Decidable Model Checking with Uniform Strategies

Natasha Alechina
University of Nottingham
Nottingham, UK
nza@cs.nott.ac.uk

Mehdi Dastani
University of Utrecht
Utrecht, The Netherlands
M.M.Dastani@uu.nl

Brian Logan
University of Nottingham
Nottingham, UK
bsl@cs.nott.ac.uk

ABSTRACT

The logic of strategic ability Resource-Bounded Alternating Time Syntactic Epistemic Logic (RB±ATSEL) has a decidable model-checking problem for coalition uniform strategies. A strategy is coalition uniform if agents in a coalition select the same joint action in all states where the knowledge of the coalition is the same. However, this presupposes free and unbounded communication between the agents in the coalition before every action selection. In this paper we present a modified version of RB±ATSEL, RB±ATSEL$_c$, with explicit (and explicitly costed) communication actions. RB±ATSEL$_c$ is interpreted on communication models which have an explicit communication step before every action selection. We show that, unlike standard ATL under imperfect information, the model checking problem for RB±ATSEL$_c$ is decidable under perfect recall uniform strategies. Our decidability result also applies to ATL with imperfect information and perfect recall when interpreted on communication models.

ACM Reference Format:
Natasha Alechina, Mehdi Dastani, and Brian Logan. 2019. Decidable Model Checking with Uniform Strategies. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 9 pages.

1 INTRODUCTION

We consider the problem of verifying the existence of uniform strategies in multi-agent systems where the agents act under resource constraints and imperfect information. *Uniform strategies* are strategies where agents select the same actions in all states where they have the same information available to them. Uniform strategies are important because real agents can only select actions based on the information they have. However the model-checking problem for Alternating-Time Temporal Logic (ATL) under imperfect information and with uniform perfect recall strategies is known to be undecidable [9–11].

In previous work [3], we introduced Resource-Bounded Alternating Time Syntactic Epistemic Logic (RB±ATSEL). RB±ATSEL has a decidable model-checking problem for so-called coalition-uniform strategies, and an algorithm for verifying existence of coalition-uniform (rather than uniform) strategies for RB±ATSEL is given in [3]. A strategy is *coalition-uniform* if agents in a coalition select the same joint action in all states where the *knowledge of the coalition* is the same. The decidability result holds for any notion of coalition knowledge, where coalition-indistinguishability between states is decidable. In particular, the result in [3] applies for uniformity with respect to the distributed knowledge of a coalition, which is the notion we use in this paper.

However, the notion that a coalition can select actions based on, say, its distributed knowledge, presupposes free and unbounded communication between the agents in the coalition before every action selection. This rather goes against the grain of the resource-bounded setting in [3]. In this paper, we present a variant of RB±ATSEL, RB±ATSEL$_c$, with explicit communication strategies and interpreted on *communication models* that have an explicit (and explicitly costed) communication step before each action selection. We show that RB±ATSEL$_c$ model checking is decidable for uniform strategies. Our decidability result also applies to ATL$_{iR}$ (ATL with imperfect information and perfect recall) when interpreted on communication models. ATL is a sublogic of RB±ATSEL$_c$ that can be obtained by considering only infinite resource bounds and omitting the knowledge modality. While the decidability of the model-checking problem for ATL$_{iR}$ for strategies uniform with respect to distributed knowledge is known [9, 11], to the best of our knowledge, there has been no work on showing how to convert such strategies to uniform strategies while preserving decidability.

The remainder of the paper is organised as follows. In Section 2 we introduce the syntax and semantics of RB±ATSEL$_c$, and in Section 3 we illustrate RB±ATSEL$_c$ using a simple example. In Section 4 we show that the model checking problem for RB±ATSEL$_c$ is decidable by providing correct and terminating model-checking algorithms. In Section 5 we present an approach to generating less costly bounded communication strategies. In Section 6, we survey related work, focussing on other approaches to making ATL$_{iR}$ model checking decidable, and conclude in Section 7.

2 SYNTAX AND SEMANTICS OF RB±ATSEL$_c$

In this section, we introduce Resource-Bounded Alternating Time Syntactic Epistemic Logic With Communication, RB±ATSEL$_c$.

The language of RB±ATSEL$_c$ is parameterised by a finite set $Agt = \{a_1, \ldots, a_n\}$ of n agents, a finite set $Res = \{res_1, \ldots, res_r\}$ of r resources, and a finite set Π of propositional variables. The set of possible resource bounds or resource allocations is $B = \mathbb{N}_\infty^{Res^{Agt}}$, where $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$ (B consists of tuples of resource values of length r, one for each agent in Agt), and we denote by B_A the set of projections of tuples in B on a set of agents (coalition) $A \subseteq Agt$.

Formulas of the language $\mathcal{L}$ of RB±ATSEL$_c$ are defined by the following syntax

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \psi \mid \langle\langle A^b \rangle\rangle \bigcirc \varphi \mid \langle\langle A^b \rangle\rangle \varphi \mathcal{U} \psi \mid \langle\langle A^b \rangle\rangle \Box \varphi \mid K_a \varphi$$

where $p \in \Pi$ is a proposition, $A \subseteq Agt$, $b \in B_A$ is a resource bound, and $a \in Agt$. We denote by $\mathcal{L}^{EL}$ the language without coalition modalities.

$\langle\langle A^b \rangle\rangle \bigcirc \varphi$ means that coalition A has a strategy executable within resource bound b to ensure that the next state satisfies φ;

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

$\langle\langle A^b \rangle\rangle \varphi \,\mathcal{U}\, \psi$ means that A has a strategy executable within resource bound b to ensure ψ while maintaining the truth of φ; $\langle\langle A^b \rangle\rangle \Box \varphi$ means that A has a strategy executable within resource bound b to ensure that φ is always true, and $K_a \varphi$ is interpreted relative to the contents of agent a's knowledge base. a's knowledge base s_a is a finite set of formulas of $\mathcal{L}^{EL}$, all of the form $K_a \psi$ or $K_b \psi$, where $b \neq a$. Note that this definition of the agent's knowledge base is different from the one in [3], where it was assumed to be an arbitrary finite set of formulas, not necessarily prefixed by knowledge modalities. We refer to the subset of s_a containing only formulas of the form $K_a \psi$ as $k(s_a) = \{K_a \psi \mid K_a \psi \in s_a\}$. The formulas in $k(s_a)$ represent the knowledge of a 'proper'; the rest of the knowledge base is a's representation of the knowledge of other agents obtained by communication. If a formula of the form $K_a \varphi$ is in $k(s_a)$, then a knows that φ (and the formula $K_a \varphi$ in the language of $\mathcal{L}$ of RB$\pm$ATSEL$_c$ is true in the corresponding state). Formulas of the form $K_a K_b \psi$ in the language of $\mathcal{L}$ of RB$\pm$ATSEL$_c$ are true if $K_b \psi$ is in $s_a \setminus k(s_a)$. Note that this is a syntactic definition of knowledge. Representing knowledge this way allows us to model updating knowledge in a more straightforward way than for standard epistemic logic, and fits well with communicating syntactic objects (formulas). In the interests of brevity, we do not introduce any procedure for closing states under inference or checking for and restoring consistency here. We sketch an example of such a procedure in Section 5.

Definition 2.1. A communication model of RB$\pm$ATSEL$_c$ is a tuple $M = (\Phi, Agt, Res, \Pi, S, Act, d, c, \delta)$ where:

- Φ is a finite set of formulas of $\mathcal{L}^{EL}$ (possible contents of the local states of the agents) of the form $K_a \phi$, $a \in Agt$.
- Agt is a non-empty set of n agents, Res is a non-empty set of r resources.
- Π is a finite set of propositional variables.
- $S = S_{com} \cup S_{act}$ where S_{com} and S_{act} are disjoint non-empty sets of tuples $(s_{a_1}, \ldots, s_{a_n}, s_e)$ where $s_e \subseteq \Pi$ (s_e is the state of the environment) and for each $a \in Agt$, $s_a \subseteq \Phi \cup \{p_{act}\}$. For every $a \in Agt$ and $q \in S_{act}$, $p_{act} \in q_a$, and for every $s \in S_{com}$, $p_{act} \notin s_a$.
- $Act = Act' \cup CA$ is the union of two disjoint non-empty sets of actions. Act' is a set of ontic (non-communication) actions which contains a special action $idle$. $CA = \{com(k(s_a), A) \mid a \in A \subseteq Agt\}$, where $k(s_a) = \{K_a \psi \mid K_a \psi \in s_a\}$, is a set of communication actions.
- $d : S \times Agt \to \wp(Act) \setminus \{\emptyset\}$ is a function that assigns to each $s \in S$ a non-empty set of actions available to each agent $a \in Agt$. For $s \in S_{com}$, $d(s, a) = CA$. For $q \in S_{act}$, $d(q, a) \subseteq Act'$.
  For every $s, s' \in S, a \in Agt$, $d(s, a) = d(s', a)$ if $s_a = s'_a$.
- $c : Act \to \mathbb{Z}^{Res}$ is function which models consumption and production of resources by actions (a positive integer means consumption, a negative one production). We stipulate that $c(idle) = (0)_{i \in Res}$ and $c(com(k(s_a), A)) = (|k(s_a)|, 0, \ldots, 0)$.
- $\delta : S \times Act^n \to S$ is a partial function which, for every $s \in S$ and joint action $\sigma \in d(s, a_1) \times \cdots \times d(s, a_n)$, returns the state resulting from executing σ in s. For $q \in S_{act}$, $\delta(q, \sigma) \in S_{com}$, and for $s \in S_{com}$, $\delta(s, \sigma) \in S_{act}$. Additional constraints on δ are given in Definition 2.2 below.

Before giving a definition of the effect of a communication action, we first explain the intuitions underlying the two different kinds of states and two different kinds of actions in communication models. Communication models are intended to make explicit the assumption that agents in a coalition may share their knowledge through communication before selecting a joint action. Intuitively, S_{com} is the set of states which are initial states or states resulting from the execution of an ontic (non-communication) action. In S_{com} states, agents execute communication actions in CA. S_{act} is the set of states resulting from communication actions. In S_{act} states, agents execute ontic actions in Act'. Computations are therefore sequences of alternating states from S_{com} and S_{act}. For technical reasons, we require that for each $a \in Agt$ and $q \in S_{act}$, a distinguished propositional variable $p_{act} \in q_a$. This distinguishes agents' states in S_{act} from agents' states in S_{com} and allows them to select different actions in states which would otherwise be epistemically indistinguishable. Executing any action in a S_{act} state removes p_{act} from the agent's state.

The effect of a communication action $com(k(s_a), A)$ is to add the formulas in $k(s_a)$ to the local state of each agent $a' \in A$ that executes a corresponding communication action $com(k(s_{a'}), A)$, i.e., only agents that communicate with exactly the same set of agents exchange information. This is an important restriction on communication: if agents A execute communication actions of the form $com(k(s_a), A)$, and an agent b in $Agt \setminus A$ sends its formulas to some agent a in A, these formulas will not be added to a's state. We denote the joint communication action by a coalition A in which all agents $a \in A$ execute $com(k(s_a), A)$ by $com_A(s) = (com(k(s_a), A))_{a \in A}$.

As explained above, the state of a may contain formulas of the form $K_b \psi$ in addition to formulas of the form $K_a \psi$. Formulas of the form $K_b \psi$ represent results of previous communication actions by other agents; they are not communicated by a to other agents, and are forgotten by a at the next communication step. We made this modelling choice to avoid explicitly updating the state of a to remove the formulas about b's knowledge that are inconsistent with a new communication from b.

Definition 2.2. For every $s = (s_{a_1}, \ldots, s_{a_n}, s_e) \in S_{com}$ and joint communication action $\sigma = ((com(k(s_{a_1}), A_1), \ldots, com\,(k(s_{a_n}), A_n))$, $\delta(s, \sigma) = (q_{a_1}, \ldots, q_{a_n}, s_e)$, where $q_a = k(s_a) \cup \{p_{act}\} \cup \bigcup_{a' \in A} k(s_{a'})$ if $a \in A$ and each $a' \in A$ executes $com(k(s_{a'}), A)$, otherwise $q_a = k(s_a) \cup \{p_{act}\}$. For every $s \in S_{com}$, and every $a \in Agt$, $k(s_a) = s_a$ and $p_{act} \notin s_a$.

We introduce some standard terminology and notational conventions. We denote by $D(s)$ the set of joint actions by all agents in s: $D(s) = d(s, a_1) \times \cdots \times d(s, a_n)$. A *computation* is an infinite sequence $s_0, q_0, s_1, q_1, \ldots$ of alternating states from S_{com} and S_{act}, such that for every s_i, q_i, s_{i+1}, $q_i = \delta(s_i, \sigma)$ for some $\sigma \in D(s_i)$, and $s_{i+1} = \delta(q_i, \sigma)$ for some $\sigma \in D(q_i)$. We denote by $D_A(s)$ the set of all joint actions by agents in A at s. For $\sigma \in D(s)$, we denote by σ_A the restriction of σ to A, and we set $c(\sigma_A) = (c(\sigma_a))_{a \in A}$. The outcomes of $\sigma \in D_A(s)$ is the set of states reached when A executes σ: $out(s, \sigma) = \{s' \in S \mid \exists \sigma' \in D(s) : \sigma = \sigma'_A \wedge s' = \delta(s, \sigma')\}$.

A *(perfect recall) strategy for a coalition* $A \subseteq Agt$ is a mapping $F_A : S^+ \to Act^{|A|}$ (from finite non-empty sequences of states to joint actions by A) such that, for every $\lambda s \in S^+$, (a finite sequence consisting of a sequence λ followed by s), $F_A(\lambda s) \in D_A(s)$. A computation $\lambda \in S^\omega$ is consistent with a strategy F_A iff, for all $i \geq 0$, the $i+1$st state of λ, $\lambda[i+1]$, is in $out(\lambda[i], F_A(\lambda[0,i]))$ (where $\lambda[i,j]$ is the subsequence of λ between indices i and j). Overloading notation, we denote the set of all computations λ consistent with F_A that start from s by $out(s, F_A)$. Given a bound $b \in B$, a computation $\lambda \in out(s, F_A)$ is b-consistent with F_A iff, for every $i \geq 0$, for every $a \in A$,

$$b_a - \sum_{j=0}^{j=i-1} c(F_a(\lambda[0,j])) \geq c(F_a(\lambda[0,i]))$$

where b_a is a's allocation of resources from b and $F_a(\lambda[0,j])$ is the action by a in the joint action returned by F_A for the sequence of states $\lambda[0,j]$. This condition requires that the resources accumulated by a on the path so far, plus the original bound, is greater than or equal to the cost of executing the next action by a in the strategy. It is equivalent to the condition that the resource values are never negative.

F_A is a b-strategy if all $\lambda \in out(s, F_A)$ are b-consistent. We need this notion for evaluating $\langle\langle A^b \rangle\rangle \Box \phi$ modalities: for formulas of this form to be true, there should be a strategy of A where on all computations A never runs out of resources. For formulas of the form $\langle\langle A^b \rangle\rangle \bigcirc \phi$ and $\langle\langle A^b \rangle\rangle \phi \,\mathcal{U}\, \psi$ we only need to ensure that there exists a strategy where every computation satisfies ϕ in the next state (respectively, satisfies ψ after finitely many steps) before possibly running out of resources. A computation λ is b-maximal for a strategy F_A if it cannot be extended further without losing b-consistency (the next action prescribed by F_A would violate b-consistency). The set of all b-maximal computations starting from state s that are b-consistent with F_A is denoted by $out(s, F_A, b)$. Note that $out(s, F_A, b)$ is always non-empty since it includes a computation consisting of just s itself. The resulting semantics is essentially the same as the semantics in terms of maximal resource-extended paths defined in [7].

In the presence of imperfect information, it makes sense to consider only *uniform* strategies rather than arbitrary ones. A strategy is uniform if, after epistemically indistinguishable histories, agents select the same actions. Two states s and t are epistemically indistinguishable by agent a, denoted by $s \sim_a t$, if a has the same local state (knows the same formulas) in s and t: $s \sim_a t$ iff $s_a = t_a$. Two histories $s_0, \ldots, s_k$ and $t_0, \ldots, t_k$ are indistinguishable by a (also denoted by $\sim_a$) if, and only if, for all $j \in [0,k]$, $s_j \sim_a t_j$. An agent's strategy is uniform iff $F_a(\lambda) = F_a(\lambda')$ for all $\lambda \sim_a \lambda'$.

We can now give the truth definition for RB$\pm$ATSEL$_c$.

- $M, s \models p$ iff $p \in s_e$
- boolean connectives have standard truth definitions
- $M, s \in S_{com} \models \langle\langle A^b \rangle\rangle \bigcirc \phi$ iff $\exists$ a uniform strategy F_A such that for all $\lambda \in out(s, F_A, b)$: $M, \lambda[2] \models \phi$
- $M, s \in S_{act} \models \langle\langle A^b \rangle\rangle \bigcirc \phi$ iff $\exists$ a uniform strategy F_A such that for all $\lambda \in out(s, F_A, b)$: $M, \lambda[1] \models \phi$
- $M, s \models \langle\langle A^b \rangle\rangle \phi \,\mathcal{U}\, \psi$ iff $\exists$ a uniform strategy F_A such that for all $\lambda \in out(s, F_A, b)$, $\exists i \geq 0$: $M, \lambda[i] \models \psi$ and $M, \lambda[j] \models \phi$ for all $j \in \{0, \ldots, i-1\}$
- $M, s \models \langle\langle A^b \rangle\rangle \Box \phi$ iff $\exists$ a uniform b-strategy F_A such that for all $\lambda \in out(s, F_A)$ and $i \geq 0$: $M, \lambda[i] \models \phi$
- $M, s \models K_a\phi$ iff $K_a\phi \in k(s_a)$ or $\phi \in s_a$ for $K_a\phi \notin k(s_a)$

3 EXAMPLE

In this section, we illustrate RB$\pm$ATSEL$_c$ using a simple example based on that given in [3]. In the scenario, two robotic agents 1 and 2 monitor a space for something bad happening. In [3], the space was an art gallery, and 'something bad' could be a fire. If something bad happens, both robots should clear the space of visitors. Whether something bad happens is determined by the environment agent e. Monitoring the space requires energy, and the robots must periodically return to a charging station to recharge. We assume that is not possible to observe the space from the charging station, and that a single (charged) robot is sufficient to detect something bad happening.

At each timestep, the environment can perform an *idle* action or cause something bad to happen, *bad*. Agents 1 and 2 can perform an *idle* action, an observation action, *obs* (which, for simplicity, implicitly moves the agent from the charging station to the space before performing the observation), clear the space of visitors *clr*, recharge their battery *gen*, or communicate *com*. The agents require a single resource, *energy*. The *gen* action produces five units of energy; *obs* and *clr* consume one unit of energy; $com(k(s_a), A)$ consumes $|k(s_a)|$ units of energy; and *idle* consumes no energy.

We use proposition b to denote that something bad has happened, c_i ($i \in \{1,2\}$) to denote that agent i has just charged their battery, and r to denote that the space has been cleared of visitors. The global system state is represented by $s = (s_1, s_2, s_e)$, where s_i ($i = \{1,2\}$) is the local state of i, and s_e is the state of the environment. The set of formulas Φ which constitute possible contents of agents' states includes information on whether the agents have (just) charged, whether something bad has happened, and whether the space has been cleared.

The scenario can be modelled by a communication model where $\Phi = \{K_1 b, K_2 b, K_1 c_1, K_2 c_2, K_1 r, K_2 r\}$, $Agt = \{1, 2, e\}$, $Res = \{energy\}$, $\Pi = \{b, c_1, c_2, r\}$, $S_{com} = 2^\Phi \times 2^\Phi \times 2^\Pi$, $S_{act} = 2^{\Phi \cup \{p_{act}\}} \times 2^{\Phi \cup \{p_{act}\}} \times 2^\Pi$, $Act = \{idle, bad, obs, clr, gen\} \cup CA$.

d contains communication actions and *idle* for $s \in S_{com}$ and is defined for $q \in S_{act}$ as follows:

(1) $idle \in d(q, i)$ for all $i \in \{1, 2, e\}$
(2) $bad \in d(q, e)$ iff $b \notin q_e$
(3) $obs \in d(q, i)$ for all $i \in \{1, 2\}$
(4) $gen \in d(q, i)$ for all $i \in \{1, 2\}$
(5) $clr \in d(q, i)$ for all $i \in \{1, 2\}$ iff $b \in q_i$ (the space is only cleared if something bad has happened)

$c(idle) = 0$, $c(gen) = -5$, $c(obs) = c(clr) = 1$, $c(com(k(s_a), A)) = |k(s_a)|$.

δ is defined based on the following post conditions of actions:

(1) *idle* performed by agent $i \in \{1,2\}$ removes $K_i c_i$ from s_i and c_i from s_e; *idle* performed by e removes b from s_e;
(2) *bad* performed by e adds b to s_e;
(3) *obs* performed by agent $i \in \{1,2\}$ removes $K_i c_i$ from s_i and c_i from s_e, and, if performed in a state where b is true (false), adds (removes) $K_i b$ to (from) s_i;
(4) *clr*, if performed by both agents 1 and 2, removes $K_i c_i$ from s_i and c_i from s_e, and adds $K_i r$ to s_i and r to s_e; otherwise (if only performed by one agent i) it removes $K_i c_i$ from s_i and c_i from s_e.

(5) *gen* performed by agent $i \in \{1,2\}$ adds $K_i c_i$ to s_i and c_i to s_e;
(6) $com(k(s_a), A)$ adds the formulas in $k(s_a)$ to the state of each agent in A if each $a' \in A$ executes a corresponding communication action $com(k(s_{a'}), A)$.

Intuitively, if agents 1 and 2 have the goal of clearing the space if something bad happens, they should take turns charging and observing. If one of the agents is always observing, then that agent will observe that the bad thing happened (will come to know b), and if the agents communicate their states to each other before selecting the next action, the other agent will acquire $K_i b$ where i is the observing agent; then the agents can synchronise and both execute *clr* action and achieve their goal of clearing the space.

The following property states that if something bad happens, then this will be known by the observing agent in the next state, and in the next state after that, the space will be cleared (which requires both agents to cooperate)

$$b \to \langle\langle \{1,2\}^{(2,0)} \rangle\rangle \bigcirc ((K_1 b \vee K_2 b) \wedge \langle\langle \{1,2\}^{(1,1)} \rangle\rangle \bigcirc r)$$

4 MODEL CHECKING RB±ATSEL$_c$

In this section prove the following theorem:

THEOREM 4.1. *The model checking problem for RB± ATSEL$_c$ is decidable.*

To prove decidability, we give an algorithm that, given a communication model M and formula ϕ_0, returns the set of states $[\phi_0]_M$ satisfying ϕ_0.[1]

4.1 Coalition Uniform Strategies

The algorithm (and proof) relies on the notion of a *coalition uniform* (with respect to distributed knowledge of the coalition) strategy introduced in [3]. We first recall the definition of coalition uniformity from [3] and explain its relationship to uniform strategies.

For a coalition A, the indistinguishability of two states s and t, $s \sim_A t$, means that A as a whole has the same knowledge in s and t; for distributed knowledge, it means $\bigcup_{a\in A} s_a = \bigcup_{a\in A} t_a$. $\sim_A$ can be lifted to histories in the same way as $\sim_a$: $s_0, \ldots, s_k \sim_A t_0, \ldots, t_k$ iff for all $j \in [0,k]$, $s_j \sim_A t_j$.

Definition 4.2. A strategy F_A for A is coalition-uniform with respect to $\sim_A$ if for all $\lambda, \lambda' \in S^+$, if $\lambda \sim_A \lambda'$, then $F_A(\lambda) = F_A(\lambda')$.

It is known that the model-checking problem for ATL with imperfect information and perfect recall is decidable for coalition uniformity with respect to distributed knowledge of the coalition [9, 11]. In [3], it was shown that for RB±ATSEL this also holds, for any decidable notion of coalition uniformity (not just distributed knowledge).

Coalition uniform strategies do not have to be uniform, and vice versa. As an example of a coalition uniform strategy that is not uniform, let A be a coalition of two agents, 1 and 2. The strategy requires that in a state where agent 1 knows p, the coalition should perform a joint action (α, α), while if neither agent knows p, they should perform (β, β). The distributed knowledge of the coalition is p in the first case and empty in the second, so the situations are coalitionally distinguishable and the strategy is coalition uniform for A. For agent 2 however the two states are not distinguishable; so this strategy requires 2 to perform different actions in indistinguishable states and is not a uniform strategy for 2.

Uniform strategies also do not have to be coalition uniform. Consider the following strategy that is uniform for 1 and for 2. Agent $i \in \{1,2\}$ performs α if i knows p, and β otherwise. Then in a state where 1 knows p and 2 does not, the agents will execute (α, β) and in a state where 1 does not know p and 2 does, they will execute (β, α). However this strategy is not coalition uniform, since the knowledge of the coalition is the same in both cases, i.e., $\{p\}$, while the joint action chosen by the coalition is different.

In RB ± ATSEL$_c$, we add a communication step to make each agent's knowledge the same as the distributed knowledge of the coalition. To ensure that coalition uniform and uniform strategies coincide in RB±ATSEL$_c$, formulas in the agents's states are 'labelled' with the name of the agent. That is, instead of saying that agent 1's state contains p, we say that agent 1's state contains $K_1 p$. When the agent communicates the contents of its state, it communicates $K_1 p$. When the distributed knowledge of the coalition consists of such labelled formulas, the example above of a uniform strategy that is not coalition uniform cannot be constructed. The distributed knowledge of the coalition is $\{K_1 p\}$ when only agent 1 knows p, and $\{K_2 p\}$ when only agent 2 knows p. A strategy that selects (α, β) in the first case and (β, α) in the second is coalition uniform.

THEOREM 4.3. *In communication models, if at each communication step all agents in A execute $com(k(s_a), A)$, then a strategy for coalition A is coalition uniform if and only if it is uniform for all $a \in A$.*

PROOF. Clearly action selection at each communication step (in a state $s \in S_{com}$) is both coalition uniform and uniform. In fact, action selection in S_{com} only depends on the last state in the history (is memoryless), since each agent in A sends the contents of its state to A.

Next we show that action selection in histories that end in a state $s \in S_{act}$ is coalition uniform if and only if it is uniform. In order to do this, we show that, under the assumption that at each communication step all agents in A execute $com(k(s_a), A)$, two states in $s, t \in S_{act}$ are indistinguishable for A if and only if they are indistinguishable for each $a \in A$.

Let $\forall a \in A$, $s_a = t_a$. Then clearly $\bigcup_{a\in A} s_a = \bigcup_{a\in A} t_a$. Let $\bigcup_{a\in A} s_a = \bigcup_{a\in A} t_a$. By Definition 2.2, $\bigcup_{a\in A} s_a = \bigcup_{a\in A} k(s_a) \cup \{p_{act}\}$ and $\bigcup_{a\in A} t_a = \bigcup_{a\in A} k(t_a) \cup \{p_{act}\}$. Observe also that for all $a \in A$ and all ϕ, $K_a\phi \in \bigcup_{a\in A} s_a$ iff $K_a\phi \in k(s_a)$ (the only way a formula prefixed with K_a can appear in any agent's state is if it was in the state of a at the previous step, and hence is still in $k(s_a)$). The same applies to t.

Assume by contradiction that for some $a \in A$, $s_a \neq t_a$. Then either for some ϕ, $K_a\phi \in s_a$ and $K_a\phi \notin t_a$, or for some ψ, and $b \in A$, $b \neq a$, $K_b\psi \in s_a$ and $K_b\psi \notin t_a$. The second case implies that also $K_b\psi$ is in $k(s_b)$ and not in $k(t_b)$ (since b communicates all its formulas to a), so it suffices to consider the first case. If $K_a\phi \in k(s_a)$, then $K_a\phi \in \bigcup_{a\in A} s_a = \bigcup_{a\in A} k(s_a) \cup \{p_{act}\}$. If $K_a\phi \notin t_a$, then $K_a\phi \notin \bigcup_{a\in A} k(t_a) \cup \{p_{act}\}$. But then contrary to assumption, $\bigcup_{a\in A} s_a \neq \bigcup_{a\in A} t_a$.

[1] The model checking algorithm for RB±ATSEL given in [3] does not consider communication models, or the cost of communication actions.

This implies that two sequences of states are indistinguishable for A if and only if they are indistinguishable for each $a \in A$. Hence a strategy is coalition uniform if and only if it is uniform. □

4.2 Model Checking Algorithms

We now give the model checking algorithm for RB±ATSEL$_c$, which is shown in Algorithm 1. Given ϕ_0, we produce a set of subformulas $Sub(\phi_0)$ of ϕ_0 in the usual way (but excluding subformulas in the scope of a knowledge modality), ordered in increasing order of complexity. We then proceed by cases. For all formulas in $Sub(\phi)$ apart from $K_a\phi$, $\langle\langle A^b\rangle\rangle\bigcirc\phi$, $\langle\langle A^b\rangle\rangle\phi\,\mathcal{U}\,\psi$ and $\langle\langle A^b\rangle\rangle\Box\phi$ we essentially run the standard ATL model-checking algorithm [5]. For formulas of the form $K_a\phi$, we simply check whether $K_a\phi$ is in $k(s_a)$ or ϕ is in $s_a \setminus k(s_a)$.

Algorithm 1 Labelling ϕ_0

function RB±ATSEL$_c$-LABEL(M, ϕ_0)
  for $\phi' \in Sub(\phi_0)$ **do**
    case $\phi' = p,\ \neg\phi,\ \phi \vee \psi$ standard, see [5]
    case $\phi' = K_a\phi$
      $[\phi']_M \leftarrow \{\, s \in S \mid K_a\phi \in k(s_a) \vee \phi \in s_a \setminus k(s_a)\}$
    case $\phi' = \langle\langle A^b\rangle\rangle\bigcirc\phi$
      $[\phi']_M \leftarrow \{\, s \in S \mid$ NEXT($[node_0(s, b)]$, $\{\ \}$, $\langle\langle A^b\rangle\rangle\bigcirc\phi$)$\}$
    case $\phi' = \langle\langle A^b\rangle\rangle\phi\,\mathcal{U}\,\psi$
      $[\phi']_M \leftarrow \{\, s \in S \mid$ UNTIL($[node_0(s, b)]$, $\{\ \}$, $\langle\langle A^b\rangle\rangle\phi\,\mathcal{U}\,\psi$)$\}$
    case $\phi' = \langle\langle A^b\rangle\rangle\Box\phi$
      $[\phi']_M \leftarrow \{\, s \in S \mid$ BOX($[node_0(s, b)]$, $\{\ \}$, $\langle\langle A^b\rangle\rangle\Box\phi$)$\}$
  return $[\phi_0]_M$

Labelling states with formulas of the form $\langle\langle A^b\rangle\rangle\bigcirc\psi$, $\langle\langle A^b\rangle\rangle\phi\,\mathcal{U}\,\psi$ and $\langle\langle A^b\rangle\rangle\Box\phi$ is done by the functions NEXT, UNTIL and BOX respectively. Each algorithm proceeds by depth-first and-or search of M, and takes a stack (list) of 'open' nodes B, a set of 'closed' nodes C, and a formula of the appropriate form in $Sub(\phi_0)$ as input. The set of 'closed' nodes C is used to check for uniformity of strategies with respect to the coalition indistinguishability relation $\sim_A$. As shown by Theorem 4.3, this is necessary and sufficient to ensure uniformity. Information about the state of the search is recorded in a tree of nodes. A *node* consists of a state of M, the resources available to the agents A in that state, and the finite path of nodes from the root node to this node. Edges in the tree correspond to joint actions by agents in A and are labelled with the action taken. For each node n, the function $s(n)$ returns the state represented by n, $p(n)$ returns the nodes on the path to n, and $a(n)$ returns the joint action taken by A to reach $s(n)$. We use $p(n)[i]$ to denote the i-th node in the path $p(n)$, and $p(n)[0, j-1]$ to denote the prefix of $p(n)$ up to the j-th node. The function $e(n)$ returns the resource availability of agents, and $e_{i,k}$ is the i-th resource for agent $k \in A$ in $s(n)$. The function $node_0(s, b)$ returns the root node, i.e., a node n_0 such that $s(n_0) = s$, $p(n_0) = [\ ]$, $a(n_0) = \textit{no-op}$, and $e_{i,k}(n_0) = b_{i,k}$ for all resources i and agents $k \in A$. The function $node(n, \sigma, s')$ returns a node n' where $s(n') = s'$, $p(n') = [p(n) \cdot n]$, $a(n') = \sigma$, and for all resources i and agents $k \in A$, $e_{i,k}(n') = e_{i,k}(n) - c(\sigma_k)$.

Functions $hd(u)$, $tl(u)$ return the head and tail of a list u, and $u \circ v$ concatenates the lists u and v.

Algorithm 2 Labelling $\langle\langle A^b\rangle\rangle\bigcirc\psi$

1: **function** NEXT(B, C, $\langle\langle A^b\rangle\rangle\bigcirc\psi$)
2:   **if** $B = [\ ]$ **then**
3:     **return** *true*
4:   $n \leftarrow hd(B)$
5:   **if** $s(n) \in S_{com}$ **then**
6:     **if** $|p(n)| < 1$ **then**
7:       $\sigma \leftarrow com_A(s(n))$
8:       **if** $c(\sigma) \leq e(n)$ **then**
9:         $P \leftarrow \{node(n, \sigma, s') \mid s' \in out(s(n), \sigma)\}$
10:         **return** NEXT($P \circ tl(B)$, C, $\langle\langle A^b\rangle\rangle\bigcirc\psi$)
11:     **else**
12:       **if** $s(n) \in [\psi]_M$ **then**
13:         **return** NEXT($tl(B)$, $C \cup \{n\}$, $\langle\langle A^b\rangle\rangle\bigcirc\psi$)
14:   **else**
15:     **if** $\exists n' \in C$ **then**
16:       $\sigma \leftarrow a(n')$
17:       **if** $\sigma \in D_A(s(n))$ **then**
18:         $P \leftarrow \{node(n, \sigma, s') \mid s' \in out(s(n), \sigma)\}$
19:         **return** NEXT($P \circ tl(B)$, C, $\langle\langle A^b\rangle\rangle\bigcirc\psi$)
20:     **else**
21:       $ActA \leftarrow \{\sigma \in D_A(s(n)) \mid c(\sigma) \leq e(n)\}$
22:       **for** $\sigma \in ActA$ **do**
23:         $P \leftarrow \{node(n, \sigma, s') \mid s' \in out(s(n), \sigma)\}$
24:         **if** NEXT($P \circ tl(B)$, C, $\langle\langle A^b\rangle\rangle\bigcirc\psi$) **then**
25:           **return** *true*
26:   **return** *false*

The function NEXT for $\langle\langle A^b\rangle\rangle\bigcirc\psi$ formulas is shown in Algorithm 2. If $B = [\ ]$ there are no more open nodes to consider, and NEXT returns true, indicating that a strategy exists to enforce $\langle\langle A^b\rangle\rangle\bigcirc\psi$ (lines 2–3). Otherwise there are two cases. The first case, $s(n) \in S_{com}$ (line 5), breaks down into two sub-cases. In the first sub-case, $s(n)$ is the initial state ($\lambda[0]$), so we choose the communication action $com_A(\lambda[0])$. If the cost of the communication action is less than the resource availability in $s(n)$, we call NEXT recursively to continue the search, pushing the nodes corresponding to the successor states onto the stack of open nodes. The second sub-case is where we are in a state where we have performed a communication and a non-communication action (i.e., we are in $\lambda[2]$). If the state satisfies ψ, we terminate the current branch of the search by adding the current node n to the set of closed nodes, and return the result of calling NEXT recursively on the remaining open branches. Note that this means that the closed list contains only two-step strategies. In the second case $s(n) \in S_{act}$ (line 14), and we must choose a non-communication action. Again there are two sub-cases. If there is a node n' in the closed set C, we must choose the action $a(n')$. For all $n' \in C$, the state $s(p(n')[1]) \sim_A s(n)$ is in S_{act} (since in each $s(p(n')[1])$, the knowledge of each $a \in Agt$ has changed in the same way as a result of communication), so, for a uniform strategy, we must choose the same action. We generate a new node for each possible outcome state of the action, and call NEXT recursively to continue the search, pushing the nodes corresponding to the successor states onto the stack of open nodes. In the second sub-case (line 20) no action is required at the current state for uniformity. For each action that is possible in the current state given the current resource availability, we attempt to find a strategy for each of the outcome states of that action, again calling NEXT recursively to continue the search. If the current open branch is 1 or 2 steps long and $s(n)$ does not satisfy ψ, or the communication action or a non-communication action necessary for uniformity requires more resources than are available in the current state, or there is no joint action $\sigma \in ActA$ where ψ can be enforced in all outcome states, we 'fall through' and return false (line 26).

Algorithm 3 Labelling $\langle\langle A^b \rangle\rangle \phi\, \mathcal{U}\, \psi$

1: **function** UNTIL(B, C, $\langle\langle A^b \rangle\rangle \phi\, \mathcal{U}\, \psi$)
2: **if** $B = [\,]$ **then**
3: **return** *true*
4: $n \leftarrow hd(B)$
5: **if** $\exists n' \in p(n) : s(n') = s(n) \wedge (\forall i, k : e_{i,k}(n') \geq e_{i,k}(n))$ **then**
6: **return** *false*
7: **for** $(i, k) \in \{(i, k) \mid i \in Res, k \in A, \exists n' \in p(n) : s(n') = s(n) \wedge (\forall j, m : e_{j,m}(n') \leq e_{j,m}(n)) \wedge e_{i,k}(n') < e_{i,k}(n)\}$ **do**
8: $e_{i,k}(n) \leftarrow \infty$
9: **if** $s(n) \in [\psi]_M$ **then**
10: **return** UNTIL($tl(B)$, $C \cup \{n\}$, $\langle\langle A^b \rangle\rangle \phi\, \mathcal{U}\, \psi$)
11: **if** $s(n) \notin [\phi]_M$ **then**
12: **return** *false*
13: **if** $s(n) \in S_{com}$ **then**
14: $\sigma \leftarrow com_A(s(n))$
15: **if** $c(\sigma) \leq e(n)$ **then**
16: $P \leftarrow \{node(n, \sigma, s') \mid s' \in out(s(n), \sigma)\}$
17: **return** UNTIL($P \circ tl(B)$, C, $\langle\langle A^b \rangle\rangle \phi\, \mathcal{U}\, \psi$)
18: **else**
19: **if** $\exists n' \in C : p(n) \cdot n \sim_A p(n')[0, |p(n)|] \wedge |p(n')| > |p(n)|$ **then**
20: $\sigma \leftarrow a(p(n')[|p(n)| + 1])$
21: $P \leftarrow \{node(n, \sigma, s') \mid s' \in out(s(n), \sigma)\}$
22: **return** UNTIL($P \circ tl(B)$, C, $\langle\langle A^b \rangle\rangle \phi\, \mathcal{U}\, \psi$)
23: **else**
24: $ActA \leftarrow \{\sigma \in D_A(s(n)) \mid c(\sigma) \leq e(n)\}$
25: **for** $\sigma \in ActA$ **do**
26: $P \leftarrow \{node(n, \sigma, s') \mid s' \in out(s(n), \sigma)\}$
27: **if** UNTIL($P \circ tl(B)$, C, $\langle\langle A^b \rangle\rangle \phi\, \mathcal{U}\, \psi$) **then**
28: **return** *true*
29: **return** *false*

The function UNTIL for $\phi' = \langle\langle A^b \rangle\rangle \phi\, \mathcal{U}\, \psi$ formulas is shown in Algorithm 3. As in NEXT, if there are no more open nodes to consider, UNTIL returns true. Otherwise we check whether the state $s(n)$ has been encountered before on $p(n)$, i.e., $p(n)$ ends in a loop. If the loop is unproductive (i.e., resource availability has not increased since the previous occurrence of $s(n)$ on the path $p(n)$), then the loop is not necessary for a successful strategy, and search on this branch is terminated (lines 5–6). However, if the loop strictly increases the availability of at least one resource i for some agent k and does not decrease the availability of other resources, then $e_{i,k}(n)$ is replaced with ∞ as a shorthand denoting that any finite amount of i can be produced by repeating the loop sufficiently many times (lines 7–8). We then check if the second argument ψ of ϕ' is true in $s(n)$. If so, search terminates on the current branch, the current node n is added to the set of closed nodes and search continues on a different branch by expanding the next open node in B (lines 9–10). If the current branch is not closed (i.e., ψ is not true in $s(n)$, but ϕ is true in $s(n)$, (lines 11–12)), search continues on this branch. As for NEXT, there are two cases. The first case is when $s(n) \in S_{com}$, and the agents must execute a communication action. If the cost of the communication action is less than the resource availability in $s(n)$, we generate a new node for each possible outcome state of communication, and call UNTIL recursively to continue the search, pushing the nodes corresponding to the successor states onto the stack of open nodes (lines 13–17). In the second case $s(n) \in S_{act}$ (line 18), and we must choose a non-communication action. As in NEXT, there are two sub-cases. In the first sub-case, the current path (including the current node) is epistemically indistinguishable from a (prefix of) a path to a closed node n' (line 19), and the same action, σ, must be selected in the current state as in the corresponding state in n' (line 20). We generate a new node for each possible outcome state of the action, and call UNTIL recursively to continue the search, pushing the nodes corresponding to the successor states onto the stack of open nodes (lines 21–22). In the second sub-case, no action is required at the current state for uniformity, and, for each action that is possible in the current state given the current resource availability, we attempt to find a strategy for each of the outcome states of that action (lines 24–28). If a strategy cannot be found for any action possible in $s(n)$, UNTIL returns false (line 29).

The function BOX for $\phi' = \langle\langle A^b \rangle\rangle \Box \phi$ formulas is shown in Algorithm 4. As for NEXT and UNTIL, if there are no more open nodes to consider, BOX returns true (lines 2–3). If ϕ is false in the state represented by node n, $s(n)$, it returns false, terminating search of the current branch of the search tree (lines 5–6). Otherwise we check whether $p(n)$ ends in a loop. If the loop decreases the amount of at least one resource for one agent without increasing the availability of any other resource, it cannot form part of a successful strategy, and the search terminates returning false (lines 7–8). If a non-decreasing loop is found, then it is possible to maintain the invariant formula ϕ forever without expending any resources, and the search terminates on the current branch, the current node n is added to the set of closed nodes, and search continues on a different branch by expanding the next open node in B (lines 9-10). The remaining cases are similar to UNTIL. If the current branch is not closed, search continues on the branch. The first case is when a communication action must be performed in the current state (lines 12–15). If a non-communication action must be performed in $s(n)$, we check if an action is required for the strategy to be uniform (lines 17–20), and, if not, we consider each action that is possible in the current state given the current resource availability (lines 22–26).

Next, we prove Theorem 4.1.

Proof. The proof of termination and correctness of Algorithm 1 with respect to coalition uniform strategies is similar to the proof in [3]. In particular, it also works for the ATL case (infinite bounds). By Theorem 4.3, a coalition uniform strategy exists if and only if a uniform strategy exists, where all agents in a coalition communicate their knowledge to each other. No cheaper uniform strategy exists, since the cost of a communication action is always proportional

Algorithm 4 Labelling $\langle\langle A^b \rangle\rangle \Box \phi$

function BOX(B, C, $\langle\langle A^b \rangle\rangle \Box \phi$)
  if $B = [\,]$ **then**
    return *true*
  $n \leftarrow hd(B)$
  if $s(n) \notin [\phi]_M$ **then**
    return *false*
  if $\exists n' \in p(n) : s(n') = s(n) \wedge$
      $(\forall j, k : e_{j,k}(n') \geq e_{j,k}(n)) \wedge$
      $(\exists j, k : e_{j,k}(n') > e_{j,k}(n))$ **then**
    return *false*
  if $\exists n' \in p(n) : s(n') = s(n) \wedge$
      $(\forall j, k : e_{j,k}(n') \leq e_{j,k}(n))$ **then**
    return BOX($tl(B)$, $C \cup \{n\}$, $\langle\langle A^b \rangle\rangle \Box \phi$)
  if $s(n) \in S_{com}$ **then**
    $\sigma \leftarrow com_A(s(n))$
    if $c(\sigma) \leq e(n)$ **then**
      $P \leftarrow \{node(n, \sigma, s') \mid s' \in out(s(n), \sigma)\}$
      return BOX($P \circ tl(B)$, C, $\langle\langle A^b \rangle\rangle \phi \,\mathcal{U}\, \psi$)
  else
    if $\exists n' \in C : p(n) \cdot n \sim_A p(n')[0, |p(n)|] \wedge$
        $|p(n')| > |p(n)|$ **then**
      $\sigma \leftarrow a(p(n')[|p(n)| + 1])$
      $P \leftarrow \{node(n, \sigma, s') \mid s' \in out(s(n), \sigma)\}$
      return BOX($P \circ tl(B)$, C, $\langle\langle A^b \rangle\rangle \Box \phi$)
    else
      $ActA \leftarrow \{\sigma \in D_A(s(n)) \mid c(\sigma) \leq e(n)\}$
      for $\sigma \in ActA$ **do**
        $P \leftarrow \{node(n, \sigma, s') \mid s' \in out(s(n), \sigma)\}$
        if BOX($P \circ tl(B)$, C, $\langle\langle A^b \rangle\rangle \Box \phi$) **then**
          return *true*
  return *false*

to the size of the agent's state, and communicating with agents outside the current coalition is just as expensive and results in more indistinguishable states. □

4.3 Complexity

The lower bound on the complexity of the model checking problem is provided by the complexity of model checking RB±ATL [2]:

Theorem 4.4. *The model checking problem for RB± ATSEL$_c$ is 2EXPTIME hard.*

We conjecture that the upper bound is 3EXPTIME.

5 REDUCING COMMUNICATION COST

In this section, we briefly introduce a variant type of communication models called *flexible communication models*, where the whole content of the agent's state does not have to be communicated. We use flexible communication models to illustrate a simple form of agent reasoning (closure of the state). As a motivating example, consider a variant of the museum guards scenario where the guards are both watching the same security monitor; if something bad happens, not only do they both know this, but they also know that each other knows, without the need for communication. In other words, agent 1 can derive $K_2 b$ in addition to $K_1 b$. This means that agent 2 does not have to communicate $K_2 b$ to agent 1 (and vice versa). We assume that agents' inferences about each other's knowledge are correct: if $K_i \phi \in s_j$, then $K_i \phi \in s_i$.

Definition 5.1. A flexible communication model of RB±ATSEL$_c$ is a tuple $M = (\Phi, Agt, Res, \Pi, S, Act, d, c, \delta, cl)$ where:

- Φ, Agt, Res, Π, S are as in Definition 2.1;
- in S_{com}, the set of available actions is
$$CA = \{com(X_a, B_a) \mid X_a = \{(a_1, s_1), \ldots, (a_k, s_k)\}, \\ s_i \subseteq k(s_a), a_i \in Agt, B_a \subseteq Agt\}$$
that is, agent a can send different subsets of its state to different agents in $B_a \subseteq Agt$.
- cl (closure) is an operation on s_a for $a \in Agt$ that adds to s_a formulas of the form $K_b \phi$, $b \neq a$, so that for every $s \in S$, $K_b \phi \in cl(s_a) \Rightarrow K_b \phi \in s_b$.
- δ is defined as: for every $s \in S_{com}$ and joint communication action $\sigma = ((com(X_1, B_1), \ldots, com(X_n, B_n)) \in D(s)$, δ returns the resulting state in S_{act}: $\delta(s, \sigma) = (t_{a_1}, \ldots, t_{a_n}, s_e)$, where
$$t_{a_i} = cl(k(s_{a_i})) \cup \{p_{act}\} \bigcup_{(a_i, s'_a) \in X_a \text{ such that } a \in B_i} s'_a$$

For a coalition uniform strategy to be a uniform strategy, we need to ensure that each agent's knowledge is the same as the distributed knowledge of the coalition. Now it is not necessary for agents to communicate their entire state to achieve this, since some formulas can be added by reasoning (closure).

Our approach to reducing the number of communicated formulas is as follows. In a state $s \in S_{com}$, for a coalition A, we know what the state of each agent $a \in A$ should contain after communication to make selection of ontic actions uniform: it is the distributed knowledge of A, $\cup_{a \in A} k(s_a)$. Given that Φ is finite, and s_a for each $a \in A$ is finite, there is a finite number of communication actions a can perform to try to make each agent's state to be $\cup_{a \in A} k(s_a)$: it can do some $com(X_a, B_a)$ where for each $(a_i, s_i) \in X_a$, $s_i \subseteq k(s_a)$ and $B_a \subseteq A$. The only thing we need to ensure is that this communication action selection is uniform: every time the agent is in an indistinguishable history, it performs the same communication action. We cannot do this for perfect recall communication strategies but we can do this for bounded or memoryless strategies since the model checking problem for those strategies is decidable [19]. That is, using a closed list of bounded histories including communication actions, we check all possible communication actions the agents in A can perform that result in $cl(q_b) = \cup_{a \in A} k(s_a)$ for every $b \in A$, while remaining uniform.

In the interests of brevity, we give only the modified algorithm for formulas of the form $\langle\langle A^b \rangle\rangle \phi \,\mathcal{U}\, \psi$; the algorithms for $\langle\langle A^b \rangle\rangle \bigcirc \psi$ and $\langle\langle A^b \rangle\rangle \Box \phi$ are similar.

The modified algorithm is shown in Algorithm 5. The only change required is to the selection of communication actions (lines 14–18); the remainder of the algorithm is identical to Algorithm 3. Rather than a single $com_A(s(n))$ communication action, we now generate all possible communication actions $ComA$ that (a) result in the knowledge of each agent being the same as the distributed knowledge of the coalition, (b) are uniform under bounded recall for histories of length k, and (c) cost less than the current resources of the coalition (line 14). The function $p_k(n)$ returns the last k nodes

Algorithm 5 Labelling $\langle\langle A^b \rangle\rangle \phi\, \mathcal{U}\, \psi$ for flexible communication models

1: **function** UNTIL(B, C, $\langle\langle A^b \rangle\rangle \phi\, \mathcal{U}\, \psi$)
2: **if** $B = [\,]$ **then**
3: **return** *true*
4: $n \leftarrow hd(B)$
5: **if** $\exists n' \in p(n) : s(n') = s(n) \wedge (\forall i, k : e_{i,k}(n') \geq e_{i,k}(n))$ **then**
6: **return** *false*
7: **for** $(i, k) \in \{(i, k) \mid i \in Res, k \in A, \exists n' \in p(n) : s(n') = s(n) \wedge (\forall j, m : e_{j,m}(n') \leq e_{j,m}(n)) \wedge e_{i,k}(n') < e_{i,k}(n)\}$ **do**
8: $e_{i,k}(n) \leftarrow \infty$
9: **if** $s(n) \in [\psi]_M$ **then**
10: **return** UNTIL($tl(B)$, $C \cup \{n\}$, $\langle\langle A^b \rangle\rangle \phi\, \mathcal{U}\, \psi$)
11: **if** $s(n) \notin [\phi]_M$ **then**
12: **return** *false*
13: **if** $s(n) \in S_{com}$ **then**
14: $ComA \leftarrow \{\sigma \in D_A(s(n)) \mid \forall s' \in out(s(n), \sigma)\ cl(s'_{a \in A}) = \bigcup_{a' \in A} k(s'_{a'}) \wedge \exists n' \in C : p_{k-1}(n) \cdot n \sim_{a \in A} p_k(n') \rightarrow \sigma_a = a_a(n') \wedge c(\sigma) \leq e(n)\}$
15: **for** $\sigma \in ComA$ **do**
16: $P \leftarrow \{node(n, \sigma, s') \mid s' \in out(s(n), \sigma)\}$
17: **if** UNTIL($P \circ tl(B)$, C, $\langle\langle A^b \rangle\rangle \phi\, \mathcal{U}\, \psi$) **then**
18: **return** *true*
19: **else**
20: **if** $\exists n' \in C : p(n) \cdot n \sim_A p(n')[0, |p(n)|] \wedge |p(n')| > |p(n)|$ **then**
21: $\sigma \leftarrow a(p(n')[|p(n)| + 1])$
22: $P \leftarrow \{node(n, \sigma, s') \mid s' \in out(s(n), \sigma)\}$
23: **return** UNTIL($P \circ tl(B)$, C, $\langle\langle A^b \rangle\rangle \phi\, \mathcal{U}\, \psi$)
24: **else**
25: $ActA \leftarrow \{\sigma \in D_A(s(n)) \mid c(\sigma) \leq e(n)\}$
26: **for** $\sigma \in ActA$ **do**
27: $P \leftarrow \{node(n, \sigma, s') \mid s' \in out(s(n), \sigma)\}$
28: **if** UNTIL($P \circ tl(B)$, C, $\langle\langle A^b \rangle\rangle \phi\, \mathcal{U}\, \psi$) **then**
29: **return** *true*
30: **return** *false*

on the path to n (not including n), and $p_{k-1}(n)$ returns the last $k-1$ nodes. σ_a is the action performed by agent a in the joint action σ and $a_a(n)$ is the action taken by a in the joint action to reach n. Line 14 constructs a set of joint communication actions such that each agent a for which $p_{k-1}(n) \cdot n \sim_a p_k(n')$ for some n' (where k is the bound on recall), chooses communication action $a_a(n')$. That is, if the last k states on the current history (including $s(n)$) are $\sim_a$ to the last k states on the path to a node n', then a must choose the same communication action as it did in $a(n')$ (the joint action taken by A to reach $s(n')$). Otherwise agents in A choose communication actions constrained only by cost. For each such joint communication action, we generate a new node for each possible outcome state of communication, and call UNTIL recursively to continue the search, pushing the nodes corresponding to the successor states onto the stack of open nodes (lines 13–17). This makes the choice of communication action a strategic decision similar to that for ontic actions (lines 26–29) and the algorithm could be refactored accordingly.[2]

THEOREM 5.2. *If Algorithm 5 returns true, then the Until formula is true.*

The proof is straightforward; since any bounded uniform strategy is a uniform strategy, the formula is satisfied under the RB± ATSEL$_c$ semantics. However if the algorithm returns false, this does not mean that a perfect recall uniform strategy does not exist. It may simply be that the bound k is not sufficient to realise such a strategy, or that it is not necessary for uniformity to achieve distributed knowledge (there is a cheaper, in terms of communication cost, uniform strategy).

6 RELATED WORK

There is a large body of work on ATL with imperfect information, e.g., [1, 9–13, 15–18], and extensions of ATL with resources are considered in e.g., [4, 7, 8].

We build on the work of [3] where model checking algorithms were proposed for RB±ATSEL with coalition uniform strategies. It is known that the model checking problem for ATL with imperfect information and perfect recall is decidable for uniformity with respect to distributed knowledge [9, 11], which is a special case of coalition uniformity. We are not aware of any work which uses this result in combination with explicit communication to make the model checking problem decidable for perfect recall uniform strategies, as we do in this paper.

There are other approaches to making the model-checking problem decidable in the presence of imperfect information and perfect recall. In [14], coalitions with a single indistinguishability relation are shown to behave as a single agent, and the model checking problem for the resulting variant of ATL is decidable as a consequence. Another approach considers a class of models with only 'public actions' (every agent can distinguish the states resulting from the execution of two different joint actions) [6]. In communication models, communication actions are public actions for the coalition, but ontic actions are not.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a modification of the logic RB±ATSEL defined in [3] where model checking is decidable for *uniform strategies*. Our approach works only for a special kind of models which we call *communication models*. In communication models, agents always perform a communication step before selecting and executing actions, and follow a specific communication strategy (sending their entire knowledge to all other agents in the coalition). We sketch a sound but incomplete approach to generating a potentially reduced cost communication strategy.

ACKNOWLEDGEMENTS

We would like to thank the reviewers of AAMAS 2019 and a previous version of this paper for their insightful comments.

[2] We did not do this to facilitate comparison with Algorithm 3.

REFERENCES

[1] T. Ågotnes. 2006. Action and Knowledge in Alternating-Time Temporal Logic. *Synthese* 149, 2 (2006), 375–407.

[2] Natasha Alechina, Nils Bulling, Stéphane Demri, and Brian Logan. 2016. On the Complexity of Resource-Bounded Logics. In *Reachability Problems - 10th International Workshop, RP 2016, Proceedings (Lecture Notes in Computer Science)*, Kim Guldstrand Larsen, Igor Potapov, and Jirí Srba (Eds.), Vol. 9899. Springer, 36–50.

[3] Natasha Alechina, Mehdi Dastani, and Brian Logan. 2016. Verifying Existence of Resource-Bounded Coalition Uniform Strategies. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, Subbarao Kambhampati (Ed.). IJCAI/AAAI, 24–30.

[4] N. Alechina, B. Logan, H. N. Nguyen, and F. Raimondi. 2014. Decidable Model-Checking for a Resource Logic with Production of Resources. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*. 9–14.

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

[6] Francesco Belardinelli, Alessio Lomuscio, Aniello Murano, and Sasha Rubin. 2017. Verification of Multi-agent Systems with Imperfect Information and Public Actions. In *Proceedings of the 16th Conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2017*, Kate Larson, Michael Winikoff, Sanmay Das, and Edmund H. Durfee (Eds.). ACM, 1268–1276.

[7] N. Bulling and B. Farwer. 2010. On the (Un-)Decidability of Model Checking Resource-Bounded Agents. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*. 567–572.

[8] N. Bulling and V. Goranko. 2013. How to Be Both Rich and Happy: Combining Quantitative and Qualitative Strategic Reasoning about Multi-Player Games (Extended Abstract). In *Proceedings 1st International Workshop on Strategic Reasoning, SR 2013 (EPTCS)*, Vol. 112. 33–41.

[9] Catalin Dima, Constantin Enea, and Dimitar P. Guelev. 2010. Model-Checking an Alternating-time Temporal Logic with Knowledge, Imperfect Information, Perfect Recall and Communicating Coalitions. In *Proceedings First Symposium on Games, Automata, Logic, and Formal Verification, GANDALF 2010 (EPTCS)*, Vol. 25. 103–117.

[10] C. Dima and F. L. Tiplea. 2011. Model-checking ATL under Imperfect Information and Perfect Recall Semantics is Undecidable. *CoRR* abs/1102.4225 (2011). http://arxiv.org/abs/1102.4225

[11] D. P. Guelev, C. Dima, and C. Enea. 2011. An alternating-time temporal logic with knowledge, perfect recall and past: axiomatisation and model-checking. *Journal of Applied Non-Classical Logics* 21, 1 (2011), 93–131.

[12] Wojciech Jamroga and Thomas Ågotnes. 2007. Constructive knowledge: what agents can achieve under imperfect information. *Journal of Applied Non-Classical Logics* 17, 4 (2007), 423–475. https://doi.org/10.3166/jancl.17.423-475

[13] W. Jamroga and W. van der Hoek. 2004. Agents that Know How to Play. *Fundam. Inform.* 63, 2-3 (2004), 185–219.

[14] Piotr Kazmierczak, Thomas Ågotnes, and Wojciech Jamroga. 2014. Multi-agency Is Coordination and (Limited) Communication. In *Proceedings of PRIMA 2014: Principles and Practice of Multi-Agent Systems - 17th International Conference (Lecture Notes in Computer Science)*, Hoa Khanh Dam, Jeremy V. Pitt, Yang Xu, Guido Governatori, and Takayuki Ito (Eds.), Vol. 8861. Springer, 91–106.

[15] A. Lomuscio, H. Qu, and F. Raimondi. 2009. MCMAS: A Model Checker for the Verification of Multi-Agent Systems. In *Proceedings of the 21st International Conference on Computer Aided Verification (CAV 2009) (LNCS)*, Vol. 5643. Springer, 682–688.

[16] A. Lomuscio and F. Raimondi. 2006. Model checking knowledge, strategies, and games in multi-agent systems. In *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*. ACM, 161–168.

[17] W. van der Hoek, A. Lomuscio, and M. Wooldridge. 2006. On the complexity of practical ATL model checking. In *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*. ACM, 201–208.

[18] W. van der Hoek and M. Wooldridge. 2002. Tractable multiagent planning for epistemic goals. In *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002*. ACM, 1167–1174.

[19] Steen Vester. 2013. Alternating-time temporal logic with finite-memory strategies. In *Proceedings Fourth International Symposium on Games, Automata, Logics and Formal Verification, (GandALF 2013) (EPTCS)*, Vol. 119. 194–207. https://doi.org/10.4204/EPTCS.119