Rational Verification with Quantitative Probabilistic Goals

David Hyland
University of Oxford
Oxford, United Kingdom
david.hyland@cs.ox.ac.uk

Julian Gutierrez
Monash University
Melbourne, Australia
julian.gutierrez@monash.edu

Krishna Shankaranarayanan
IIT Bombay
Mumbai, India
krishnas@cse.iitb.ac.in

Michael Wooldridge
University of Oxford
Oxford, United Kingdom
mjw@cs.ox.ac.uk

ABSTRACT

We study the rational verification problem for multi-agent systems in a setting where agents have quantitative probabilistic goals. We use concurrent stochastic games to model multi-agent systems and assume players desire to maximise the probability of satisfying their goals, specified using Linear Temporal Logic (LTL). The main decision problem in this setting is whether a given LTL formula is almost surely satisfied on some pure Nash equilibrium of a given game. We prove that this problem is undecidable in the general case, and then characterise the complexity of this problem under various restrictions on strategies. We also study the problem of deciding whether a given strategy profile is a Nash equilibrium in a given game and show that, unlike the previous verification problem, this question is decidable for several common strategy models.

KEYWORDS

Multi-agent systems; formal verification; quantitative probabilistic goals; Linear Temporal Logic; computational game theory.

ACM Reference Format:
David Hyland, Julian Gutierrez, Krishna Shankaranarayanan, and Michael Wooldridge. 2024. Rational Verification with Quantitative Probabilistic Goals. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 9 pages.

1 INTRODUCTION

Over the past decade, there has been much concern about issues surrounding the safety and reliability of AI systems. This apprehension has prompted renewed research into the *verification* of AI systems – demonstrating formally that they behave as intended. In the context of multi-agent systems, one important approach to this problem is that of *rational verification*, whereby we aim to formally verify which temporal logic properties will hold in the system under the assumption that component agents behave strategically and rationally (i.e., game-theoretically) in pursuit of their goals [17, 19, 20, 22, 23, 35]. A standard idea in rational verification is to represent agent preferences by associating with each agent a temporal logical goal formula that it desires to see satisfied. Players are assumed to seek the satisfaction of their goal formulae, taking into account the fact that other players are acting in pursuit of theirs, and that other players are also strategic actors. Note that the use of temporal logic as a framework for representing both agent goals and queries about possible system behaviours leads to commonalities – intuitive and formal – with both the model checking paradigm for automated verification [4, 13] and the automated synthesis paradigm for concurrent and reactive systems [30].

Rational verification has been studied in a wide range of settings, relating both to the semantic model used to represent a multi-agent system, and to the way in which agent preferences are modelled; see [1] for a comprehensive survey. For example, rational verification has been studied for multi-agent systems modelled as deterministic and nondeterministic games, for agent preferences modelled as temporal logic and ω-regular goals, for players having access to either infinite or finite memory strategies, for systems with perfect and imperfect information, and very recently for systems with agents' goals accounting for qualitative probabilistic behaviour [18]. In this paper, we are also interested in systems with probabilistic behaviour, and consider the more general – and often more intractable – setting with *quantitative* probabilistic behaviour [4].

More specifically, we study the rational verification problem in a setting where players have *quantitative probabilistic goals.* As our basic semantic model, we use concurrent stochastic games, which can capture probabilistic and nondeterministic behaviours in multi-agent settings. In our model, agents in the system seek to *maximise the probability* of satisfying their objectives, specified as Linear Temporal Logic (LTL) formulae – a quantitative probabilistic reasoning – while, the central question is whether a given LTL specification is almost surely satisfied in some pure strategy Nash equilibrium of a given game – a qualitative probabilistic condition of satisfaction, which we refer to as the E-Nash problem. We show that, in the most general setting without restrictions on the memory of players' strategies, this problem is undecidable. Given this, we then seek to address the following questions: (1) Can we identify decidable restrictions on the different strategy models? (2) What is the computational complexity of the E-Nash problem under these restrictions? (3) How does the complexity of E-Nash compare to the related problem of checking whether a given strategy profile is a Nash equilibrium in a given game, i.e., the Membership problem? We investigate the computational complexity of both the E-Nash and Membership problems under various natural restrictions on the form of strategies permitted, including memoryless, k-bounded,

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand.

Model / Problem	E-Nash	Membership
General	Undecidable	2EXPTIME
Myopic	Undecidable	2EXPTIME
k-bounded	PSPACE	PSPACE
Memoryless	PSPACE	PSPACE

Table 1: Summary of complexity results.

and myopic strategies. For E-Nash, some of the restrictions we consider give rise to decidability. In the case of Membership, we show that it is decidable in every strategy model we study. Table 1 summarises our main complexity results. For all decidable problems we consider, we give matching upper and lower complexity bounds.

The paper is organised as follows. In Section 2, we give some preliminary definitions and in Sections 3–5, we present the main technical results in the paper, beginning with our main result and then investigating various restrictions on strategies. In Section 6, we finish with some conclusions and a summary of the most relevant related work.

2 PRELIMINARIES

For a finite set X, a (rational) *probability distribution* over X is a function $\Pr : X \to [0,1] \cap \mathbb{Q}$ such that $\Sigma_{x\in X} \Pr(x) = 1$. We write $\mathsf{D}(X)$ for the set of probability distributions on X. For a tuple $\vec{x} = (x_1, \ldots, x_n)$, we write $\mathrm{proj}_i(\vec{x})$ to refer to x_i, *i.e.*, to its i-th *projection* or component; thus, we have $\mathrm{proj}_i(\vec{x}) = x_i$.

Markov Chains: A *Markov chain* (MC) is a tuple $C = (S, s_\iota, \mathrm{tr}, \lambda)$, where S is a set of states, s_ι is the initial state, $\mathrm{tr} : S \to \mathsf{D}(S)$ is a function that assigns a probability distribution (on S) to all states $s \in S$, and $\lambda : S \to 2^{\mathrm{AP}}$ is a labelling function mapping each state to a set of propositions taken from a set of atomic propositions AP. The set of infinite paths in C starting from $s \in S$ is $\mathrm{Paths}(C, s) = \{\pi = s_0 s_1 \cdots \in S^\omega : s_0 = s, \forall k \in \mathbb{N}.\ \mathrm{tr}(s_k, s_{k+1}) > 0\}$. The set of all infinite paths in C is $\mathrm{Paths}(C) = \bigcup_{s\in S} \mathrm{Paths}(C, s)$. The set of finite paths starting from $s \in S$ is defined as $\mathrm{Fpaths}(C, s) = \{\hat{\pi} = s_0 \cdots s_n \in S^+ : \exists \pi \in \mathrm{Paths}(C).\ \hat{\pi}\pi \in \mathrm{Paths}(C, s)\}$ and $\mathrm{Fpaths}(C) = \bigcup_{s\in S} \mathrm{Fpaths}(C, s)$. The *cylinder set* of a finite path $\hat{\pi} \in \mathrm{Fpaths}(C)$ is defined by $\mathrm{Cyl}(\hat{\pi}) = \{\pi \in \mathrm{Paths}(C) : \exists \tilde{\pi} \in \mathrm{Paths}(C).\ \pi = \hat{\pi}\tilde{\pi} \in \mathrm{Paths}(C)\}$. Following [34], we define the probability distribution over the space of infinite paths, as usual, via cylinder sets. We denote this probability distribution over the set of infinite paths beginning from some state s by $\Pr_C^s$. We also write $\Pr_C$ when s is clear from the context.

Concurrent Stochastic Game Arenas: A *concurrent stochastic game arena* (CSGA) is a tuple $\mathcal{A} = (\mathrm{N}, \mathrm{St}, s^0, (\mathrm{Ac}_i)_{i\in\mathrm{N}}, \mathrm{tr})$, where N is a finite set of players, St is a finite set of states, s^0 is the initial state, Ac_i is a finite set of actions for each $i \in \mathrm{N}$. With each player i and each state $s \in \mathrm{St}$, we associate a non-empty set $\mathrm{Ac}_i(s)$ of *available* actions that, intuitively, i can perform when in state s. When all players have fixed their actions, we have an action profile $\vec{a} = (a_1, \ldots, a_n) \in \vec{\mathrm{Ac}} = \mathrm{Ac}_1 \times \cdots \times \mathrm{Ac}_n$, also referred to as a *direction*. A direction $\vec{a}$ is available at $s \in \mathrm{St}$ if for all $i \in \mathrm{N}$ we have $a_i \in \mathrm{Ac}_i(s)$. We write $\vec{\mathrm{Ac}}(s)$ for the set of available directions at state s. For a given set of players $A \subseteq \mathrm{N}$ and an action profile $\vec{a}$, we let $\vec{a}_A$ and $\vec{a}_{-A}$ be two tuples of actions, respectively, one for each player in A and one for each player in $\mathrm{N} \setminus A$. For two directions $\vec{a}$ and $\vec{a}'$, we write $(\vec{a}_A, \vec{a}'_{-A})$ for the direction where the actions for players in A are taken from $\vec{a}$ and the actions for players in $\mathrm{N} \setminus A$ are taken from $\vec{a}'$. Finally, $\mathrm{tr} : \mathrm{St} \times \vec{\mathrm{Ac}} \to \mathsf{D}(\mathrm{St})$ is a probabilistic transition function.

A *sequential stochastic game arena* (SSGA) is a special case of CSGA, given by $\mathcal{S} = (\mathrm{N}, \mathrm{St}, s^0, (\mathrm{Ac}_i)_{i\in\mathrm{N}}, \mathrm{tr})$, which differs from a CSGA only in the available actions of players $\mathrm{Ac}_i(s)$ in states $s \in \mathrm{St}$. In a SSGA, each state $s \in \mathrm{St}$ "belongs" to a player; that is, if s belongs to player i, then $\mathrm{Ac}_i(s) \neq \emptyset$ and $|\mathrm{Ac}_j(s)| = 1$ for all $j \neq i$. Thus, only one player may have multiple actions enabled at each $s \in \mathrm{St}$.

Linear Temporal Logic: Linear Temporal Logic (LTL) [29] extends propositional logic with two operators, $\bigcirc$ ("next") and $\mathbf{U}$ ("until"), that can express properties of paths. The LTL syntax is defined using a set AP of propositional variables and the following grammar:

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

where $p \in \mathrm{AP}$. Other connectives are defined in terms of $\neg$ and $\vee$ in the usual way. Two key derived LTL operators are $\Diamond$ ("eventually") and $\Box$ ("always"), which are defined in terms of $\mathbf{U}$ as follows: $\Diamond\varphi \equiv \top \,\mathbf{U}\, \varphi$ and $\Box\varphi \equiv \neg\Diamond\neg\varphi$.

We interpret formulae of LTL with respect to triples (π, t, λ), where $\pi \in \mathrm{St}^\omega$ is a path, $t \in \mathbb{N}$ is a temporal index in π such that π_t is the tth state in π, and $\lambda : \mathrm{St} \to 2^{\mathrm{AP}}$ is a labelling function that indicates which propositional variables are true in every state. The semantics of LTL is given by the following rules:

$(\pi, t, \lambda) \models \top$		
$(\pi, t, \lambda) \models p$	iff	$p \in \lambda(\pi_t)$
$(\pi, t, \lambda) \models \neg\varphi$	iff	it is not the case that $(\pi, t, \lambda) \models \varphi$
$(\pi, t, \lambda) \models \varphi \vee \psi$	iff	$(\pi, t, \lambda) \models \varphi$ or $(\pi, t, \lambda) \models \psi$
$(\pi, t, \lambda) \models \bigcirc\varphi$	iff	$(\pi, t+1, \lambda) \models \varphi$
$(\pi, t, \lambda) \models \varphi \,\mathbf{U}\, \psi$	iff	there is some $t' \geq t : \big((\pi, t', \lambda) \models \psi$ and for all $t \leq t'' < t' : (\pi, t'', \lambda) \models \varphi\big)$

If $(\pi, 0, \lambda) \models \varphi$, we write $\pi \models \varphi$ and say that π *satisfies* φ.

Concurrent Stochastic Games: A *concurrent stochastic game* (CSG) is a tuple $\mathcal{G} = (\mathcal{A}, (\gamma_i)_{i\in\mathrm{N}}, \lambda)$, where $\mathcal{A}$ is a CSGA, γ_i is an LTL formula that represents the *goal* of player i, and $\lambda : \mathrm{St} \to 2^{\mathrm{AP}}$ is a labelling function. A game is played by each player i selecting a *strategy* σ_i that defines how choices are made over time. A strategy for player i can be understood as a function $\sigma_i : \mathrm{St}^+ \to \mathrm{Ac}_i$ that assigns to every non-empty finite sequence of states an action to be chosen from player i's set of actions. Strategies may require memory to remember the game's history. When a strategy remembers a finite amount of information about the past, we call it *finite-memory*.

We model strategies as transducers. Formally, a strategy in $\mathcal{G}$ for player i is a transducer $\sigma_i = (Q_i, q_i^0, \delta_i, \tau_i)$, where Q_i is a (possibly infinite) set of *internal states*, q_i^0 is the *initial state*, $\delta_i : Q_i \times \mathrm{St} \times \vec{\mathrm{Ac}} \to Q_i$ is a deterministic *internal transition function*, and $\tau_i : Q_i \times \mathrm{St} \times Ac_i \to \{0, 1\}$ a deterministic *action function* that selects a single action (with probability 1) from Ac_i such that for every $q_i \in Q_i$, $s \in \mathrm{St}$, and $a \in Ac_i$, we have that $\tau_i(q_i, s, a) = 1$ only if $a \in \mathrm{Ac}_i(s)$. Let Σ_i be the set of strategies for player i. A strategy is *memoryless* if there exists a transducer encoding the strategy with $|Q_i| = 1$, *i.e.*, the choice of action only depends on the current state of the game, and finite-memory if $|Q_i| < \infty$.

Once every player i has selected a strategy σ_i, we have a *strategy profile* $\vec{\sigma} = (\sigma_1, \dots, \sigma_n)$. We write $\vec{\sigma}_A$ and $\vec{\sigma}_{-A}$ to denote a strategy profile for players in $A \subseteq \mathrm{N}$ and $N \setminus A$, respectively. We also write $(\vec{\sigma}_A, \vec{\sigma}'_{-A})$ to denote the strategy profile where the strategies for players in A are taken from $\vec{\sigma}$, and the strategies for players in $\mathrm{N} \setminus A$ are taken from $\vec{\sigma}'$. Observe that a strategy profile $\vec{\sigma}$ for a game $\mathcal{G}$ resolves nondeterminism in the underlying $\mathcal{A}$. That is, a strategy profile $\vec{\sigma}$ for a game $\mathcal{G}$ induces an MC $C_{\vec{\sigma}} = (S, s_\iota, \mathrm{tr}', \lambda)$, where $S = \mathrm{St} \times \bigtimes_{i \in \mathrm{N}} Q_i$, $s_\iota = (s^0, q_1^0, \dots, q_n^0)$, and for $v, v' \in S$, the probability $\mathrm{tr}'(v, v')$ is given by

$$\sum_{\vec{a} \in \vec{\mathrm{Ac}}} \prod_{a_i \in \vec{a}} \tau_i(\mathrm{proj}_{q_i}(v), \mathrm{proj}_s(v), a_i) \cdot \mathrm{tr}(\mathrm{proj}_s(v), \vec{a}, \mathrm{proj}_s(v')).$$

For a given game $\mathcal{G}$, strategy profile $\vec{\sigma}$, and LTL formula φ, let

$$\Pr(\mathcal{G}, \vec{\sigma}, \varphi) = \Pr_{C_{\vec{\sigma}}}(\{\pi \in \mathrm{Paths}(C_{\vec{\sigma}}, s^0) : \pi \models \varphi\})$$

denote the probability that φ is satisfied in $\mathcal{G}$ under $\vec{\sigma}$. An LTL formula φ is said to be satisfied with probability p, written $\vec{\sigma} \models_p \varphi$, if and only if $\Pr(\mathcal{G}, \vec{\sigma}, \varphi) = p$. Likewise, φ is satisfied with probability at least (similarly, at most) p, written $\vec{\sigma} \models_{[p,1]} \varphi$ ($\vec{\sigma} \models_{[0,p]} \varphi$), iff $\Pr(\mathcal{G}, \vec{\sigma}, \varphi) \geq p$ ($\Pr(\mathcal{G}, \vec{\sigma}, \varphi) \leq p$). Finally, φ is satisfied *almost surely*, written $\vec{\sigma} \models \mathsf{AS}(\varphi)$, when we have $\vec{\sigma} \models_1 \varphi$, and with *non-zero* probability, when we have $\vec{\sigma} \models_{(0,1]} \varphi$. Given a game $\mathcal{G}$ and a path $\pi \in \mathrm{Paths}(C_{\vec{\sigma}}, s^0)$, the utility of each player $i \in \mathrm{N}$ is 1 if $\pi \models \gamma_i$ and 0 otherwise. With this, the expected utility of a player $i \in \mathrm{N}$ under a strategy profile $\vec{\sigma}$ is simply defined $u_i(\vec{\sigma}) = \Pr(\mathcal{G}, \vec{\sigma}, \gamma_i)$. Finally, a *sequential stochastic game* (SSG) $\mathcal{G} = (\mathcal{S}, (\gamma_i)_{i \in \mathrm{N}}, \lambda)$ is a special case of CSG where the underlying arena is an SSGA.

Rational Verification: We now introduce the rational verification problems, in particular, with respect to (pure-strategy) *Nash equilibria* [27]. Formally, given a game $\mathcal{G}$, a strategy profile $\vec{\sigma}$ is a *Nash equilibrium* of $\mathcal{G}$ if, for every player i and strategy $\sigma'_i \in \Sigma_i$, we have

$$u_i(\vec{\sigma}) \geq u_i(\vec{\sigma}_{-i}, \sigma'_i),$$

where $(\vec{\sigma}_{-i}, \sigma'_i)$ is $(\sigma_1, \dots, \sigma_{i-1}, \sigma'_i, \sigma_{i+1}, \dots, \sigma_n)$, the strategy profile where the strategy of player i in $\vec{\sigma}$ is replaced by σ'_i. Let $\mathrm{NE}(\mathcal{G})$ denote the set of Nash equilibria of $\mathcal{G}$. With these definitions in place, we can define the two key decision problems relating to rational verification, in particular for a global LTL property almost-surely satisfied.

MEMBERSHIP
Given: CSG $\mathcal{G}$, strategy profile $\vec{\sigma}$.
Question: Is it the case that $\vec{\sigma} \in \mathrm{NE}(\mathcal{G})$?

E-NASH
Given: CSG $\mathcal{G}$, LTL formula φ.
Question: Is it the case that

$$\exists \vec{\sigma} \in \mathrm{NE}(\mathcal{G}).\ \vec{\sigma} \models \mathsf{AS}(\varphi)?$$

Finally, the intuitively simpler question of asking whether a CSG $\mathcal{G}$ has any Nash equilibria, typically known as the NON-EMPTINESS problem in the rational verification literature, can be solved simply by checking if $(\mathcal{G}, \top) \in$ E-NASH. Note that, in general, the question of NON-EMPTINESS may be non-trivial, as the fact that in our setting strategies can have infinite memory (and thus there are infinitely many of them) means we cannot straightforwardly apply

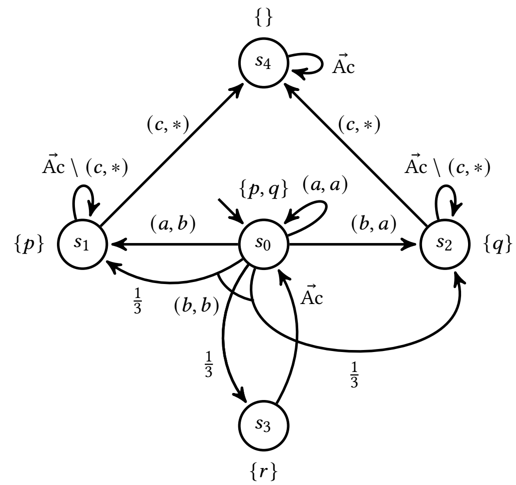

Figure 1: CSGA for Example 1. Edges that are not labelled with probabilities represent deterministic transitions, and $(c,*)$ represents the set of action profiles $\{(c,a),(c,b)\}$.

Nash's theorem. This is a well-known fact in the rational verification literature, which is made even more critical in the quantitative probabilistic setting where optimal behaviour, even for very simple games, may require infinite memory strategies.

Special Cases: Our first main result shows that E-NASH is undecidable in the general case (Theorem 1). Given this, we consider different restrictions on the set of permitted strategies. Firstly, we study the case where players are restricted to using pure memoryless strategies. Secondly, we consider a setting where players are restricted to using polynomially bounded strategies. Formally, given a CSG $\mathcal{G}$, a strategy σ_i for a player i is said to be *k-bounded*, for a given $k \geq 0$, if $|Q_i| \leq |\mathrm{St}|^k$, that is, the number of states encoding σ_i is bounded from above by $|\mathrm{St}|^k$. This restriction can be thought of as a "bounded rationality" assumption. Finally, we consider the case of myopic strategies, which captures imperfect information in that any player cannot condition their behaviour on the actions taken by other players, but only on the history so far. More precisely, a strategy $\sigma_i = (Q_i, q_i^0, \delta_i, \tau_i)$ is *myopic* if its internal transition function can be defined as $\delta_i : Q_i \times \mathrm{St} \to Q_i$. With this, we let Σ_i^M denote the set of *memoryless strategies*, Σ_i^Y the set of *myopic strategies*, and Σ_i^k the set of *k-bounded strategies* for player i. For $X \in \{M, Y, k\}$, let $\Sigma^X = \prod_{i \in N} \Sigma_i^X$. We can define Nash equilibria with respect to each of these restricted strategy sets in the obvious manner.

EXAMPLE 1. *Consider a game $\mathcal{G}_1$ whose CSGA is depicted in Example 1. $\mathcal{G}_1$ consists of two players, where player 1's action set is $\{a,b,c\}$ and player 2's action set is $\{a,b\}$. Actions a and b are available to both players at all states, and action c is available to player 1 only at states s_1 and s_2. Now suppose that player 1's LTL goal is given by*

$$\gamma_1 = (\square\, (p \leftrightarrow q)) \vee (\diamond r),$$

and that player 2's LTL goal is given by

$$\gamma_2 = (\diamond\square(q \wedge \neg p)) \vee (\diamond\square(p \wedge \neg q)) \vee (\bigcirc r \wedge \square(p \leftrightarrow q)).$$

We observe that $\mathcal{G}_1$ has two memoryless Nash equilibria by considering cases for all possible moves at s_0. If (a, b) is played at s_0, then player 1 has a beneficial deviation to (b, b); this improves their odds of winning from 0 to $\frac{1}{3}$. If (b, a) is played, then player 1 has a beneficial deviation to (a, a). If (a, a) is played at s_0 and player 1 threatens to play c at s_1 and s_2, then player 1's expected utility is 1, player 2's expected utility is 0, and neither player has a beneficial deviation, so this is a Nash equilibrium. Similarly, the scenario where (b, b) is played at s_0 and player 1 plays c at both s_1 and s_2 is a Nash equilibrium, where player 1's expected utility is $\frac{1}{3}$, and player 2's expected utility is 0. Thus, we see that there are Nash equilibria for which players' goals may be satisfied qualitatively (almost surely) and others for which satisfaction is quantitative (with probability between 0 and 1). Moreover, permitting k-bounded strategies (with $k = 1$) allows for another Nash equilibrium $\vec{\sigma}$ in which both players' expected utilities are $\frac{1}{3}$. Under $\vec{\sigma}$, both players select (b, b) in the first round. This reaches s_1, s_2, s_3 each with probability $\frac{1}{3}$. From s_3, after returning to s_0, both players switch to playing (a, a) indefinitely. From s_1 and s_2, player 1 plays c from then onwards. This acts as a threat of punishment if player 2 ever decides to deviate. Such a strategy also constitutes a myopic Nash equilibrium, as the only information both players need is the history of states visited at any point in the game.

3 UNDECIDABILITY

We now present our main result of the paper: in the general case, the rational verification problem for sequential stochastic multiplayer games (SSGs) is undecidable. Chatterjee et. al. [12] showed that any SSG has a Nash equilibrium, and gave an algorithm for computing one. Indeed, their algorithm may compute an equilibrium where all players lose almost surely (i.e. receive expected payoff 0), while there exist other equilibria where all players win almost surely (i.e. receive expected payoff 1). When one considers "superior" Nash equilibria where one or more players win almost surely, the existence problem becomes undecidable [33] for SSGs. The undecidability result (Remark 4.11, [33]) with reachability objectives for SSGs required 9 players. A subsequent paper [15] claimed to improve this result to 5 players. However, this latter proof has some issues, which we discuss in the Appendix. We now show that undecidability can be achieved with 3 players.

Theorem 1. *E-Nash is undecidable in sequential stochastic games with 3 players.*

Proof. We show undecidability by (i) constructing an SSG $\mathcal{G}$ with 3 players and an LTL formula φ and (ii) reducing the non-halting problem of two counter machines to the existence of a Nash equilibrium $\vec{\sigma}$ such that $\vec{\sigma} \models \mathsf{AS}(\varphi)$ in $\mathcal{G}$.

A two-counter machine $\mathcal{M}$ consists of a sequence of instructions $\iota_0, \ldots, \iota_m$ where each instruction is one of the following: (i) "inc(j); goto k" (increment counter j by 1 and go to instruction k); (ii) "zero(j) ? goto k : dec(j); goto l" (if the value of counter j is zero, go to instruction k; otherwise, decrement counter j by one and go to instruction l); (iii) "halt" (Halt). Here j ranges over the counters $1, 2$, $k \neq l$ range over the instructions $0, \ldots, m$. A configuration of $\mathcal{M}$ is a triple $C = (i, c_1, c_2) \in \{0, \ldots, m\} \times \mathbb{N} \times \mathbb{N}$, where i is the current instruction and c_j is the current value of counter j. A configuration C' is the *successor* of configuration C, denoted by $C \vdash C'$, if it results from C by executing instruction ι_i; a configuration $C = (i, c_1, c_2)$ with ι_i = "halt" has no successor configuration. The *computation of* $\mathcal{M}$ is the unique maximal sequence $\rho = \rho(0)\rho(1)\ldots$ such that $\rho(0) \vdash \rho(1) \vdash \ldots$ and $\rho(0) = (0, 0, 0)$ (the *initial configuration*). Note that ρ is either infinite, or it ends in a configuration $C = (i, c_1, c_2)$ such that ι_i = "halt". The *halting problem* is to decide, given a machine $\mathcal{M}$, whether $\mathcal{M}$ reaches "halt". It is well-known that the *halting problem*, as well as its dual, whether "halt"will not be reached, are both undecidable [25].

Description of $\mathcal{G}$. Given an instance of a two counter machine with instructions $\iota_1, \ldots, \iota_m$, we construct a SSG $\mathcal{G}$ with 3 players A, B, and 0. This game is depicted in Figure 2 where the players A, B, and 0 are depicted as red, yellow, and green states. For any player $i \in \{A, B\}$ and state s, $\mathrm{Ac}_i(s) = \{co, ex\}$ represents choices to continue or exit the game. Likewise, at any player 0 state s, $\mathrm{Ac}_0(s)$= $\{next, zero, dec, prob, loop, stgt\}$. The actions $next, zero, dec$ represent proceeding with the next instruction and choosing a zero or decrement instruction in the gadgets $I^t_{i,\gamma}$, while the actions $loop, stgt$ represent the choice of taking the loop or going straight from the player 0 state labelled g in $Gad^t_{j,\gamma}$ gadget. Square states are terminal; that is, they have a self-loop, and these could be of any player. Let $AP = \{A, B, halt, a_t, b_t, p_t, q_t, g \mid t \in \{0, 1\}, j \in \{1, 2\}\}$. The labelling function λ is defined with respect to AP, and labels are shown adjacent to their corresponding state nodes in Figure 2. If no such label is present next to a state, its label is $\emptyset$.

If $\iota_1, \ldots, \iota_m \in \{\mathrm{inc}(j), \mathrm{dec}(j), \mathrm{zero}(j) \mid j{=}1, 2\}$, let $\Gamma = \{\mathrm{init}, halt\} \cup \{\iota_1, \ldots, \iota_m\}$. For each $i \in \{1, \ldots, m\}, \gamma \in \Gamma, j \in \{1, 2\}$ and $t \in \{0, 1\}$, the game $\mathcal{G}$ has gadgets $Exit, Main^t_{i,\gamma}, Gad^t_{j,\gamma}, I^t_{i,\gamma}$. We begin with the gadget $Main^0_{1,\mathrm{init}}$, with player A state v_0. $(\mathcal{G}, v_0)$ represents the initialised game starting at v_0.

LTL Goals for players. Define φ^t_A as $\Box(a_t \Rightarrow \Diamond p_t)$, for $t \in \{0, 1\}$. φ^t_A captures reaching terminal states marked p_t, given any history hv where v is labelled a_t (v is in the $Main^t$ gadget). Likewise, define φ^t_B to be $\Box(b_t \Rightarrow \Diamond q_t)$. The goals of players 0, A, and B respectively are $\gamma_0 = \Box\neg halt$, $\gamma_A = \bigwedge_{t \in \{0,1\}} \varphi^t_A$, and $\gamma_B = \bigwedge_{t \in \{0,1\}} \varphi^t_B$.

Let $\varphi = \Box\neg halt$. We answer E-Nash on the constructed SSG $\mathcal{G}$ with respect to the formula φ; is there is a Nash equilibrium $\vec{\sigma}$ such that $\vec{\sigma} \models \mathsf{AS}(\varphi)$? Note that $\vec{\sigma} \models \mathsf{AS}(\varphi)$ iff $\vec{\sigma}$ is a strategy profile where player 0 wins almost-surely.

Key Elements of the Reduction. Given a sequence of instructions $\gamma_0, \gamma_1, \ldots,$ of $\mathcal{M}$, let $v_0 \prec h_1 v_1 \prec h_2 v_2 \prec \ldots$, be the unique sequence of consecutive histories visited under a strategy profile $\vec{\sigma}$ such that v_k is the player A state labelled $a_{k \bmod 2}$ in a $Main^{k \bmod 2}_{i,\gamma_k}$ gadget corresponding to the instruction γ_k, and $h_1, h_2, \dots \in V^*$, such that $Pr(\mathcal{G}, \vec{\sigma}, \Diamond v_{k \bmod 2}) > 0$ for each $k \in \mathbb{N}$ (that is, we can reach $v_{k \bmod 2}$).

Define $a^n = Pr(\mathcal{G}, \vec{\sigma}, \varphi^{n \bmod 2}_A)$ as the expected utility under $\vec{\sigma}$ for satisfying $\varphi^{n \bmod 2}_A$ (that is, reaching a terminal state labelled $p_{n \bmod 2}$) given a history hv_n. Likewise, let $b^n = Pr(\mathcal{G}, \vec{\sigma}, \varphi^{n \bmod 2}_B)$ be the expected utility under $\vec{\sigma}$ for satisfying $\varphi^{n \bmod 2}_B$ (that is, reaching a terminal state labelled $q_{n \bmod 2}$) given a history hv_n. For $t{\in}\{0, 1\}$, define $\eta_t {=} \neg \bigcirc A \,\mathsf{U}\, p_t$ and $\varphi^{t,1-t}_A {=} \Box(a_t \Rightarrow \{\eta_t \vee [\neg \bigcirc A \,\mathsf{U}\, (a_{1-t} \wedge \eta_t)]\})$ for $t{\in}\{0, 1\}$. $\varphi^{t,1-t}_A$ captures the property of reaching a terminal state marked p_t, over the instructions γ_n, γ_{n+1}.

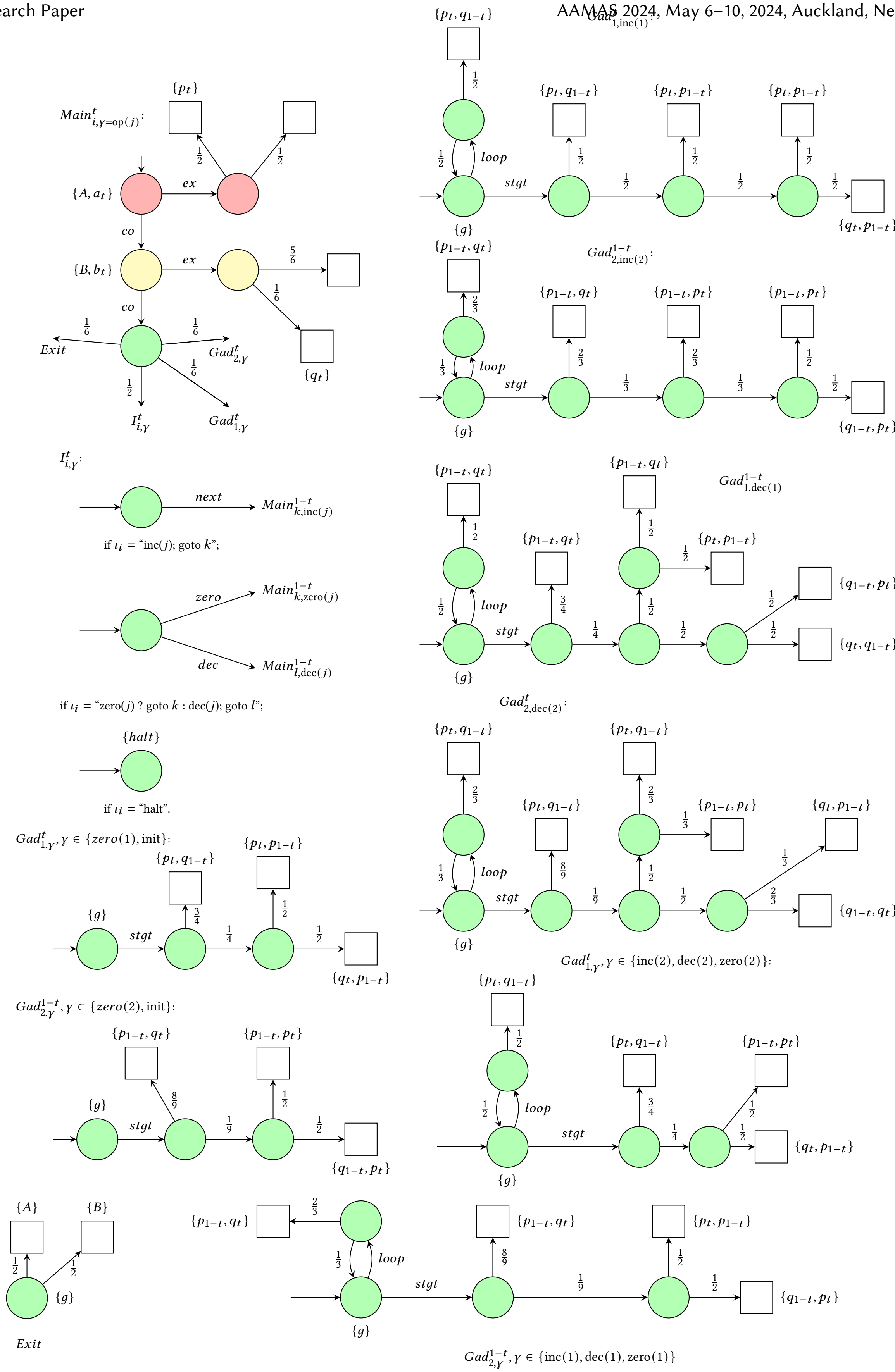

Figure 2: Simulating a two-counter machine.

Let $z^n = Pr(\mathcal{G}, \vec{\sigma}, \varphi_A^{n \bmod 2, n+1 \bmod 2})$ be the expected utility under $\vec{\sigma}$ for satisfying $\varphi_A^{n \bmod 2, n+1 \bmod 2}$ (that is, reaching a terminal state labelled $p_{n \bmod 2}$) given consecutive histories hv_n and $hv_n h' v_{n+1}$ where h' does not see any player A states. For each $j \in \{1, 2\}$ and $n \in \mathbb{N}$, define max_j^n to be the maximum number of times the loop involving the player 0 state labelled g_j is taken inside any of the gadgets $Gad_{j,\gamma_n}^{n \bmod 2}$. In the gadgets $Gad_{j,\gamma_n}^{n \bmod 2}$, $max_j^n = 0$ when $\gamma_n \in \{\text{zero}(j), \text{init}\}$.

Lemma 2. *Let $\vec{\sigma}$ be a strategy profile of $(\mathcal{G}, v_0)$ where player 0 wins almost surely. Then $\vec{\sigma}$ is a Nash equilibrium iff the following holds.*

$$max_j^{n+1} = \begin{cases} 1 + max_j^n & \text{if } \gamma_{n+1} = \text{inc}(j), \\ max_j^n - 1 & \text{if } \gamma_{n+1} = \text{dec}(j), \\ max_j^n = 0 & \text{if } \gamma_{n+1} = \text{zero}(j), \\ max_j^n & \text{otherwise} \end{cases} \quad (1)$$

for all $j \in \{1, 2\}$ and $n \in \mathbb{N}$.

Lemma 2 ensures a faithful simulation of the two counter machine: the number of times the loop is taken in gadget $Gad_{j,\gamma_{n+1}}^{n+1 \bmod 2}$ depends on the instruction γ_{n+1}, and the expected value of counter j after γ_{n+1}. Lemma 2 goes through many sublemmas. In the following, let $\vec{\sigma}$ be a strategy profile of $(\mathcal{G}, v_0)$ where player 0 wins almost surely. Lemma 3 proves that player A gets an expected utility of $\frac{1}{2}$ in a strategy where she continues the simulation (choosing *co*).

Lemma 3. *$\vec{\sigma}$ is a NE iff $a^n = \frac{1}{2}$ for all $n \in \mathbb{N}$.*

Lemma 4 is based on the observation that terminal states of all gadgets (other than *Exit*) are labelled either p_t or q_t. With Lemma 3, this gives $b^n = \frac{1}{6}$. Lemma 4 then follows based on the relationship between a^n and z^n. Recall that a^n denotes reaching $p_{n \bmod 2}$ over $\gamma_n, \gamma_{n+1}, \ldots$, while z^n represents reaching it over γ_n, γ_{n+1}. Terminal states labelled $p_{n \bmod 2}$ are reachable from the gadgets $Gad_{1,\gamma}^t, Gad_{2,\gamma}^t, Gad_{1,\gamma}^{1-t}, Gad_{2,\gamma}^{1-t}$ for $t = n \bmod 2$. From the player 0 state in $Main_{i,\gamma}^t$, $Gad_{1,\gamma}^t, Gad_{2,\gamma}^t$ are reachable with probability $\frac{2}{6}$. Hence, $a^n = z^n + \frac{2}{6} a^{n+2}$.

Lemma 4. *$a^n = \frac{1}{2}$ iff $z^n = \frac{1}{3}$ for all $n \in \mathbb{N}$.*

Using these lemmas, it suffices to show that $z^n = \frac{1}{3}$ iff (1) holds. This also proves Lemma 2. As observed already, z^n can be expressed as the sum of the payoffs of reaching a terminal node labelled $p_{n \bmod 2}$ in the gadgets *Exit*, $Gad_{j,\gamma_n}^{n \bmod 2}$ (call this α_n^j) as well as the gadgets $Gad_{j,\gamma_{n+1}}^{n+1 \bmod 2}$ (call this α_{n+1}^j), $j \in \{1, 2\}$. The proof of Lemma 2 is done by a case analysis of all possibilities for γ_n, γ_{n+1} and proving in each case, that $z^n = \frac{1}{3}$ iff (1).

The proof is completed by showing that $\mathcal{M}$ has an infinite computation iff $(\mathcal{G}, v_0)$ has a Nash equilibrium where player 0 wins almost surely. Note that pure strategies suffice: for players A, B, this is simply choosing the action *co*. For player 0, corresponding to each history ending in a gadget $I_{i,\gamma}^t$ for $t \in \{0, 1\}$, after visiting a player 0 state in the *Main* gadgets n times, the strategy is to play to reach the gadget $Main_{k,\gamma'}^{1-t}$ such that the configuration $\rho(n)$ of $\mathcal{M}$ corresponds to the instruction ι_k; this is possible if the configuration $\rho(n-1)$ of $\mathcal{M}$ corresponds to instruction ι_i. For a history reaching a gadget $Gad_{j,\gamma'}^t$, $t \in \{0, 1\}$, after visiting player A state in the *Main* gadget n times, player 0 chooses the action *loop* m times from the state labelled g iff m is the value of counter j in configuration $\rho(n-1)$. □

Remark. Note that the LTL objectives of the 3 players A, B, and 0 use only unary LTL (i.e., where the full power of "until" is not needed). The formulae $\eta_t, \varphi_A^{t,1-t}$ are used only to specify the payoff z^n precisely. As described in the construction, given a history hv_n, z^n is the accumulated payoff of reaching $p_{n \bmod 2}$ in the gadgets $Gad_{j,\gamma_n}^{n \bmod 2}, Gad_{j,\gamma_{n+1}}^{n+1 \bmod 2}$ after the instructions γ_n, γ_{n+1} for $j = 1, 2$.

4 SPECIAL CASES OF E-NASH

In this section, we study the E-Nash problem under three natural restrictions on strategies: memoryless, k-bounded, and myopic strategies. Such strategies may be more appropriate when consdering games in which players' goals do not use the full expressivity of LTL, such as safety objectives. In the case of k-bounded strategies, decision problems are specified with k given as an input to the problem (and for simplicity, we assume that k is given in unary encoding).

Theorem 5. *E-Nash with memoryless and k-bounded strategies is PSPACE-complete.*

Proof. For the upper bound, Algorithm 1 can be used to decide E-Nash. First note that guessing a memoryless (k-bounded) strategy profile $\vec{\sigma} \in \Sigma^M (\Sigma^k)$ can be done in NP. Secondly, checking whether $\vec{\sigma} \models \mathsf{AS}(\varphi)$ can be done in PSPACE [14]. Finally, checking whether $\vec{\sigma} \in \text{NE}(\mathcal{G})$ can be done in PSPACE according to Theorem 7. Thus, the overall procedure lies in PSPACE.

For hardness, we reduce from the universality problem for LTL properties on Markov chains (MCs), which is known to be PSPACE-complete [34]. The problem is to decide, given a finite Markov chain $C = (S, s_\iota, \text{tr}, \lambda)$ and an LTL formula φ, whether $C \models \mathsf{AS}(\varphi)$, that is, whether the MC satisfies φ with probability equal to 1. Clearly, this problem can be encoded in a CSG $\mathcal{G}$ with the same state space, initial state, and labelling function as C. Additionally $\mathcal{G}$ has a single player whose goal is $\gamma_1 = \top$, and whose action set consists of a single action at each state which induces the same transition probabilities as those in C. The answer to E-Nash with φ as the global formula will be "yes" iff the answer to the universality problem is "yes", as a witness Nash equilibrium strategy would not require any memory. This also therefore shows establishes the lower bound for E-Nash with k-bounded strategies. □

Theorem 6. *E-Nash with myopic strategies and 3 players is undecidable.*

Algorithm 1 Memoryless/k-bounded E-Nash

Input: Game $\mathcal{G}$, LTL formula φ

```
Guess NE σ⃗ ∈ Σ^M (or Σ^k)
if σ⃗ ⊨ AS(φ) and σ⃗ ∈ NE(G) then
    return "Yes"
end if
return "No"
```

Proof. Observe that the same reduction used in Theorem 1 can be applied to this setting, because the CSG $\mathcal{G}$ constructed to simulate a two-counter machine does not require any player to observe the actions taken by other players. In particular, there is a non-halting computation of $\mathcal{M}$ if and only if there is a *myopic* Nash equilibrium in the CSG $\mathcal{G}$. □

We can also ask the "universal" counterpart of E-Nash, known as A-Nash, a problem with the same input as E-Nash but where we ask whether the input LTL formula φ is satisfied almost surely on *every* Nash equilibrium strategy profile of the input game. With the results above, the complexity of A-Nash naturally follows, as this can be solved by simply checking a modified version of E-Nash, such that the answer to this modified version is "yes" iff the LTL formula is satisfied with *non-zero probability*, as opposed to being satisfied almost surely. Then, we can run this modified version of E-Nash on $\neg\varphi$, where φ is the LTL formula to be checked in the instance of A-Nash – there is a Nash equilibrium that satisfies $\neg\varphi$ with non-zero probability if and only if the answer to A-Nash is "no". For more details, see Section 3 of [18].

5 MEMBERSHIP

Recall that the Membership problem asks whether a given strategy profile forms a Nash equilibrium. We now study the Membership problem, and show that it is decidable for all the types of strategies we have consider in this paper.

Theorem 7. *Membership with memoryless and k-bounded strategies is PSPACE-complete.*

Proof. Algorithm 2 decides Membership for these two cases. This follows from the fact that the algorithm returns "no" if and only if there exists some player $i \in \mathrm{N}$ and alternative strategy σ_i' that improves the player's probability of satisfying their goal under a unilateral definition. This is precisely the definition of a Nash Equilibrium; hence, the algorithm decides Membership. For the complexity of the algorithm, note first that computing $p_i = \Pr(\mathcal{G}, \vec{\sigma}, \gamma_i)$ in line 2 of Algorithm 2 can be done in time that is polynomial with respect to the CSGA $\mathcal{A}$ and exponential with respect to the goals γ_i, for each $i \in \mathrm{N}$ (see [14] for more details). However, the construction in [14] uses at most polynomial space and thus, computing this probability can be done in polynomial space. Secondly, observe that guessing a memoryless strategy σ_i' (line 3 of the algorithm) can be done in NPSPACE. Next, computing the probability $\Pr(\mathcal{G}, (\vec{\sigma}_{-i}, \sigma_i'), \gamma_i)$ line 4 of Algorithm 2 can again be done in polynomial space as with line 2. Thus, the overall procedure lies in PSPACE.

For hardness, we can use a similar reduction to the one from Theorem 5, that is, we reduce from the universality problem for LTL properties on finite Markov chains. However, we modify $\mathcal{G}$ by adding two states s_0', s_∞ to the game so that it starts at s_0' and there are two actions $\{a, b\}$ available to player 1 at state s_0'. If player 1 chooses action a in this state, then the game transitions to s_∞ and remains there forever. If they choose action b on the other hand, then the game transitions into the state s_ι in $\mathcal{G}$ corresponding to the initial state from the MC C, and the game proceeds as previously outlined. Let $\lambda(s_0') = \emptyset$ and $\lambda(s_\infty) = \{q\}$, where q is a new propositional variable. Finally, let player 1's LTL goal be given by $\gamma_1 = q \vee \varphi$ and let this modified game be denoted by $\mathcal{G}'$. Given this construction, any memoryless strategy σ_1 in $\mathcal{G}'$ beginning with player 1 choosing b is a memoryless NE if and only if φ is satisfied in C with probability 1. To see why, observe that if $\vec{\sigma}$ is a memoryless NE, then player 1 does not benefit from switching to action a in the first round, which would guarantee the satisfaction of their goal almost surely. Thus, by choosing to simulate the MC, it must be the case that C satisfies γ_1 (and hence φ) with probability 1. Similarly, if C does *not* satisfy φ almost surely, then player 1 has a beneficial deviation from σ_1 by playing a in the first round. Thus, σ_1 is a Nash equilibrium in $\mathcal{G}'$ if and only if $C \models \mathsf{AS}(\varphi)$. □

Theorem 8. *Membership with myopic and general (finite-memory) strategies is 2EXPTIME-complete.*

Proof. We use Algorithm 2 again. The time complexity of line 2 is exponential in the size of the MC induced by $\vec{\sigma}$ on $\mathcal{G}$. In line 3, we construct the MDP $\mathcal{G}_{\vec{\sigma}_{-i}}$ and find a σ_i' that maximises the probability of satisfying γ_i in $\mathcal{G}_{\vec{\sigma}_{-i}}$; this is done in time that is polynomial in the size of the induced MDP and in doubly exponential time wrt the size of the LTL goal of i [14]. Finally, line 4 can be checked in exponential time as with line 2. For hardness, we reduce from the qualitative model-checking problem for Σ_1^{QLTL} formulae in MDPs, which is known to be 2EXPTIME-complete [28]. *QLTL* is an ω-regular extension of LTL, which allows one to quantify over propositional variables and Σ_1^{QLTL} is the set of *QLTL* formulae with exactly one quantifier, which must be existential. The qualitative model-checking problem for Σ_1^{QLTL} formulae in MDPs that we will work with asks whether, given an MDP $\mathcal{M}$ and a Σ_1^{QLTL} formula Ψ, it is the case that the maximum probability of satisfying Φ in the MDP's initial state s^0 is non-zero.

Given an MDP $\mathcal{M} = (\{1\}, \mathrm{St}, s^0, \mathrm{Ac}_1, \mathrm{tr}, \lambda)$ (i.e., a labelled CSGA with 1 player) and a Σ_1^{QLTL} formula $\Psi = \exists x.\varphi$, where φ is an LTL formula, we construct a CSG $\mathcal{G} = (\{1\}, \mathrm{St}', s^{0'}, \mathrm{Ac}_1', \mathrm{tr}', \gamma_1, \lambda')$ with one player as follows: firstly, we clone the entire MDP structure and assign the label x to all states in this cloned structure. Thus, the set of states St' in $\mathcal{G}$ will consist of two copies of the set of states St in the MDP. For every state $s \in \mathrm{St}$, let s' denote its clone, so that both $s, s' \in \mathrm{St}'$. Secondly, for every state-action pair $(s, a) \in \mathrm{St} \times \mathrm{Ac}_1$, we create another state-action pair (s, a^c) that mirrors the transition probabilities of (s, a), except that the transitions go to the corresponding state in $\mathcal{M}$. More precisely, the transition function tr' of $\mathcal{G}$ is s.t. for all pairs of states $s, t \in \mathrm{St}$ and all actions $a \in \mathrm{Ac}$,

$$\mathrm{tr}'(s,a)(t) = \mathrm{tr}'(s,a^c)(t') = \mathrm{tr}'(s',a^c)(t') = \mathrm{tr}'(s',a)(t) = \mathrm{tr}(s,a)(t);$$

$$\mathrm{tr}'(s,a)(t') = \mathrm{tr}'(s,a^c)(t) = \mathrm{tr}'(s',a^c)(t) = \mathrm{tr}'(s',a)(t') = 0,$$

where $\mathrm{tr}'(s,a)(t)$ denotes the probability assigned to the transition from s to t under $\mathrm{tr}'(s,a)$. Then, we add a new initial state $s^{0'}$ and a new sink state s^∞ such that $\lambda(s^{0'}) = \emptyset$ and $\lambda(s^\infty) = \{y\}$, where y is a new propositional variable. The player has two possible transitions with probability one from $s^{0'}$: either transition to s^0 from the original MDP, or transition to s^∞. The player's goal is $\gamma_1 = (\Box\neg y) \wedge (\bigcirc\varphi)$. The labelling function λ' is such that for all states $s \in \mathrm{St}$, we have $\lambda'(s) = \lambda(s)$ and $\lambda'(s') = \lambda(s) \cup \{x\}$. Finally, we check whether the strategy profile that transitions from $s^{0'}$ to s^∞ and remains there forever is a NE. If it is, then the player has no strategy that could make the probability of satisfying γ_1 positive. If not, then there is a myopic strategy for the player such that γ_1 is satisfied with positive probability. This is true iff the QLTL formula Ψ is satisfied with positive probability on the MDP $\mathcal{M}$. □

Remark. Note that the best response is guessed nondeterministically and does not return a single strategy as such. We chose to represent line 3 in Algorithm 2 as computing a Best-Response and refer to the same in Theorems 7,8, though their actual implementations are different for these two cases. To convert this into a deterministic algorithm in the case of Theorem 7, we can replace line 3 with a loop over all strategies in the class under consideration (i.e., memoryless or k-bounded).

6 CONCLUSIONS AND RELATED WORK

As indicated in [10, 32, 33], most results for the complexity of Nash equilibria in n-player, general-sum, concurrent stochastic games are either unknown or undecidable, unless at least one of the restrictions is dropped – that is, unless the games are 2-player [11], or zero-sum, or sequential [12], and even in some of these cases, further restrictions must be imposed – typically on the model of strategies [33], the type of goals [8, 9], or the precision on Nash equilibrium computation – in order to obtain decidability. Our results show that rational verification is consistent with these results, particularly for 3-player, general-sum, stochastic games. Our results are optimal in various ways: the problem is decidable if we allow games to be either n-player and deterministic, or 2-player and zero-sum. Thus, an undecidability result for 3-player SSGs that works for myopic strategies is hard to improve.

Computing the Nash equilibria of n-player, general-sum, perfect information, deterministic concurrent games has been studied under a variety of player objectives such as LTL [23], ω-regular [5], and limit-average [32] (see also [7] for a study of sequential deterministic games). These problems become undecidable when considering imperfect information and LTL or ω-regular goals [16, 24]. In the LTL setting with reactive module games [24], decidability is recovered in EXPSPACE/NEXPTIME/PSPACE for myopic/memoryless/k-bounded strategies; with LTL goals, two-player games are also decidable.

A quantitative extension of strategy logic is $SL[\mathcal{F}]$, which allows one to express concepts such as the existence of a Nash equilibrium in concurrent game structures [6, 26]. However, $SL[\mathcal{F}]$ is defined over structures having a *deterministic* transition function. Thus, $SL[\mathcal{F}]$ cannot be used to represent solution concepts in our model.

Algorithm 2 MEMBERSHIP

Input: Game $\mathcal{G}$, strategy profile $\vec{\sigma}$

```
1: for i ∈ N do
2:     Compute p_i = Pr(G, σ⃗, γ_i)
3:     Compute σ'_i = Best-Response(G, σ⃗, i)
4:     if Pr(G, (σ⃗_{-i}, σ'_i), γ_i) > p_i then
5:         return "no"
6:     end if
7: end for
8: return "yes"
```

Probabilistic Strategy Logic (PSL) allows one to reason over CSGs and can hence be used to express our E-NASH problem [3]. However, the undecidability of model checking PSL cannot be used to prove the undecidability of E-NASH, so our main result establishes this for a natural special case of PSL model checking where players have LTL goals. Furthermore, our PSPACE-completeness result for E-NASH with memoryless strategies significantly improves the naive upper-bound of 3EXPSPACE, which one would obtain by encoding the problem into PSL restricted to memoryless strategies.

A key aspect of our study is the focus on pure strategies. This is motivated by previous results on stochastic games, in which Nash equilibrium is known to be undecidable for n-player stochastic games, unless extremely simple goal types are considered. Deterministic or finite strategies may not be powerful enough to achieve optimal behaviour (especially with a small number of players). [33] shows that considering finite strategies with 14 players gives undecidability in SSGs. As with our improvement on the 9-player Pure NE undecidability of [33], we conjecture that an improvement can be made from 14 players to a smaller number under finite strategies.

A final distinctive feature of our study is the use of combined techniques for qualitative and quantitative probabilistic reasoning. The *probabilistically quantitative* part of our model assumes that players seek, *locally and individually*, to maximise the probability of achieving their LTL goals. On the other hand, the *probabilistically qualitative* part is used to check if a *global* LTL *specification* is almost-surely satisfied on some Nash equilibrium. This is in contrast to [18] where only qualitative probabilistic behaviour is considered on both the goals and global specification. Other ways of combining qualitative and quantitative probabilistic reasoning can be found elsewhere, *e.g.*, sometimes using a lexicographic order of qualitative and quantitative preferences [21], and sometimes by associating a real value to the probabilistic satisfaction of a temporal logic formula [2]. All these types of reasoning are different from the one we propose to use here.

As automated AI systems become more pervasive, ensuring their reliability and safety will become increasingly important through formal verification techniques [31]. This study opens up several important avenues for future research. Our results characterise the computational complexity of rational verification problems for multi-agent systems under different strategy models. While some cases remain undecidable, identifying decidable strategy restrictions and objectives in increasingly realistic game models sheds light on how we might design autonomous systems to ensure their verifiability. By elucidating the boundaries of decidability in game-theoretic verification, this research helps to bridge the gap between principled verification methods and increasingly capable real-world AI systems by better understanding what models of game structures, player strategies, and objectives are amenable to automated verification, particularly in the context of multi-agent interactions.

ACKNOWLEDGMENTS

Michael Wooldridge was supported by the UKRI under a Turing AI World Leading Research Fellowship (EP/W002949/1). Julian Gutierrez was supported by the DARPA Assured Neuro Symbolic Learning and Reasoning (ANSR) program under award number FA8750-23-2-1016.

REFERENCES

[1] Alessandro Abate, Julian Gutierrez, Lewis Hammond, Paul Harrenstein, Marta Kwiatkowska, Muhammad Najib, Giuseppe Perelli, Thomas Steeples, and Michael J. Wooldridge. 2021. 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

[2] Shaull Almagor, Udi Boker, and Orna Kupferman. 2016. Formally Reasoning About Quality. *J. ACM* 63, 3 (2016), 24:1–24:56. https://doi.org/10.1145/2875421

[3] Benjamin Aminof, Marta Kwiatkowska, Bastien Maubert, Aniello Murano, and Sasha Rubin. 2019. Probabilistic strategy logic. *Artificial Intelligence* (2019).

[4] C. Baier and J.-P. Katoen. 2008. *Principles of Model Checking*. The MIT Press: Cambridge, MA.

[5] Patricia Bouyer, Romain Brenguier, Nicolas Markey, and Michael Ummels. 2015. Pure Nash Equilibria in Concurrent Deterministic Games. *Log. Methods Comput. Sci.* 11, 2 (2015). https://doi.org/10.2168/LMCS-11(2:9)2015

[6] Patricia Bouyer, Orna Kupferman, Nicolas Markey, Bastien Maubert, Aniello Murano, and Giuseppe Perelli. 2023. Reasoning about Quality and Fuzziness of Strategic Behaviors. *ACM Trans. Comput. Logic* 24, 3, Article 21 (apr 2023), 38 pages. https://doi.org/10.1145/3582498

[7] Léonard Brice, Jean-François Raskin, and Marie van den Bogaard. 2023. Rational verification for nash and subgame-perfect equilibria in graph games. In *48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[8] Krishnendu Chatterjee. 2012. The complexity of stochastic Müller games. *Information and Computation* 211 (2012), 29–48.

[9] Krishnendu Chatterjee, Luca De Alfaro, and Thomas A Henzinger. 2005. The complexity of stochastic Rabin and Streett games. In *International Colloquium on Automata, Languages, and Programming*. Springer, 878–890.

[10] K. Chatterjee and T. A. Henzinger. 2012. A survey of stochastic omega-regular games. *Journal Of Computer And System Sciences* 78 (2012), 394–413.

[11] Krishnendu Chatterjee, Marcin Jurdzinski, and Thomas A. Henzinger. 2004. Quantitative stochastic parity games. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, New Orleans, Louisiana, USA, January 11-14, 2004*, J. Ian Munro (Ed.). SIAM, 121–130. http://dl.acm.org/citation.cfm?id=982792.982808

[12] Krishnendu Chatterjee, Rupak Majumdar, and Marcin Jurdzinski. 2004. On Nash Equilibria in Stochastic Games. In *Computer Science Logic, 18th International Workshop, CSL 2004, 13th Annual Conference of the EACSL, Karpacz, Poland, September 20-24, 2004, Proceedings (Lecture Notes in Computer Science, Vol. 3210)*, Jerzy Marcinkowski and Andrzej Tarlecki (Eds.). Springer, 26–40. https://doi.org/10.1007/978-3-540-30124-0_6

[13] E. M. Clarke, O. Grumberg, and D. A. Peled. 2000. *Model Checking*. The MIT Press: Cambridge, MA.

[14] Costas Courcoubetis and Mihalis Yannakakis. 1995. The complexity of probabilistic verification. *Journal of the ACM (JACM)* 42, 4 (1995), 857–907.

[15] Ankush Das, Shankara Narayanan Krishna, Lakshmi Manasa, Ashutosh Trivedi, and Dominik Wojtczak. 2015. On Pure Nash Equilibria in Stochastic Games. In *Theory and Applications of Models of Computation - 12th Annual Conference, TAMC 2015, Singapore, May 18-20, 2015, Proceedings (Lecture Notes in Computer Science, Vol. 9076)*, Rahul Jain, Sanjay Jain, and Frank Stephan (Eds.). Springer, 359–371. https://doi.org/10.1007/978-3-319-17142-5_31

[16] Emmanuel Filiot, Raffaella Gentilini, and Jean-François Raskin. 2018. Rational Synthesis Under Imperfect Information. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*. ACM. https://doi.org/10.1145/3209108.3209164

[17] D. Fisman, O. Kupferman, and Y. Lustig. 2010. Rational Synthesis. In *TACAS (LNCS, Vol. 6015)*. Springer, 190–204.

[18] Julian Gutierrez, Lewis Hammond, Anthony W. Lin, Muhammad Najib, and Michael J. Wooldridge. 2021. Rational Verification for Probabilistic Systems. In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning, KR 2021, Online event, November 3-12, 2021*, Meghyn Bienvenu, Gerhard Lakemeyer, and Esra Erdem (Eds.). 312–322. https://doi.org/10.24963/kr.2021/30

[19] J. Gutierrez, P. Harrenstein, and M. Wooldridge. 2015. Iterated Boolean Games. *Information and Computation* 242 (2015), 53–79.

[20] Julian Gutierrez, Szymon Kowara, Sarit Kraus, Thomas Steeples, and Michael J. Wooldridge. 2023. Cooperative concurrent games. *Artif. Intell.* 314 (2023), 103806. https://doi.org/10.1016/j.artint.2022.103806

[21] Julian Gutierrez, Aniello Murano, Giuseppe Perelli, Sasha Rubin, Thomas Steeples, and Michael Wooldridge. 2020. Equilibria for games with combined qualitative and quantitative objectives. *Acta Informatica* (2020), 1–26. Publisher: Springer.

[22] Julian Gutierrez, Muhammad Najib, Giuseppe Perelli, and Michael J. Wooldridge. 2019. On Computational Tractability for Rational Verification. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 329–335. https://doi.org/10.24963/ijcai.2019/47

[23] Julian Gutierrez, Muhammad Najib, Giuseppe Perelli, and Michael J. Wooldridge. 2020. Automated temporal equilibrium analysis: Verification and synthesis of multi-player games. *Artif. Intell.* 287 (2020), 103353. https://doi.org/10.1016/j.artint.2020.103353

[24] Julian Gutierrez, Giuseppe Perelli, and Michael Wooldridge. 2018. Imperfect information in Reactive Modules games. *Information and Computation* 261 (2018), 650 – 675. https://doi.org/10.1016/j.ic.2018.02.023 4th International Workshop on Strategic Reasoning (SR 2016).

[25] Marvin Lee Minsky. 1967. *Computation*. Prentice-Hall Englewood Cliffs.

[26] Munyque Mittelmann, Bastien Maubert, Aniello Murano, and Laurent Perrussel. 2022. Automated synthesis of mechanisms. In *31st International Joint Conference on Artificial Intelligence (IJCAI-22)*. International Joint Conferences on Artificial Intelligence Organization, 426–432.

[27] Martin J Osborne and Ariel Rubinstein. 1994. *A course in game theory*. MIT press.

[28] Jakob Piribauer, Christel Baier, Nathalie Bertrand, and Ocan Sankur. 2021. Quantified Linear Temporal Logic over Probabilistic Systems with an Application to Vacuity Checking. In *CONCUR 2021-32nd International Conference on Concurrency Theory*.

[29] Amir Pnueli. 1977. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*. ieee, 46–57.

[30] A. Pnueli and R. Rosner. 1989. On the Synthesis of an Asynchronous Reactive Module. In *Proceedings of the Sixteenth International Colloquium on Automata, Languages, and Programs*.

[31] Max Tegmark and Steve Omohundro. 2023. Provably safe systems: the only path to controllable AGI. *arXiv preprint arXiv:2309.01933* (2023).

[32] Michael Ummels and Dominik Wojtczak. 2011. The Complexity of Nash Equilibria in Limit-Average Games. *CoRR* abs/1109.6220 (2011). http://arxiv.org/abs/1109.6220

[33] Michael Ummels and Dominik Wojtczak. 2011. The Complexity of Nash Equilibria in Stochastic Multiplayer Games. *Log. Methods Comput. Sci.* 7, 3 (2011). https://doi.org/10.2168/LMCS-7(3:20)2011

[34] Moshe Y Vardi. 1985. Automatic verification of probabilistic concurrent finite state programs. In *26th Annual Symposium on Foundations of Computer Science (SFCS 1985)*. IEEE, 327–338.

[35] Michael J. Wooldridge, Julian Gutierrez, Paul Harrenstein, Enrico Marchioni, Giuseppe Perelli, and Alexis Toumi. 2016. Rational Verification: From Model Checking to Equilibrium Checking. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, Dale Schuurmans and Michael P. Wellman (Eds.). AAAI Press, 4184–4191. http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12268