}, tr)$, an m -player concurrent game structure defined as follows, and illustrated in Fig. 4:

- $Ag = \{1, \dots, m\}$;
- $St = \{x_v \mid 1 \leq v \leq n\} \cup \{x'_v \mid 1 \leq v \leq n\} \cup \{y_0, y_n, y^0, y^*\}$;
- $s^0 = y^0$;

- $Ac_i = \{t, f\}$, for every $i \in Ag$, and $Ac = Ac_1 \times \dots \times Ac_m$;
- For tr , refer to the Fig. 4, such that $T = \{t_1, \dots, t_m\}$ and $F = Ac \setminus T$.

With M at hand, we build a mean-payoff game using the following weight function:

- $w_i(x_v) = 1$ if x_v is a literal in C_i and $w_i(x_v) = 0$ otherwise, for all $i \in Ag$ and $1 \leq v \leq n$
- $w_i(x'_v) = 1$ if $\neg x_v$ is a literal in C_i and $w_i(x'_v) = 0$ otherwise, for all $i \in Ag$ and $1 \leq v \leq n$
- $w_i(y_0) = w_i(y_n) = w_i(y^0) = w_i(y^*) = 0$, for all $i \in Ag$

Then, we consider the game G over M and any strategy profile (in memoryless strategies) such that $\vec{\sigma}(s^0) = y^*$. For any of such strategy profiles the mean-payoff of every player is 0. However, if φ is satisfiable, then there is a path in M , from y_0 to y_n , such that in such a path, for every player, there is a state in which its payoff is not 0. Thus, the grand coalition Ag has an incentive to deviate since traversing that path infinitely often will give each player a mean-payoff strictly greater than 0. Observe two things. Firstly, that only if the grand coalition Ag agrees, the game can visit y_0 after y^0 . Otherwise, the game will necessarily end up in y^* forever after. Secondly, because we are considering memoryless strategies, the path from y_0 to y_n followed at the beginning is the same path that will be followed thereafter, infinitely often. Then, we can conclude that there is a beneficial deviation (necessarily for Ag) if and only if φ is satisfiable, as otherwise at least one of the players in the game will not have an incentive to deviate (because its mean-payoff would continue to be 0). We then conclude that $(G, \sigma) \in \text{BENEFICIAL DEVIATION}$ if and only if φ is satisfiable. \square

From Theorem 10 it follows that checking if no coalition of players has a beneficial deviation with respect to a given strategy profile is co-NP complete, and thus we have:

Corollary 2. *If the provided strategy profile $\vec{\sigma}$ is memoryless, then CORE MEMBERSHIP is co-NP-complete.*

We can also leverage BENEFICIAL DEVIATION to obtain the following:

Theorem 11. *If we restrict the existential quantification over strategies to consider only memoryless strategies, then E-CORE is in Σ_2^P .*

Proof. Given a game G , we guess a strategy profile $\vec{\sigma}$ and check that $(G, \vec{\sigma})$ is not an instance of BENEFICIAL DEVIATION. While the former can be done in polynomial time, the latter can be solved in co-NP using an oracle for BENEFICIAL DEVIATION. Thus, we have a procedure that runs in $\text{NP}^{\text{co-NP}} = \Sigma_2^P$. \square

Owing to the alternations present in the definition of the core and beneficial deviations, we believe the E-CORE problem to be complete for Σ_2^P , but we have been unable to find a proof of this. Note that suggests a contrast with the corresponding problem for Nash equilibrium in mean-payoff games, which lies in NP [45]. More importantly, the result also shows that the (complexity) dependence on the type of coalitional deviation is only weak, in the sense that different types of beneficial deviations may be considered within the same complexity class, as long as such deviations can be checked with an NP or co-NP oracle.

4.3. Fulfilled coalitions

We now generalise the idea of a fulfilled coalition. To do this, we introduce the notion of a **lower bound**. Let $C \subseteq Ag$ be a coalition in a game G and let $\vec{z}_C \in \mathbb{Q}^C$. We say that \vec{z}_C is a lower bound for C if there is a joint strategy $\vec{\sigma}_C$ for C such that for all strategies $\vec{\sigma}_{-C}$ for $Ag \setminus C$, we have $\text{pay}_i(\rho(\vec{\sigma}_C, \vec{\sigma}_{-C})) \geq z_i$, for every $i \in C$. Based on this definition, we can prove the following, which characterises the core in terms of lower bounds:

Lemma 4. *Let ρ be a run in G . There is $\vec{\sigma} \in \text{core}(G)$ such that $\rho = \rho(\vec{\sigma})$ if and only if for every coalition $C \subseteq Ag$ and lower bound $\vec{z}_C \in \mathbb{Q}^C$ for C , there is some $i \in C$ such that $z_i \leq \text{pay}_i(\rho)$.*

Proof. To show the left-to-right direction, suppose that there exists a member of the core $\vec{\sigma} \in \text{core}(G)$ and suppose further that there is some coalition $C \subseteq Ag$ and lower bound $\vec{z}_C \in \mathbb{Q}^C$ for C , such that for every $i \in C$ we have $z_i > \text{pay}_i(\rho)$. Because \vec{z}_C is a lower bound for C , and $z_i > \text{pay}_i(\rho)$, for every $i \in C$, then there is a joint strategy $\vec{\sigma}_C$ for C such that for all strategies $\vec{\sigma}_{-C}$ for $Ag \setminus C$, we have $\text{pay}_i(\rho(\vec{\sigma}_C, \vec{\sigma}_{-C})) \geq z_i > \text{pay}_i(\rho)$, for every $i \in C$. Then, it follows that $(G, \vec{\sigma}) \in \text{BENEFICIAL DEVIATION}$, which further implies that $\vec{\sigma}$ cannot be in the core of G – a contradiction to our initial hypothesis.

For the right-to-left direction, suppose that there is ρ in G such that for every coalition $C \subseteq Ag$ and lower bound $\vec{z}_C \in \mathbb{Q}^C$ for C , there is $i \in C$ such that $z_i \leq \text{pay}_i(\rho)$. We then simply let $\vec{\sigma}$ be any strategy profile such that $\rho = \rho(\vec{\sigma})$. Now, let $C = \{j, \dots, k\} \subseteq Ag$ be any coalition and $\vec{\sigma}'_C$ be any possible deviation of C from $\vec{\sigma}$. Either $\vec{z}'_C = (\text{pay}_j(\rho(\vec{\sigma}_{-C}, \vec{\sigma}'_C)), \dots, \text{pay}_k(\rho(\vec{\sigma}_{-C}, \vec{\sigma}'_C)))$ is a lower bound for C or it is not.

If we have the former, by hypothesis, we know that there is $i \in C$ such that $\text{pay}_i(\rho(\vec{\sigma}_{-C}, \vec{\sigma}'_C)) \leq \text{pay}_i(\rho)$. Therefore, i will not have an incentive to deviate along with $C \setminus \{i\}$ from $\vec{\sigma}$, and as a consequence coalition C will not be able to beneficially deviate from $\vec{\sigma}$.

If, on the other hand, \vec{z}'_C is not a lower bound for C , then, by the definition of lower bounds, we know that it is not the case that $\vec{\sigma}'_C$ is a joint strategy for C such that for all strategies $\vec{\sigma}'_{-C}$ for $\text{Ag} \setminus C$, we have $\text{pay}_i(\rho(\vec{\sigma}'_C, \vec{\sigma}'_{-C})) \geq \text{pay}_i(\rho(\vec{\sigma}_{-C}, \vec{\sigma}'_C))$, for every $i \in C$. That is, there exists $i \in C$ and $\vec{\sigma}'_{-C}$ for $\text{Ag} \setminus C$ such that $\text{pay}_i(\rho(\vec{\sigma}'_C, \vec{\sigma}'_{-C})) < \text{pay}_i(\rho(\vec{\sigma}_{-C}, \vec{\sigma}'_C))$. We will now choose $\vec{\sigma}'_{-C}$ so that, in addition, $\text{pay}_i(\rho) \geq \text{pay}_i(\rho(\vec{\sigma}'_C, \vec{\sigma}'_{-C}))$ for some i .

Let $\vec{z}''_C = (\text{pay}_j(\rho(\vec{\sigma}_{-C}^j, \vec{\sigma}'_C)), \dots, \text{pay}_k(\rho(\vec{\sigma}_{-C}^k, \vec{\sigma}'_C)))$ where $\text{pay}_i(\rho(\vec{\sigma}_{-C}^i, \vec{\sigma}'_C))$ is defined to be $\min_{\vec{\sigma}'_{-C} \in \Sigma_{-C}} \text{pay}_i(\rho((\vec{\sigma}'_{-C}, \vec{\sigma}'_C))$). That is, $\vec{\sigma}_{-C}^i$ is a strategy for $\text{Ag} \setminus C$ which ensures the lowest mean-payoff for i assuming that C is playing the joint strategy $\vec{\sigma}'_C$. By construction \vec{z}''_C is a lower bound for C – since each $z''_i = \text{pay}_i(\rho(\vec{\sigma}_{-C}^i, \vec{\sigma}'_C))$ is the greatest mean-payoff value that i can ensure for itself when C is playing $\vec{\sigma}'_C$, no matter what coalition $\text{Ag} \setminus C$ does—and therefore, by hypothesis we know that for some $i \in C$ we have $\text{pay}_i(\rho(\vec{\sigma}_{-C}^i, \vec{\sigma}'_C)) \leq \text{pay}_i(\rho)$. As a consequence, as before, i will not have an incentive to deviate along with $C \setminus \{i\}$ from $\vec{\sigma}$, and therefore coalition C will not be able to beneficially deviate from $\vec{\sigma}$. Because C and $\vec{\sigma}'_C$ where arbitrarily chosen, we conclude that $\vec{\sigma} \in \text{core}(G)$, proving the right-to-left direction and finishing the proof. \square

With this lemma in mind, we want to determine if a given vector, \vec{z}_C , is in fact a lower bound and importantly, how efficiently we can do this. That is, to understand the following decision problem:

LOWER BOUND:

Given: Game G , coalition $C \subseteq \text{Ag}$, and vector $\vec{z}_C \in \mathbb{Q}^{\text{Ag}}$.

Question: Is \vec{z}_C a lower bound for C in G ?

Using the MULTI-MEAN-PAYOFF-THRESHOLD decision problem introduced earlier, we can prove the following theorem:

Theorem 12. LOWER BOUND is co-NP-complete.

Proof. We prove membership as well as hardness by reducing to and from MULTI-MEAN-PAYOFF-THRESHOLD in the obvious way. First, we show that LOWER BOUND lies in co-NP by reducing it to MULTI-MEAN-PAYOFF-THRESHOLD. Suppose we have an instance, (G, C, \vec{z}_C) , and we want to determine if it is in LOWER-BOUND. We can do this by forming a two-player, multi-mean-payoff game, $G' = (V_1, V_2, v^0, E, w', z^k)$. Here we have $V_1 = \text{St}$, $V_2 = \text{St} \times \text{Ac}_C$ and $v^0 = s^0$. Additionally, the set of edges of G' , E , is defined as,

$$E = \{(s, (s, \text{ac}_C)) \mid s \in \text{St}, \text{ac}_C \in \text{Ac}_C\} \\ \cup \{(s, \text{ac}_C), \text{tr}(s, (\text{ac}_C, \text{ac}_{\text{Ag} \setminus C})) \mid \text{ac}_{\text{Ag} \setminus C} \in \text{Ac}_{\text{Ag} \setminus C}\},$$

and the weight function, $w' : E \rightarrow \mathbb{Z}^{|\text{Cl}|}$, is defined by the following two patterns:

$$w'_i(s, (s, \text{ac}_C)) = w_i(s); \\ w'_i((s, \text{ac}_C), \text{tr}(s, (\text{ac}_C, \text{ac}_{\text{Ag} \setminus C}))) = w_i(s).$$

Finally, we set $z^{|\text{Cl}|}$ to be \vec{z}_C .

Informally, the two players of the game are C and $\text{Ag} \setminus C$, the vector weight function is given by aggregating the weight functions of C and the threshold is \vec{z}_C . Now, if in this game, player 1 has a winning strategy, then there exists some strategy $\vec{\sigma}_C$ such that for all strategies of player 2, $\vec{\sigma}_{\text{Ag} \setminus C}$, we have that $\rho(\vec{\sigma}_C, \vec{\sigma}_{\text{Ag} \setminus C})$ is a winning run for player 1. But this means that $\text{pay}_i(\rho(\vec{\sigma}_C, \vec{\sigma}_{\text{Ag} \setminus C})) \geq z_i$ for all $i \in C$. But it is easy to verify that this implies that \vec{z}_C is a lower bound for C in G . Conversely, if player 1 has no winning strategy, then for all strategies, $\vec{\sigma}_C$, there exists some strategy $\vec{\sigma}_{\text{Ag} \setminus C}$ such that $\rho(\vec{\sigma}_C, \vec{\sigma}_{\text{Ag} \setminus C})$ is not a winning run. This in turn implies that for some $j \in C$, we have that $\text{pay}_j(\rho(\vec{\sigma}_C, \vec{\sigma}_{\text{Ag} \setminus C})) < z_j$, which means that \vec{z}_C is not a lower bound for C in G . Also note that this construction can be performed in polynomial time, giving us the co-NP upper bound.

For the lower bound, we go the other way and reduce from MULTI-MEAN-PAYOFF-THRESHOLD. Suppose we would like to determine if an instance G is in MULTI-MEAN-PAYOFF-THRESHOLD. Then we form a concurrent mean-payoff game, G' , with $k+1$ players, where the states of G' coincide exactly with the states of G . In this game, only the 1st and $(k+1)$ th player have any influence on the strategic nature of the game. If the game is in a state in V_1 , player one can decide which state to move into next. Otherwise, if the game is in a state within V_2 , then the $(k+1)$ th player makes a move. Note we only allow moves that agree with moves allowed within G .

Now, in G' , the first k players have weight functions corresponding correspond to the k weight functions of player 1 in G . The last player can have any arbitrary weight function. With this machinery in place, we ask if z^k is a lower bound for $\{1, \dots, k\}$. In a similar manner of reasoning to the above, it is easy to verify that G is an instance of MULTI-MEAN-PAYOFF-THRESHOLD if and only if z^k is a lower bound for $\{1, \dots, k\}$ in the constructed concurrent mean-payoff game. Moreover, this reduction can be done in polynomial time and we can conclude that LOWER BOUND is co-NP-complete. \square

A notable omission from this section is that we have not presented any bounds for the complexity of E-CORE in the general case. One reason for the upper bounds remaining elusive to us is due to the fact that whilst in a multi-mean-payoff game, player 2 can act optimally with memoryless strategies, player 1 may require infinite memory [21,48]. Given the close connection between the core in our concurrent, multi-agent setting and winning strategies in multi-mean-payoff games, this raises computational concerns for the E-CORE problem. Additionally, in [47], the authors study the Pareto frontier of multi-mean-payoff games, and provide an algorithm for constructing a representation of the achievable values of a given game, but this procedure takes an exponential amount of time in the size of the input. The same paper also establishes Σ_2^P -completeness for the *polyhedron value problem*. Both of these problems appear to be intimately related to the core, and we hope we might be able to use these results to gain more insight into the E-CORE in the future.

5. Discussion and related work

In this section, we present some conclusions, briefly discuss related work, discuss some issues related to the core, and speculate about how to implement our approach using model checking techniques.

5.1. Coalition formation in cooperative games

Coalition formation with externalities has been studied in the cooperative game-theory literature [16,39,40]. These works considered several possible formulations of the core. For instance, the α -core takes the pessimistic approach that requires that all members of a deviating coalition, S , will benefit from the deviation regardless of the behaviour of the other coalitions that may be formed. Our first definition of the core follows this approach. In contrast, β -core takes an optimistic approach, and requires that the members of a deviating coalition S will benefit from at least one possibility of coalition formation of the rest of the players. In addition, γ -core [49,50] assumes that the coalition structure that will be created after a deviation will include the deviating coalition S and the rest of the coalition structure will consist of all singletons. The “worth” of S is now defined as equal to its payoff in the Nash equilibrium between S and the other players acting individually, in which the members of S play their joint best response strategy against the individually best response strategies of the remaining players. It is well-known that α - and β -characteristic functions lead to large cores [51], which is consistent with our observation that, with respect to our first definition, the core is never empty. Coalition formation is important in multi-agent system [52]. However, even though coalition formation with externalities is a widely-studied problem in multi-agent systems, not much work has studied the concept of stability in multi-agent coalition formation with externalities [53]. Instead, in artificial intelligence and multi-agent systems, most research has focused on the structure formation itself [54]. Notice that, from the point of view of cooperative game theory, our games are games of non-transferable utility (i.e., NTU games) [15, p.71]. For certain types of NTU games, there are other possible approaches to defining the core, for example through a translation into a conventional (transferable utility) game (see, e.g. [55, p.238]). This approach might conceivably be used in our setting.

5.2. Rational verification of concurrent games

The formal verification of temporal logic properties of multi-agent systems, while assuming rational behaviour of the agents in such a system, has been studied for almost a decade now; see, for instance, [2–4,43,56]. However, to the best of our knowledge, all these studies have considered a non-cooperative setting, even if coalitional power is allowed, for instance, as in a strong Nash equilibrium. Nonetheless, also in such non-cooperative settings, the complexity of checking whether a temporal logic property is satisfied in a stable outcome of the game is a 2EXPTIME-complete problem, even for two-player zero-sum games where only trivial coalitions can be formed. On the positive side, cooperative games seem to have better model-theoretic properties in the rational verification framework: with respect to our first definition of the core (which corresponds to the concept of α -core in the literature of cooperative games), a witness in the core is always guaranteed (since the core is never empty), preserved across bisimilar systems, and can be checked in practice using ATL* model checking techniques, which are supported by, e.g., MCMAS [10], an automated formal verification tool that can perform ATL* model checking (via an implementation of SL[1G], which subsumes ATL* [57]) and allows specifications of concurrent game structures in ISPL (Interpreted Systems Programming Language, the modelling language of MCMAS, based on the interpreted systems formalism [58]).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research has been supported by: a UKRI AI World Leading Researcher Fellowship, grant number EP/W002949/1 (Wooldridge); the EU Horizon 2020 programme under project 952215 “TAILOR” (Kraus and Woodridge); the EPSRC Centre

for Doctoral Training in Autonomous Intelligent Machines and Systems EP/L015897/1 (Steeple); the Ian Palmer Memorial Scholarship at the University of Oxford (Steeple); and, the Israel Science Foundation under grant 1958/20 (Kraus).

Appendix A. Proof that Example 3 has an empty strong core

We claimed earlier that there exist four players games with an empty strong core, and gave an example of such a game in Example 3. Below, we present a table giving, for every action profile, a strong beneficial deviation for some set of players. The notation $\{2 \mapsto 0, 4 \mapsto 1\}$ means that the coalition $\{2, 4\}$ has a strong beneficial deviation where player 2 plays 0 and player 4 player 1. Note that a strong beneficial deviation for a coalition does not require all players to change their action from the status quo.

Action profile	Set of winning players	Strong beneficial deviation
0000	{1, 2}	{3 \mapsto 1}
0001	{1, 2}	{3 \mapsto 1}
0010	{1, 3}	{2 \mapsto 1, 4 \mapsto 1}
0011	{1, 3}	{2 \mapsto 1, 4 \mapsto 1}
0100	{1, 2}	{3 \mapsto 1}
0101	{1, 4}	{2 \mapsto 0}
0110	{1, 3}	{2 \mapsto 1, 4 \mapsto 1}
0111	{2, 4}	{1 \mapsto 1}
1000	{3, 4}	{1 \mapsto 0}
1001	{2, 3}	{1 \mapsto 0}
1010	{2, 3}	{1 \mapsto 0}
1011	{2, 3}	{1 \mapsto 0}
1100	{1, 2}	{3 \mapsto 1}
1101	{1, 4}	{2 \mapsto 0}
1110	{2, 3}	{1 \mapsto 0}
1111	{1, 4}	{2 \mapsto 0}

This table illustrates that the provided game has an empty strong core. It is an easy, but somewhat tedious, task to verify that the deviating players win under all counter-responses under which the original winning players win.

Appendix B. Proof of Theorem 8

We will prove this theorem by assuming the existence of a three player game G with an empty strong core, and then proving two contradictory facts, which immediately implies the result. The two claims are:

1. There is some strategy $\vec{\sigma}$ on G that models exactly two of $\gamma_1, \gamma_2, \gamma_3$.
2. There is no strategy $\vec{\sigma}$ on G that models exactly two of $\gamma_1, \gamma_2, \gamma_3$.

We use the following lemma throughout:

Lemma 5. *Let G be a three-player game with an empty strong core. Then there does not exist a strategy profile which satisfies the goals of all three players.*

Proof. If G had a strategy satisfying all of the players' goals, such a strategy would be in the strong core, which is a contradiction. \square

The first claim we need to prove is the more straightforward of the two:

Lemma 6. *Let G be a three-player game with an empty strong core. Then there exists some strategy $\vec{\sigma}$ on G that models exactly two of $\gamma_1, \gamma_2, \gamma_3$.*

Proof. Given claim 1, we know that every strategy on G must model exactly zero, one or two goals, but not three. Suppose, for contradiction, that G only has strategies modelling none or one of $\gamma_1, \gamma_2, \gamma_3$. First, note that if G only had strategies satisfying zero of $\gamma_1, \gamma_2, \gamma_3$, none of those strategies could admit a strong beneficial deviation, and so would all be in the $core^+$. Hence, G must have a strategy satisfying exactly one of $\gamma_1, \gamma_2, \gamma_3$. Let such a strategy be denoted $\vec{\sigma} = (\sigma_1, \sigma_2, \sigma_3)$, and without loss of generality, we may assume that it satisfies only γ_1 . Now, as by assumption $\vec{\sigma}$ does not lie in $core^+(G)$, there must be a strong beneficial deviation from this strategy for some coalition $C \subseteq \{2, 3\}$. Since under a strong beneficial deviation, every member of the coalition is winning, and since we assumed that G only has strategies satisfying less than two of the goals, the coalition must be a singleton. Again, without loss of generality, we may assume that it is player 2 who has a deviation σ'_2 . By assumption, we have that $\sigma_1, \sigma'_2, \sigma_3$ only models γ_2 .

But again, $(\sigma_1, \sigma'_2, \sigma_3)$ must have a strong beneficial deviation for 1 or 3 to a strategy satisfying only γ_1 , or only γ_3 , respectively. If the former case were true, then this would contradict the fact that σ'_2 was a strong beneficial deviation from $\vec{\sigma}$. Hence, we must be in the latter case, and have a strong beneficial deviation σ'_3 for player 3.

The new strategy, $(\sigma_1, \sigma'_2, \sigma'_3)$ must once again admit a strong beneficial deviation. If it admitted a strong beneficial deviation σ'_1 for 1 satisfying γ_1 only, then the joint strategy (σ'_1, σ'_3) would be a punishment for σ'_2 . Otherwise, if a strong beneficial deviation σ''_2 for 2 to a strategy satisfied γ_2 only, then (σ_1, σ''_2) would constitute a punishment for σ'_3 . Either way, we reach a contradiction, concluding our proof. \square

Before proving the second claim, we will need to appeal to two additional technical lemmas. The first is the following:

Lemma 7. *Let $\{a, b, c\}$ be a permutation of $\{1, 2, 3\}$. Then there do not exist strategies $(\sigma_a, \sigma_b, \sigma_c)$, $(\sigma_a, \sigma_b, \sigma'_c)$, $(\sigma'_a, \sigma_b, \sigma'_c)$ of G such that all of the following hold:*

- $(\sigma_a, \sigma_b, \sigma_c)$ satisfies γ_a and γ_b but not γ_c
- $(\sigma_a, \sigma_b, \sigma'_c)$ satisfies γ_c only
- $(\sigma'_a, \sigma_b, \sigma'_c)$ satisfies γ_a only
- σ'_c is a beneficial deviation for c from $(\sigma_a, \sigma_b, \sigma_c)$ to $(\sigma_a, \sigma_b, \sigma'_c)$
- σ'_a is a beneficial deviation for a from $(\sigma_a, \sigma_b, \sigma'_c)$ to $(\sigma'_a, \sigma_b, \sigma'_c)$

Proof. For a contradiction, we assume the statement of the lemma is not the case, and all of the statements hold. Without loss of generality, we can also assume that $a = 1$, $b = 2$, and $c = 3$.

Now, the third strategy, $(\sigma'_1, \sigma_2, \sigma'_3)$ must have a strong beneficial deviation for 2, 3, or both. If it has a deviation (σ'_2, σ'_3) (with possibly $\sigma'_2 = \sigma_2$) to a strategy that satisfies γ_3 but not γ_1 , then it would be a punishment for σ'_1 . Therefore, the only possible sets of winners in the new strategy are $\{2\}$, $\{1, 2\}$ and $\{1, 3\}$. We consider each of these cases in turn.

If player 2 is the only winner under the new strategy, the strong beneficial deviation must be σ'_2 by 2 only, to the strategy $(\sigma'_1, \sigma'_2, \sigma'_3)$.

Again, this strategy has to have a beneficial deviation for 1, 3, or both. Let that deviation be (σ''_1, σ'_3) with possibly $\sigma''_1 = \sigma'_1$ or $\sigma'_3 = \sigma_3$ but not both. Consider the following cases for the resulting strategy, $(\sigma''_1, \sigma'_2, \sigma'_3)$:

- If only γ_1 , or exactly γ_1 and γ_3 are satisfied, then (σ''_1, σ'_3) is a punishment for σ'_2 ; a contradiction.
- If exactly γ_1 and γ_2 are satisfied, then since 3 loses, we must necessarily have $\sigma'_3 = \sigma_3$ (the deviation is just by 1). In this case, the joint strategy (σ''_1, σ'_2) can take us from $(\sigma_1, \sigma_2, \sigma'_3)$ to $(\sigma''_1, \sigma'_2, \sigma'_3)$ and form a punishment for σ'_3 ; a contradiction.
- If only γ_3 , or exactly γ_2 and γ_3 are satisfied, then since either way 1 loses, we must have $\sigma''_1 = \sigma'_1$ and the joint strategy (σ'_2, σ'_3) can take us from $(\sigma'_1, \sigma_2, \sigma'_3)$ to $(\sigma'_1, \sigma'_2, \sigma'_3)$ and form a punishment for σ'_1 ; a contradiction.

Now suppose that the winners in the new strategy are 1 and 2. Since 1 was a winner in $(\sigma'_1, \sigma_2, \sigma'_3)$ already, the strong beneficial deviation must once again be σ'_2 by 2 only, to the strategy $(\sigma'_1, \sigma'_2, \sigma'_3)$.

But now, we can see that the joint strategy (σ'_1, σ'_2) is a punishment for σ'_3 (both 1 and 2 remain winners); a contradiction.

Finally, suppose the winners in the new strategy are 1 and 3. Since 1 was a winner in $(\sigma'_1, \sigma_2, \sigma'_3)$ already, the strong beneficial deviation must this time be σ'_3 by 3 only, to the strategy $(\sigma'_1, \sigma_2, \sigma'_3)$.

The only loser, 2, must have a strong beneficial deviation σ'_2 from this strategy. Now, see that:

- If exactly γ_1 and γ_2 were satisfied in the resulting strategy – $(\sigma'_1, \sigma'_2, \sigma'_3)$, then the joint strategy (σ'_1, σ'_2) would be a punishment for σ'_3 .
- If exactly γ_2 and γ_3 were satisfied in $(\sigma'_1, \sigma'_2, \sigma'_3)$ then the joint strategy (σ'_2, σ'_3) would be a punishment for σ'_1 .

Hence, 2 must be the only winner in $(\sigma'_1, \sigma'_2, \sigma'_3)$.

Once again, this strategy must admit a strong beneficial deviation for 1, 3, or both. Let that deviation be (σ''_1, σ'_3) with possibly $\sigma''_1 = \sigma'_1$ or $\sigma'_3 = \sigma_3$ but not both. Consider the following cases for the resulting strategy, $(\sigma''_1, \sigma'_2, \sigma'_3)$:

- If only γ_1 , or exactly γ_1 and γ_2 were satisfied then we would necessarily have $\sigma'_3 = \sigma_3$ and the joint strategy (σ''_1, σ'_2) would be a punishment for σ'_3 , contradiction.
- If exactly γ_1 and γ_3 were satisfied then (σ''_1, σ'_3) would be a punishment for σ'_2 , contradiction.
- If only γ_3 , or exactly γ_2 and γ_3 were satisfied then we would necessarily have $\sigma''_1 = \sigma'_1$ and the joint strategy (σ'_2, σ'_3) would be a punishment for σ'_1 , contradiction.

Thus all three possible sets of winners lead to a contradiction, yielding our result. \square

The second of the two technical lemmas is then as follows:

Lemma 8. Let $\{a, b, c\} = \{1, 2, 3\}$ be the set Ag . There cannot be strategies $(\sigma_a, \sigma_b, \sigma_c)$, $(\sigma'_a, \sigma'_b, \sigma'_c)$ of G such that all of the following hold:

- $(\sigma_a, \sigma_b, \sigma_c)$ satisfies γ_a and γ_b but not γ_c
- $(\sigma_a, \sigma_b, \sigma'_c)$ satisfies γ_c only
- σ'_c is a strong beneficial deviation for c from $(\sigma_a, \sigma_b, \sigma_c)$ to $(\sigma_a, \sigma_b, \sigma'_c)$

Proof. For contradiction, assume otherwise. Without loss of generality, we can again assume that $a = 1$, $b = 2$, and $c = 3$.

The second strategy, $(\sigma_a, \sigma_b, \sigma'_c)$ must admit a strong beneficial deviation for 1, 2, or both. However, see that a deviation to a strategy satisfying only γ_1 or only γ_2 would contradict Lemma 7. Hence, the new strategy must have exactly two winners. Moreover, if the winners in the new strategy were exactly 1 and 2, the deviation would provide a punishment for σ'_3 . Therefore, without loss of generality, we can assume that $(\sigma_a, \sigma_b, \sigma'_c)$ has a strong beneficial deviation σ'_1 for 1 to a strategy satisfying exactly γ_1 and γ_3 .

Now, the only loser in $(\sigma'_1, \sigma_2, \sigma'_3)$, 2, must have a strong beneficial deviation σ'_2 . If however, exactly γ_1 and γ_2 were satisfied in $(\sigma'_1, \sigma'_2, \sigma'_3)$, then (σ'_1, σ'_2) would be a punishment for σ'_3 . If exactly γ_2 and γ_3 were satisfied, (σ'_2, σ'_3) would provide a punishment for σ'_1 . Hence, 2 must be the only winner.

Once again a strong beneficial deviation from $(\sigma'_1, \sigma'_2, \sigma'_3)$ to a strategy satisfying only γ_1 or only γ_3 would contradict Lemma 7. Therefore, there must be a strong beneficial deviation by 1, 3, or both, to a strategy with exactly two winners. Now:

- If the winners were 1 and 3, the deviation would provide a punishment for σ'_2 .
- If the winners were 1 and 2 then the deviation must be σ''_1 by 1 only (as 3 loses). But then, the joint strategy (σ''_1, σ'_2) would provide a punishment for the initial σ'_3 .
- If the winners were 2 and 3 then the deviation must be σ''_3 by 3 only. The joint strategy (σ'_2, σ''_3) would be a punishment for σ'_1 .

This exhausts all the cases, each of which lead to a contradiction, and finishes the proof of Lemma 8. \square

With these two technical lemmas in place, we are ready to prove the second part of our claim.

Lemma 9. Let G be a three-player game with an empty strong core. Then there does not exist a strategy profile $\vec{\sigma}$ on G that models exactly two of $\gamma_1, \gamma_2, \gamma_3$.

Proof. For a contradiction, assume otherwise. Without a loss of generality, let $(\sigma_1, \sigma_2, \sigma_3)$ be a strategy satisfying exactly γ_1 and γ_2 . It must admit a strong beneficial deviation σ'_3 for the lone loser, 3. Moreover, by Lemma 8, he cannot win alone in the new strategy. Therefore, without loss of generality, we can assume that 1 and 3 are winners in $(\sigma_1, \sigma_2, \sigma'_3)$.

Again, by Lemma 8, the new strategy must have a strong beneficial deviation σ'_2 by 2 to a strategy where exactly two players win. However, if 1 and 2 were winners, then (σ_1, σ'_2) would constitute a punishment for σ'_3 . Therefore, exactly γ_2 and γ_3 must be satisfied in $(\sigma_1, \sigma'_2, \sigma'_3)$.

But now, again by Lemma 8, there must be a strong beneficial deviation σ'_1 for 1 to another strategy with exactly two winners. However, if those were 1 and 2, (σ'_1, σ'_2) would be a punishment for σ'_3 . Else, if the winners were 1 and 3 then (σ'_1, σ'_3) would form a punishment for σ'_2 . This concludes our proof. \square

Finally, the required result simply follows by combining the two lemmas.

Theorem 13. If G is a three-player game, then it has a non-empty strong core.

Proof. Result follows by combining Lemma 6 and Lemma 9. \square

References

- [1] J. Gutierrez, S. Kraus, M.J. Wooldridge, Cooperative concurrent games, in: AAMAS, IFAAMAS, 2019, pp. 1198–1206.
- [2] J. Gutierrez, P. Harrenstein, M. Wooldridge, Iterated boolean games, Inf. Comput. 242 (2015) 53–79, <https://doi.org/10.1016/j.ic.2015.03.011>.
- [3] J. Gutierrez, P. Harrenstein, M. Wooldridge, From model checking to equilibrium checking: reactive modules for rational verification, Artif. Intell. 248 (2017) 123–157, <https://doi.org/10.1016/j.artint.2017.04.003>.
- [4] M. Wooldridge, J. Gutierrez, P. Harrenstein, E. Marchioni, G. Perelli, A. Toumi, Rational verification: from model checking to equilibrium checking, in: Proc. of AAAI, AAAI Press, 2016, pp. 4184–4191.
- [5] A. Abate, J. Gutierrez, L. Hammond, P. Harrenstein, M. Kwiatkowska, M. Najib, G. Perelli, T. Steeples, M.J. Wooldridge, Rational verification: game-theoretic verification of multi-agent systems, Appl. Intell. 51 (9) (2021) 6569–6584, <https://doi.org/10.1007/s10489-021-02658-y>.
- [6] R. Alur, T.A. Henzinger, O. Kupferman, Alternating-time temporal logic, J. ACM 49 (5) (2002) 672–713, <https://doi.org/10.1145/585265.585270>.
- [7] E.A. Emerson, Temporal and modal logic, in: Formal Models and Semantics, Elsevier, 1990, pp. 995–1072.
- [8] E.M. Clarke, O. Grumberg, D.A. Peled, Model Checking, MIT Press, 2002.

- [9] J. Gutierrez, M. Najib, G. Perelli, M. Wooldridge, EVE: a tool for temporal equilibrium analysis, in: ATVA, in: Lecture Notes in Computer Science, vol. 11138, Springer, 2018, pp. 551–557.
- [10] A. Lomuscio, H. Qu, F. Raimondi, MCMAS: an open-source model checker for the verification of multi-agent systems, *Int. J. Softw. Tools Technol. Transf.* 19 (1) (2017) 9–30, <https://doi.org/10.1007/s10009-015-0378-x>.
- [11] R. Brenguier, PRALINE: a tool for computing Nash equilibria in concurrent games, in: CAV, in: Lecture Notes in Computer Science, vol. 8044, Springer, 2013, pp. 890–895.
- [12] M. Kwiatkowska, G. Norman, D. Parker, G. Santos, {PRISM}-games 3.0: stochastic game verification with concurrency, equilibria and time, in: S.K. Lahiri, C. Wang (Eds.), Computer Aided Verification - 32nd International Conference, CAV 2020, Proceedings, Part II, Los Angeles, CA, USA, July 21–24, 2020, in: Lecture Notes in Computer Science, vol. 12225, Springer, 2020, pp. 475–487.
- [13] M.J. Osborne, A. Rubinstein, A Course in Game Theory, MIT Press, 1994.
- [14] M. Maschler, E. Solan, S. Zamir, Game Theory, Cambridge University Press, 2013.
- [15] G. Chalkiadakis, E. Elkind, M. Wooldridge, Computational Aspects of Cooperative Game Theory, Morgan-Claypool, 2011.
- [16] M. Uyanik, On the nonemptiness of the α -core of discontinuous games: transferable and nontransferable utilities, *J. Econ. Theory* 158 (2015) 213–231, <https://doi.org/10.1016/j.jet.2015.04.005>.
- [17] J. Gutierrez, P. Harrenstein, G. Perelli, M. Wooldridge, Nash equilibrium and bisimulation invariance, in: CONCUR, in: LIPIcs, vol. 85, Schloss Dagstuhl, 2017, 17.
- [18] A. Ehrenfeucht, J. Mycielski, Positional strategies for mean payoff games, *Int. J. Game Theory* 8 (2) (1979) 109–113, <https://doi.org/10.1007/bf01768705>.
- [19] E.A. Emerson, C.S. Jutla, Tree automata, mu-calculus and determinacy (extended abstract), in: 32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1–4 October 1991, IEEE Computer Society, 1991, pp. 368–377.
- [20] H. Gimbert, W. Zielonka, Games where you can play optimally without any memory, in: M. Abadi, L. de Alfaro (Eds.), CONCUR 2005 - Concurrency Theory, 16th International Conference, Proceedings, CONCUR 2005, San Francisco, CA, USA, August 23–26, 2005, in: Lecture Notes in Computer Science, vol. 3653, Springer, 2005, pp. 428–442.
- [21] E. Kopczynski, Half-positional determinacy of infinite games, in: M. Bugliesi, B. Preneel, V. Sassone, I. Wegener (Eds.), Automata, Languages and Programming, 33rd International Colloquium, Proceedings, Part II, ICALP 2006, Venice, Italy, July 10–14, 2006, in: Lecture Notes in Computer Science, vol. 4052, Springer, 2006, pp. 336–347.
- [22] A. Pnueli, The temporal logic of programs, in: 18th Annual Symposium on Foundations of Computer Science (sfcs 1977), IEEE, 1977, pp. 46–57.
- [23] F. Mogavero, A. Murano, G. Perelli, M.Y. Vardi, Reasoning about strategies, *ACM Trans. Comput. Log.* 15 (4) (2014) 1–47, <https://doi.org/10.1145/2631917>.
- [24] C. Baier, J.-P. Katoen, Principles of Model Checking, MIT Press, 2008.
- [25] R.M. Keller, Formal verification of parallel programs, *Commun. ACM* 19 (7) (1976) 371–384, <https://doi.org/10.1145/360248.360251>, Association for Computing Machinery, New York, NY, USA.
- [26] S.A. Kripke, Semantical analysis of modal logic i normal modal propositional calculi, *Z. Math. Log. Grundl. Math.* 9 (5–6) (1963) 67–96, <https://doi.org/10.1002/maalq.19630090502>.
- [27] E.A. Emerson, J.Y. Halpern, “Sometimes” and “not never” revisited, *J. ACM* 33 (1) (1986) 151–178, <https://doi.org/10.1145/4904.4999>.
- [28] A.P. Sistla, E.M. Clarke, The complexity of propositional linear temporal logics, *J. ACM* 32 (3) (1985) 733–749, <https://doi.org/10.1145/3828.3837>.
- [29] A. Pnueli, R. Rosner, On the synthesis of a reactive module, in: Conference Record of the Sixteenth Annual ACM Symposium on Principles of Programming Languages, Austin, Texas, USA, January 11–13, 1989, ACM Press, 1989, pp. 179–190.
- [30] D. Fisman, O. Kupferman, Y. Lustig, Rational synthesis, in: J. Esparza, R. Majumdar (Eds.), Tools and Algorithms for the Construction and Analysis of Systems, 16th International Conference, TACAS 2010, Held as Part of the Joint European Conferences on Theory and Practice of Software, Proceedings, ETAPS 2010, Paphos, Cyprus, March 20–28, 2010, in: Lecture Notes in Computer Science, vol. 6015, Springer, 2010, pp. 190–204.
- [31] R.J. Aumann, Acceptable points in general cooperative n -person games, in: Contributions to the Theory of Games, Vol. IV, in: Annals of Mathematics Studies, vol. 40, Princeton University Press, Princeton, N.J., 1959, pp. 287–324.
- [32] R.J. Aumann, Acceptable points in games of perfect information, *Pac. J. Math.* 10 (1960) 381–417.
- [33] B.D. Bernheim, B. Peleg, M.D. Whinston, Coalition-proof Nash equilibria I. Concepts, *J. Econ. Theory* 42 (1) (1987) 1–12, [https://doi.org/10.1016/0022-0531\(87\)90099-8](https://doi.org/10.1016/0022-0531(87)90099-8), <https://www.sciencedirect.com/science/article/pii/0022053187900998>.
- [34] B.D. Bernheim, M.D. Whinston, Coalition-proof Nash equilibria II. Applications, *J. Econ. Theory* 42 (1) (1987) 13–29, [https://doi.org/10.1016/0022-0531\(87\)90100-1](https://doi.org/10.1016/0022-0531(87)90100-1), <https://www.sciencedirect.com/science/article/pii/0022053187901001>.
- [35] M. Hennessy, R. Milner, Algebraic laws for nondeterminism and concurrency, *J. ACM* 32 (1) (1985) 137–161, <https://doi.org/10.1145/2455.2460>.
- [36] R. Alur, S. La Torre, Deterministic generators and games for Ltl fragments, *ACM Trans. Comput. Log.* 5 (1) (2004) 1–25, <https://doi.org/10.1145/963927.963928>.
- [37] O. Kupferman, M.Y. Vardi, P. Wolper, An automata-theoretic approach to branching-time model checking, *J. ACM* 47 (2) (2000) 312–360, <https://doi.org/10.1145/333979.333987>.
- [38] R. Milner, Communication and Concurrency, Prentice-Hall, Inc., USA, 1989.
- [39] S.-S. Yi, Stable coalition structures with externalities, *Games Econ. Behav.* 20 (2) (1997) 201–237, <https://doi.org/10.1006/game.1997.0567>.
- [40] M. Finus, B. Rundshagen, A non-cooperative foundation of core-stability in positive externality (NTU)-coalition games, *SSRN Electron. J.* (2003), <https://doi.org/10.2139/ssrn.406400>.
- [41] M. Mavronicolas, B. Monien, K.W. Wagner, Weighted boolean formula games, in: X. Deng, F.C. Graham (Eds.), Internet and Network Economics, Third International Workshop, Proceedings, WINE 2007, San Diego, CA, USA, December 12–14, 2007, in: Lecture Notes in Computer Science, vol. 4858, Springer, 2007, pp. 469–481.
- [42] S. Almagor, O. Kupferman, G. Perelli, Synthesis of controllable Nash equilibria in quantitative objective game, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI’18, International Joint Conferences on Artificial Intelligence Organization, event-place: Stockholm, Sweden, 2018, pp. 35–41.
- [43] O. Kupferman, G. Perelli, M.Y. Vardi, Synthesis with rational environments, *Ann. Math. Artif. Intell.* 78 (1) (2016) 3–20, <https://doi.org/10.1007/s10472-016-9508-8>.
- [44] U. Zwick, M. Paterson, The complexity of mean payoff games on graphs, *Theor. Comput. Sci.* 158 (1–2) (1996) 343–359, [https://doi.org/10.1016/0304-3975\(95\)00188-3](https://doi.org/10.1016/0304-3975(95)00188-3).
- [45] M. Ummels, D. Wojtczak, The complexity of Nash equilibria in stochastic multiplayer games, *Log. Methods Comput. Sci.* 7 (3) (Sep. 2011), [https://doi.org/10.2168/lmcs-7\(3:20\)2011](https://doi.org/10.2168/lmcs-7(3:20)2011).
- [46] R.M. Karp, A characterization of the minimum cycle mean in a digraph, *Universitext* 23 (3) (1978) 309–311, [https://doi.org/10.1016/0012-365x\(78\)90011-0](https://doi.org/10.1016/0012-365x(78)90011-0).
- [47] R. Brenguier, J.-F. Raskin, Pareto curves of multidimensional mean-payoff games, in: D. Kroening, C.S. Pasareanu (Eds.), Computer Aided Verification - 27th International Conference, Proceedings, Part II, CAV 2015, San Francisco, CA, USA, July 18–24, 2015, in: Lecture Notes in Computer Science, vol. 9207, Springer, 2015, pp. 251–267.
- [48] Y. Velner, K. Chatterjee, L. Doyen, T.A. Henzinger, A. Rabinovich, J.-F. Raskin, The complexity of multi-mean-payoff and multi-energy games, *Inf. Comput.* 241 (2015) 177–196, <https://doi.org/10.1016/j.ic.2015.03.001>.

- [49] P. Chander, et al., Cores of games with positive externalities, CORE DP 2010/4, 2010.
- [50] P. Chander, The gamma-core and coalition formation, *Int. J. Game Theory* 35 (4) (2007) 539–556, <https://doi.org/10.1007/s00182-006-0067-9>.
- [51] D. Ray, R. Vohra, Equilibrium binding agreements, *J. Econ. Theory* 73 (1) (1997) 30–78, <https://doi.org/10.1006/jeth.1996.2236>.
- [52] O. Shehory, S. Kraus, Methods for task allocation via agent coalition formation, *Artif. Intell.* 101 (1–2) (1998) 165–200, [https://doi.org/10.1016/s0004-3702\(98\)00045-9](https://doi.org/10.1016/s0004-3702(98)00045-9).
- [53] D. Mutzari, J. Gan, S. Kraus, Coalition formation in multi-defender security games, *Proc. AAAI Conf. Artif. Intell.* 35 (6) (2021) 5603–5610.
- [54] T. Rahwan, T. Michalak, M. Wooldridge, N.R. Jennings, Anytime coalition structure generation in multi-agent systems with positive or negative externalities, *Artif. Intell.* 186 (2012) 95–122, <https://doi.org/10.1016/j.artint.2012.03.007>.
- [55] P. Borm, Y. Ju, D. Wettstein, Rational bargaining in games with coalitional externalities, *J. Econ. Theory* 157 (2015) 236–254, <https://doi.org/10.1016/j.jet.2015.01.011>.
- [56] O. Kupferman, Y. Lustig, What triggers a behavior?, in: *Formal Methods in Computer Aided Design (FMCAD'07)*, in: *Lecture Notes in Computer Science*, vol. 6015, IEEE, 2007, pp. 190–204.
- [57] P. Cermák, A. Lomuscio, A. Murano, Verifying and synthesising multi-agent systems against one-goal strategy logic specifications, in: B. Bonet, S. Koenig (Eds.), *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, January 25–30, 2015, Austin, Texas, USA, AAAI Press, 2015, pp. 2038–2044.
- [58] R. Fagin, J.Y. Halpern, Y. Moses, M.Y. Vardi, *Reasoning About Knowledge*, MIT Press, 1995.