Synthesising Minimum Cost Dynamic Norms

Natasha Alechina,[1,3] **Brian Logan,**[2,3] **Giuseppe Perelli** [4]

[1]Open University Netherlands
[2]University of Aberdeen
[3]University of Utrecht
[4] Sapienza University of Rome

Abstract

A key problem in the design of normative multi-agent systems is the cost of enforcing a norm (for the system operator) or complying with the norm (for the system users). If the cost is too high, ensuring compliant behavior may be uneconomic, or users may be deterred from participating in the MAS. In this paper, we consider the problem of synthesizing *minimum cost* dynamic norms to satisfy a system-level objective specified in Alternating Time Temporal Logic with Strategy Contexts (ATL^*_{sc}). We show that synthesizing a dynamic norm under a bound on the cost of any prohibited set of actions has the same complexity as synthesizing arbitrary norms. We also show that synthesizing norms that minimize the average cost of the prohibited set of actions is unsolvable; however, synthesizing ϵ-optimal norms is possible.

1 Introduction

In normative multi-agent systems, system-level objectives are realized through norms [Chopra *et al.*, 2018]. A *norm* expresses a pattern of desired or undesired behaviour that achieves or violates a system objective. For example, a norm may specify that it is not permitted to travel on public transport without valid ticket (to ensure the financial viability of the transport authority), or to smoke in a public place (to reduce the health risks to others), etc.

A key problem in the design of normative multi-agent systems is the *cost* of enforcing or complying with the norm. Enforcing norms typically has a cost to the system (normative organisation), e.g., the cost of installing ticket barriers to control access to a public transport system. Different actions may be more or less costly to control, e.g., it may be more costly to control access to an open-air concert than to a metro station. Complying with a norm also has a cost (loss of utility) for agents which may be different for different agents; for example, a norm prohibiting smoking in public places may result in no loss of utility for an agent who does not smoke, but a significant loss of utility for an agent who does smoke. If the costs are too high, ensuring compliant behavior may be uneconomic, or users may be deterred from participating in the MAS.

While the implementation of norms in a MAS has been extensively studied, e.g., [Meyer and Wieringa, 1993; Grossi *et al.*, 2006; Ågotnes *et al.*, 2007; Astefanoaei *et al.*, 2009; Dennis *et al.*, 2010; Dastani *et al.*, 2013; Ågotnes *et al.*, 2010; Bulling *et al.*, 2013; Dignum *et al.*, 2004; Boella and van der Torre, 2004; Boella *et al.*, 2008; Tinnemeier *et al.*, 2009; Alechina *et al.*, 2013], there has been less work on how norms can be automatically synthesized to meet an objective. Synthesis of norms was first introduced in [Shoham and Tennenholtz, 1995], and also studied in [Fitoussi and Tennenholtz, 2000; Christelis and Rovatsos, 2009; Corapi *et al.*, 2011; Morales *et al.*, 2015a; Morales *et al.*, 2018]. In the context of logical objectives, norm synthesis has been studied in, for example, in [van der Hoek *et al.*, 2007; Bulling and Dastani, 2016; Huang *et al.*, 2016; Perelli, 2019; Alechina *et al.*, 2022]. However, with the exception of [Perelli, 2019] where a related problem of norm optimization was introduced, this work has not considered the cost of the synthesized norm(s).

In this paper, we consider the problem of synthesizing *minimum cost* dynamic norms to satisfy a system-level objective specified in Alternating Time Temporal Logic with Strategy Contexts (ATL^*_{sc}). We focus on dynamic norms [Huang *et al.*, 2016], as these are more flexible than 'static' (state based) norms in allowing constraints to be enforced on the actions of agents based on their history of interaction with the MAS. For example, in the interests of fairness, a norm may specify that if an agent was previously prohibited from performing an action, it should be allowed to perform the action in the current state. Our notion of the cost of prohibiting an action is quite general, in the sense that it admits any assignment of costs to prohibiting sets of actions, and can express both the loss of utility to the agents and/or the cost of enforcing the norm to the normative organisation. For example, if all actions have the same cost, a minimum cost norm can be seen as "maximally permissive", i.e., it prohibits the minimum number of actions necessary to achieve the system objective. On the other hand, if agents place greater value on being able to perform some actions than others, a minimum cost norm may be one that maximizes social welfare while achieving the objective. We use ATL^*_{sc} to express system objectives as this allows us to specify the strategic abilities of (groups of) agents in the MAS, both to state that a particular group of agents should (or should not) be able to achieve a particular goal, as well as stating that given the individual goals of agents, there is a Nash equilibrium. We consider norm synthesis for two natural notions of norm cost, *maximum single step cost* and *average step cost*. The maximum single step cost is the maximum cost of any prohibited set of actions and is relevant when the system designer wishes to minimise the maximum loss of utility/cost of enforcement incurred due to a norm. The average step cost is the average cost of the prohibited set of actions over an infinite run and and is relevant when the aim is to minimise the average loss of utility/cost of enforcement resulting from the norm. Both notions are useful depending on the context: for example, a short lockdown during a pandemic may have a high maximum single step cost (impacting individual liberties), but low average step cost (impacting economic activity). We show that synthesizing a dynamic norm under a bound on the maximum single step cost has the same complexity as synthesizing arbitrary norms in [Alechina *et al.*, 2022] (i.e., $(h+1)$-EXPTIME for an existential $\mathrm{ATL}^*_{\mathsf{sc}}$ formula with quantifier alternation number h, and in $(h+2)$-EXPTIME for a universal $\mathrm{ATL}^*_{\mathsf{sc}}$ formula). We also show that synthesizing a dynamic norm that minimizes the average step cost does not always have a solution as it requires infinite memory. However, synthesizing ϵ-optimal norms is possible for a restricted class of $\mathrm{ATL}^*_{\mathsf{sc}}$ objectives.

2 Framework

In this section we introduce the definitions needed to formalise reasoning about dynamic norms in multi-agent systems, largely following [Alechina *et al.*, 2022].

2.1 System Models

We assume a multi-agent system coordinated by zero or more dynamic norms, which we formalize as a particular kind of game.

Definition 1 (k-normed Multi-Agent System)**.** *A k-normed Multi-Agent System (k-MAS) $\mathcal{G}$ is a tuple* $\langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{AP}, \mathsf{Cap}, \mathsf{tr}, (\mathrm{Nrm}_i)_{i\le k}, \vec{\mathsf{q}}_0, (\eta_i)_{i\le k}, (\mathsf{illegal}_i)_{i\le k}\rangle$ *where:*

- $\mathrm{Ag} = \{1,\dots,N\}$ *is a finite set of N* agents*, denoted by natural numbers;*
- Ac *is a finite set of* actions *that agents can perform (in some state of the environment);*
- AP *is a finite set of* atomic propositions*; an assignment of truth values to* AP *determines environment states of the system;*
- $\mathsf{Cap} : \mathrm{Ag} \times 2^{\mathrm{AP}} \to 2^{\mathrm{Ac}}$ *is a* capability function *that assigns to each agent in each environment state the set of actions it is capable of performing in that state*
- $\mathsf{tr} : 2^{\mathrm{AP}} \times \mathrm{Ac}^{\mathrm{Ag}} \to 2^{\mathrm{AP}}$ *is a* transition function *that determines the next state of the environment given the current state of the environment and the actions performed by the agents;*
- Nrm_i *is a finite set of* normative states*, one for each* $i \le k$*;* $\vec{\mathrm{Nrm}} = \mathrm{Nrm}_1 \times \ldots \times \mathrm{Nrm}_k$*, is the* normative vector state space *(the set of tuples containing the state of each norm);*
- $\vec{\mathsf{q}}_0 \in \vec{\mathrm{Nrm}}$ *is a designated* initial normative state*;*
- $\eta_i : \mathrm{Nrm}_i \times 2^{\mathrm{AP}} \to \mathrm{Nrm}_i$ *is a* normative function *that determines the next state of a norm given the current state of the norm and the environment;*
- $\mathsf{illegal}_i : \mathrm{Nrm}_i \times 2^{\mathrm{AP}} \to 2^{\mathrm{Ac}^- \times \mathrm{Ag}}$ *is the* illegality function *that returns a set pairs of actions and agents that are illegal given the current state of a norm and the environment.*

Unlike [Alechina *et al.*, 2022], we assume that there is a distinguished action $noop \in \mathrm{Ac}$ which is always available to every agent and is never illegal, i.e., intuitively it is always possible for an agent to do nothing. We denote by Ac^- the set $\mathrm{Ac} \setminus \{noop\}$. This change does not affect the results in [Alechina *et al.*, 2022].

We illustrate a k-MAS with a simple example of a normative multi-agent system.

Example 1 (k-MAS)**.** *Consider a traffic intersection with two agents. Agent 1 moves North-South and agent 2 moves East-West. Each agent has two actions, go and noop.* $\mathrm{AP} = \{east,\ west,\ north,\ south,\ crash\}$*. Intuitively, east holds when agent 2 is in the East, etc. The action noop does not change an agent's position; go moves agent 1 from North to South or back, and agent 2 from East to West or back (if the other agent does noop); go, go results in a crash. The initial state is when agent 1 is in the North and 2 is in the East. There is one norm ($k=1$) with two states, q_1 and q_2, which alternate (on any input). For any state of the environment, in q_1 there is a single illegal action $(go, 2)$ and in q_2 there is a single illegal action $(go, 1)$. The initial norm state is q_1. The norm acts as a traffic light, allowing only one agent to move at every step. The states of the norm can be seen as encoding the history in the game so far and hence whose turn it is to go at the current step.*

Evolution Starting from some initial state of the environment (a set of atomic propositions) π_0 and the initial vector of normative states $\vec{\mathsf{q}}_0$, a game moves forward according to the transition function, which given the current state of the environment and an action tuple $\vec{a} \in \mathrm{Ac}^{\mathrm{Ag}}$, determines the next state of the environment (assignment to the propositions in AP). Simultaneously, each normative component is updated by the corresponding normative function, which, given the current state of the norm and the current environment state, determines the next state of the norm.

Configuration, Illegal Actions, Available Actions A *configuration* of $\mathcal{G}$ is a tuple $c = (\pi, \vec{\mathsf{q}}) \in 2^{\mathrm{AP}} \times \vec{\mathrm{Nrm}}$ (a tuple consisting of the state of the environment and the states of the norms). The set of actions that are made illegal for agent j by the i-th normative component is denoted by $\mathsf{illegal}_i(\mathsf{q}_i, \pi, j) \doteq \{a \in Ac : (a, j) \in \mathsf{illegal}_i(\mathsf{q}_i, \pi)\}$. The set of actions available to agent j in configuration c, where $\vec{\mathsf{q}}^i$ is the i-th component of $\vec{\mathsf{q}}$, is denoted by $\mathsf{Avl}_{\mathcal{G}}(c, j) \doteq \mathsf{Cap}(j, \pi) \setminus (\cup_{i\le k}\mathsf{illegal}_i(\mathsf{q}_i, \pi, j))$. The set $\mathsf{Avl}_{\mathcal{G}}(c) \doteq \mathsf{Avl}_{\mathcal{G}}(c, 1) \times \ldots \times \mathsf{Avl}_{\mathcal{G}}(c, N)$ denotes the action vectors that are available in a configuration c.

In Example 1, the initial configuration is $(\{north, east\}, \mathsf{q}_1)$, and agent 1 has actions $go, noop$ available, while agent 2 only has available action $noop$, since $(go, 2)$ is illegal.

At each configuration $c = (\pi, \vec{\mathsf{q}})$, each agent j can select only an action $a^j \in \mathsf{Avl}_{\mathcal{G}}(c, j)$. Once each agent j has chosen an available action a^j and the corresponding action vector $\vec{a} = (a^1, \ldots, a^k)$ is formed, the system moves its components forward to the configuration $(\pi', \vec{\mathsf{q}}')$, with $\pi' = \mathsf{tr}(\pi, \vec{a})$ and $\vec{\mathsf{q}}' = (\eta_1(\mathsf{q}^1, \pi), \ldots, \eta_k(\mathsf{q}^k, \pi))$.

Runs of the system A *legal run*, or simply *run* is an infinite sequence $r \in (2^{\mathrm{AP}} \times \vec{\mathrm{Nrm}})^\omega$ such that, for each $n \in \mathbb{N}$, there exists an action vector $\vec{a}_n \in \mathsf{Avl}_{\mathcal{G}}(r_n)$, such that

$$r_{n+1} = (\mathsf{tr}(\pi_n, \vec{a}_n), \eta_1(\mathsf{q}_n^1, \pi_n), \ldots, \eta_k(\mathsf{q}_n^k, \pi_n))$$

with $r_n = (\pi_n, \vec{\mathsf{q}}_n)$. We use the notation $r_{\leq n}$ to denote the prefix of r up to and including r_n. Similarly, $r_{\geq n}$ is the suffix of r starting from r_n. $c \xrightarrow{\vec{a}} c'$ is used to denote that the action vector $\vec{a}$ determines a transition from configuration c to configuration c'. A run is *initial* if it starts from the initial configuration, that is, $r_0 = (\pi_0, \vec{\mathsf{q}}_0)$.

A possible run of the system in Example 1 is $(\{north, east\}, \mathsf{q}_1)$, $(\{south, east\}, \mathsf{q}_2)$, $(\{south, east\}, \mathsf{q}_1)$, $(\{north, east\}, \mathsf{q}_2), \ldots$ where agent 1 executes *go* whenever it can, and agent 2 always executes *noop* even when *go* is available.

Strategy A *strategy* for agent j in the game $\mathcal{G}$ is a Mealy machine of the form

$$\sigma_j = (S_j, s_j^0, 2^{\mathrm{AP}} \times \vec{\mathrm{Nrm}}, \mathrm{Ac}, \delta_j, \tau_j).$$

For each internal state $s \in S_j$ and a configuration $c = (\pi, \vec{\mathsf{q}})$ of $\mathcal{G}$, a strategy selects an action in Ac determined by $\tau_j(s, (\pi, \vec{\mathsf{q}})) \in \mathrm{Ac}$ and updates its internal state $\delta_j(s, (\pi, \vec{\mathsf{q}})) \in S_j$ accordingly. Only the strategies that comply with the *normative requirements* specified by the game are available to the agents. A strategy σ_j is *legal* with respect to $\mathcal{G}$ if, and only if, $\tau_j(s, (\pi, \vec{\mathsf{q}})) \in \mathsf{Avl}(\vec{\mathsf{q}}, \pi, j)$. From now on, we restrict our attention to legal strategies, and, unless otherwise stated, we refer to them simply as strategies. Moreover, for simplicity, for a given strategy σ_j and a finite sequence $\hat{r} \in (2^{\mathrm{AP}} \times \vec{\mathrm{Nrm}})^*$, by $\sigma_j(\hat{r}) \in \mathrm{Ac}$ we denote the action determined by the function τ_j in σ_j after the sequence $\hat{r}$ has been fed to the internal transition function δ_j.

A simple strategy for agent 2 would be to always execute *noop*. This requires a Mealy machine with a single state, that on any input returns *noop* and loops back to the same state. A strategy for agent 1 would be executing *go* on input of q_1 (regardless of the state of the world, that is, of the propositional assignment), and *noop* on input of q_2.

Note that the restriction to regular strategies representable by Mealy machines is without loss of generality, as when the specification language is ω-regular, synthesizing a strategy satisfying the specification is equivalent to synthesizing a regular strategy [Pnueli and Rosner, 1989].

Runs compatible with strategies A run r is *compatible* with σ_A (where A is a set of agents) if, for every $n \in \mathbb{N}$, it holds that there exists an action vector $\vec{a}_n$, with $a_n^j = \sigma_j(r_{\leq n})$ for each $j \in A$ and $\sigma_j \in \sigma_A$, such that r_{n+1} is obtained from r_n by applying $\vec{a}_n$. In other words, a run r is compatible with σ_A if it can be generated when the agents in A play according to their respective strategies. The set of runs starting from a given configuration c and compatible with σ_A is denoted by $\mathsf{out}_{\mathcal{G}}(c, \sigma_A)$. Observe that the set of runs of a given k-MAS $\mathcal{G}$ starting from a configuration c, sometimes denoted $\mathsf{Paths}_{\mathcal{G}}(c)$, can also be written as $\mathsf{out}_{\mathcal{G}}(c, \emptyset)$. When it is clear from the context we use $\mathsf{Paths}(c)$ or $\mathsf{out}(c, \sigma_A)$.

2.2 $\mathbf{ATL}^*_{\mathsf{sc}}$– Alternating-Time Temporal Logic with Strategy Contexts

We use an extension of Alternating-Time Temporal Logic (ATL^*) [Alur *et al.*, 2002], Alternating-Time Temporal Logic with Strategy Contexts ($\mathrm{ATL}^*_{\mathsf{sc}}$) introduced in [Da Costa Lopes *et al.*, 2010] to express system objectives. We refer the reader to [Laroussinie and Markey, 2015] for a detailed discussion of $\mathrm{ATL}^*_{\mathsf{sc}}$.

$\mathrm{ATL}^*_{\mathsf{sc}}$ formulas are built inductively from the set of atomic propositions AP and agents Ag. There are two types of $\mathrm{ATL}^*_{\mathsf{sc}}$ formulas, state formulas and path formulas. State formulas are built by using the following grammar, where $p \in \mathrm{AP}$, $A \subseteq \mathrm{Ag}$ and ψ is $\mathrm{ATL}^*_{\mathsf{sc}}$ path formula:

$$\phi := p \mid \neg\phi \mid \phi \wedge \phi \mid \langle\cdot A\cdot\rangle\psi \mid \rangle\cdot A\cdot\langle\psi$$

Path formulas are built using the following grammar, where ϕ is a state formula of $\mathrm{ATL}^*_{\mathsf{sc}}$:

$$\psi := \phi \mid \neg\psi \mid \psi \wedge \psi \mid \mathtt{X}\psi \mid \psi\mathtt{U}\psi$$

We also use the following syntactic sugar notation: $\varphi_1 \vee \varphi_2 \doteq \neg(\neg\varphi_1 \wedge \neg\varphi_2)$, $\varphi_1 \rightarrow \varphi_2 \doteq \neg\varphi_1 \vee \varphi_2$, $[\cdot A\cdot]\varphi \doteq \neg\langle\cdot A\cdot\rangle\neg\varphi$, $\mathtt{F}\,\varphi \doteq \mathtt{true}\,\mathtt{U}\,\varphi$, and $\mathtt{G}\,\varphi \doteq \neg\mathtt{F}\neg\varphi$.

Formulas of $\mathrm{ATL}^*_{\mathsf{sc}}$ are evaluated relatively to a *strategy context:* a fixed strategy σ_B for a group of agents B. The formula $\langle\cdot A\cdot\rangle\psi$ means that the agents in A have a collective strategy such that whatever the agents $\mathrm{Ag} \setminus A$ do (provided the strategies in σ_B are fixed), the resulting outcomes satisfy ψ. Conversely, the formula $\rangle\cdot A\cdot\langle\psi$ means that whenever the strategies for agents in coalition A are removed from σ_B, the resulting outcomes satisfy ψ. (Note that the strategy quantifier of ATL^* can be defined as $\langle\langle A\rangle\rangle \doteq \rangle\cdot\mathrm{Ag}\cdot\langle\,\langle\cdot A\cdot\rangle$.) In addition, the formula $[\cdot A\cdot]\psi$ means that the agents in A do not have a strategy to prevent ψ given σ_B is fixed for agents in B.

For a given k-MAS $\mathcal{G}$, a run r over it, and a strategy context σ_B, the semantics of an $\mathrm{ATL}^*_{\mathsf{sc}}$ path formula ψ, denoted $\mathcal{G}, r \models_{\sigma_B} \psi$, is given recursively as follows:

- $\mathcal{G}, r \models_{\sigma_B} \phi$ iff $\mathcal{G}, r_0 \models_{\sigma_B} \phi$, where ϕ is a state formula;
- $\mathcal{G}, r \models_{\sigma_B} \neg\psi$ iff $\mathcal{G}, r \not\models_{\sigma_B} \psi$;
- $\mathcal{G}, r \models_{\sigma_B} \psi_1 \wedge \psi_2$ iff $\mathcal{G}, r \models_{\sigma_B} \psi_1$ and $\mathcal{G}, r \models_{\sigma_B} \psi_2$;
- $\mathcal{G}, r \models_{\sigma_B} \mathtt{X}\psi$ iff $\mathcal{G}, r_{\geq 1} \models_{\sigma_B} \psi$;
- $\mathcal{G}, r \models_{\sigma_B} \psi_1\mathtt{U}\psi_2$ iff there exists $j \in \mathbb{N}$ such that $\mathcal{G}, r_{\geq i} \models_{\sigma_B} \psi_1$, for all $i < j$, and $\mathcal{G}, r_{\geq j} \models_{\sigma_B} \psi_2$.

The semantics of an $\mathrm{ATL}^*_{\mathsf{sc}}$ state formula ϕ is defined relative to a state r_i on a run r as follows (we omit the cases for $\neg$ and $\wedge$ as they are obvious):

- $\mathcal{G}, r_i \models_{\sigma_B} p$ iff $p \in \pi_i$, with $r_i = (\pi_i, \vec{\mathsf{q}}_i)$ for some $\vec{\mathsf{q}}_i \in \vec{\mathrm{Nrm}}$;

- $\mathcal{G}, r_i \models_{\sigma_B} \langle\cdot A\cdot\rangle\psi$ iff there is a strategy σ_A such that $\mathcal{G}, r' \models_{\sigma_B \circ \sigma_A} \psi$ for all $r' \in \mathtt{out}(r_0, \sigma_B \circ \sigma_A)$, where $\sigma_B \circ \sigma_A \doteq \sigma_B \cup \sigma_{A\setminus B}$ denotes the set of strategies obtained from σ_B by adding strategies of σ_A that are for agents in A but not in B.
- $\mathcal{G}, r_i \models_{\sigma_B} \rangle\cdot A\cdot\langle\psi$ iff $\mathcal{G}, r_i \models_{\sigma_{B\setminus A}} \psi$.

In Example 1, a simple objective is

$$\langle\cdot 1\cdot\rangle(\mathtt{G}\neg crash \wedge \mathtt{G}(north \rightarrow \mathtt{F}south) \wedge \mathtt{G}(south \rightarrow \mathtt{F}north)).$$

It states that agent 1 has a strategy to avoid a crash and to be always able to cross from North to South and back again. In the system in Example 1, such a strategy exists because of the presence of the norm that stops agent 2 from executing *go* at every other step (so agent 1 can execute *go* without causing a crash). Without the norm, however, agent 1 alone cannot ensure both the absence of crashing and being able to cross.

The objective above can be expressed in ATL*. However, ATL$^*_{\mathsf{sc}}$ is more expressive, and can specify useful objectives that are not expressible in ATL*, for example, the existence of Nash equilibrium. If each agent $j \in \mathrm{Ag}$ has a goal expressed by a temporal formula ψ_j, the existence of Nash equilibrium can be expressed as $\langle\cdot\mathrm{Ag}\cdot\rangle(\bigwedge_{j\in\mathrm{Ag}}(\neg\psi_j \rightarrow \neg\langle\cdot j\cdot\rangle\psi_j))$ (see [Laroussinie and Markey, 2015]; the formula says that there exists a joint strategy such that if j's goal is not satisfied, then no deviation from this strategy satisfies it).

The *model-checking problem* for ATL$^*_{\mathsf{sc}}$ is: given a structure $\mathcal{G}$, a run r, a strategy context σ_B and an ATL$^*_{\mathsf{sc}}$ formula φ, does it hold that $\mathcal{G}, r \models_{\sigma_B} \varphi$? The complexity of ATL$^*_{\mathsf{sc}}$ model-checking depends on the nesting of quantifiers in a formula, and in particular on how many times existential quantifiers alternate with universal quantifiers. The *quantifier alternation number* of a ATL$^*_{\mathsf{sc}}$ formula φ is the number of times an existential quantification $\langle\cdots\rangle$ is followed by a universal one $[\cdots]$, and vice-versa.

Theorem 1. *[Laroussinie and Markey, 2015, Corollary 14] The model-checking problem for ATL$^*_{\mathsf{sc}}$ is (h+1)EXPTIME-complete for a formula with quantifier alternation number h.*

Although this result was established for concurrent game structures (essentially, 0-MAS without normative components), it also holds for k-MAS because every k-MAS corresponds to a concurrent game structure of size polynomial in the size of the k-MAS, however exponential in k, that is the number of normative components.

2.3 Norms

As in [Alechina *et al.*, 2022], we define a *Norm* over a k-MAS $\mathcal{G}$ as a Mealy machine of the form: $\mathcal{N} = \langle \mathrm{Nrm}, \mathsf{q}_0, 2^{\mathrm{AP}}, 2^{\mathrm{Ac}\times\mathrm{Ag}}, \eta, \mathsf{illegal}\rangle$. A norm takes an environment state as input and returns a set pairs of actions and agents that are illegal given the current state of a norm and the environment. $\mathcal{N}$ is well-defined on every k-MAS $\mathcal{G}$ having the same set Ag of agents, Ac of actions and AP of propositions.

Consider a norm $\mathcal{N}_{k+1}$ whose components are all indexed with $k+1$. $\mathcal{N}_{k+1}$ can be implemented on a k-MAS $\mathcal{G}$ to obtain a $(k+1)$-MAS $\mathcal{G} \oplus \mathcal{N}_{k+1} = \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{AP}, \mathsf{Cap}, \mathsf{tr}, (\mathrm{Nrm}_i)_{i\le k+1}, \vec{\mathsf{q}}_0, (\eta_i)_{i\le k+1}, (\mathsf{illegal}_i)_{i\le k+1}\rangle$ containing an extra normative state component, which are the states of $\mathcal{N}_{k+1}$, and whose evolution is determined by the normative function η_{k+1}.

For every configuration c in the original game and its extension c' with the state of $\mathcal{N}_{k+1}$, it holds that $\mathsf{Avl}_{\mathcal{G}\oplus\mathcal{N}_{k+1}}(c', j) \subseteq \mathsf{Avl}_{\mathcal{G}}(c, j)$ for every agent $j \in \mathrm{Ag}$. Intuitively, $\mathcal{N}_{k+1}$ introduces more restrictions on the actions for the agents when implemented in a given k-MAS $\mathcal{G}$. Note that all norms allow the agents to perform at least *noop*, so there will always be an action available to each agent regardless of how many norms we combine.

Observe also that every normative state component i of a k-MAS $\mathcal{G}$ can be regarded as a norm $\mathcal{N}_i = \langle \mathrm{Nrm}_i, e_0^i, 2^{\mathrm{AP}}, 2^{\mathrm{Ac}\times\mathrm{Ag}}, \eta_i, \mathsf{illegal}_i\rangle$ and so $\mathcal{G}$ can be obtained from a 0-MAS where the norms $\mathcal{N}_1 \ldots, \mathcal{N}_k$ have been applied one by one. The norm synthesis problem is as follows (where $\models_\emptyset$ means true in an empty context).

Definition 2 (Norm Synthesis)**.** *For a given k-MAS $\mathcal{G}$ and an ATL$^*_{\mathsf{sc}}$ formula φ, determine whether there exists a norm $\mathcal{N}_{k+1}$ such that $\mathcal{G} \oplus \mathcal{N}_{k+1} \models_\emptyset \varphi$.*

In [Alechina *et al.*, 2022], it was shown that the norm synthesis problem for ATL* objectives is decidable in $(h+2)$-EXPTIME where h is defined before Theorem 1.

3 Cost of Prohibiting Actions

Alechina *et al.* do not consider the cost of the synthesized norm. In this section, we define the cost of prohibiting actions to agents which we use below to define minimum cost norms. Our approach is general, in the sense that it admits any assignment of costs to prohibiting sets of actions, and can express both the loss of welfare or utility to the agents and/or the cost of enforcing the norm to the normative organisation. However we only consider static costs, and defer consideration of dynamic costs to future work.

First we introduce the notion of a cost of prohibiting an agent to execute a particular action:

$$cost : \mathrm{Ac} \times \mathrm{Ag} \longrightarrow \mathbb{N}$$

The simplest cost function would be to assign the same cost of 1 to each pair of an action and an agent. However, some actions may be more 'expensive' to prohibit or some agents may suffer greater loss of utility from prohibiting the action. For example, a cost of (a, j) may depend on whether $j \in A$ for some group of agents A.

Example 2. *In the Example 1 scenario, assume that it is significantly more costly to prevent agent 1 from moving; e.g., if agent 1 is an ambulance. Then we could set $cost(go, 1) = 100$ and $cost(go, 2) = 1$.*

A cost function over pairs (a, j) of actions and agents can be lifted to a cost function over sets of action, agent pairs as:

$$cost : 2^{\mathrm{Ac}\times\mathrm{Ag}} \longrightarrow \mathbb{N}$$

Depending on the application, for $X \in 2^{\mathrm{Ac}\times\mathrm{Ag}}$, different lifted cost functions may be appropriate.

Example 3. *If $cost(a, j)$ is 1 for all actions a and agents j, a natural measure of cost is to sum up the costs of prohibited actions across all agents: $cost(X) = \sum_{(a,j)\in X} cost(a, j)$*

Example 4. *If we aim to avoid all costs falling on only a small number of agents, a more natural measure of cost may be to sum the cost of prohibited actions for each agent:* $cost(X) = \max_{a \in \mathrm{Ac}} \left(\sum_{(a,j) \in X} cost(a, j) \right)$.

Example 5. *Prohibitions such as quarantine restrictions that apply to large numbers of agents may be compared based on the most costly prohibited action, regardless of the identity or number of agents:* $cost(X) = \max_{(a,j) \in X}(\{cost(a, j)\})$.

So far, we have considered the costs of prohibiting actions and sets of actions. However, in a run of the system, a norm specifies a *sequence* of prohibited sets of of actions for each agent. In what follows, we consider two natural interpretations of the cost of a norm, the *maximum cost* incurred on a run, and the *average cost* incurred on a run.

4 Maximum Single Step Cost Optimization

The maximum cost of any step along a run is a good measure of the prohibitiveness of a norm; for example, it represents the maximum loss of utility or cost of enforcement incurred due the norm.

Definition 3 (Maximum Single Step Cost)**.** *Let* $cost : 2^{\mathrm{Ac} \times \mathrm{Ag}} \longrightarrow \mathbb{N}$ *be given. The* per step cost of a norm $\mathcal{N} = \langle \mathrm{Nrm}, \mathsf{q}_0, 2^{\mathrm{AP}}, 2^{\mathrm{Ac} \times \mathrm{Ag}}, \eta, \mathsf{illegal} \rangle$ *is*

$$max(\{cost(\mathsf{illegal}(n, \pi)) \mid n \in \mathrm{Nrm},\ \pi \in 2^{\mathrm{AP}}\})$$

i.e., the cost of the most expensive set of prohibited actions for this norm.

A *maximum single step cost least prohibitive norm* for a given k-MAS and objective φ is a norm that makes the objective true when added to k-MAS and there is no norm with strictly lower single step cost which makes the objective true. This leads to the following decision problem:

Definition 4 (Maximum Single Step Cost Optimization)**.** *Given* $cost : 2^{\mathrm{Ac} \times \mathrm{Ag}} \longrightarrow \mathbb{N}$*, k-MAS and objective φ, is there a norm $\mathcal{N}$ of maximum step cost at most m such that $\mathcal{N}$ makes φ true when added to the k-MAS.*

Example 6. *If* $cost(go, i) = 1$ *for* $i \in \{1, 2\}$ *in the Example 1 scenario, the norm in that example is a maximum single step cost least prohibitive norm which ensures the objective*

$$\langle \cdot 1 \cdot \rangle (\mathtt{G} \neg crash \wedge \mathtt{G}(north \rightarrow \mathtt{F} south) \wedge \mathtt{G}(south \rightarrow \mathtt{F} north))$$

To solve the Maximum Single Step Cost Optimization problem, we reduce it to $\mathrm{ATL}^*_{\mathsf{sc}}$ model-checking. First we recall the construction of the accessory game given in [Alechina *et al.*, 2022]. The intuition behind this construction is that, given a game $\mathcal{G}$, an extra "normative agent" 0 is introduced who plays a role of a norm (its actions correspond to sets of actions of other agents made illegal by the norm). The resulting game is denoted $\mathcal{G}'$.

Construction 1 (Accessory game)**.** *For a given game $\mathcal{G}$, the accessory game* $\mathcal{G}' = \langle \mathrm{Ag}', \mathrm{Ac}', \mathrm{AP}', \mathsf{Cap}', \mathrm{Nrm}_1, \mathsf{q}_0^1, \mathsf{tr}', \mathsf{illegal}'_1, \eta'_1 \rangle$ *[Alechina* et al.*, 2022] is defined as:*

- $\mathrm{Ag}' = \{0\} \cup \mathrm{Ag}$ *includes a 0-agent, sometimes called the* normative agent*;*
- $\mathrm{Ac}' = \mathrm{Ac} \cup (2^{\mathrm{Ac} \times \mathrm{Ag}})$ *includes all possible sets of pairs of actions and agents as possible actions;*
- $\mathrm{AP}' = \mathrm{AP} \cup (\mathrm{Ac} \times \mathrm{Ag})$ *includes the set of pairs of actions and agents in the atomic propositions;*
- $\mathsf{Cap}'(j, \pi') = \begin{cases} 2^{\mathrm{Ac} \times \mathrm{Ag}}, & \text{if } j = 0 \\ \mathsf{Cap}(j, \pi'_{\restriction \mathrm{AP}}) \setminus (\{j\} \cap \pi'), & \text{o/w} \end{cases}$
- $\mathsf{tr}'(\pi', \vec{a}) = \mathsf{tr}(\pi'_{\restriction \mathrm{AP}}, \vec{a}_{-0}) \cup \vec{a}_0$*;*
- $\mathsf{illegal}'(\mathsf{q}_1, \pi') = \mathsf{illegal}(\mathsf{q}_1, \pi'_{\restriction \mathrm{AP}})$*;*
- $\eta'(\mathsf{q}_1, \pi') = \eta(\mathsf{q}_1, \pi'_{\restriction \mathrm{AP}})$.

Formally, the accessory game connects norms in $\mathcal{G}$ and strategies for the normative agent 0 in $\mathcal{G}'$ as follows.

Lemma 1. *[Alechina* et al.*, 2022, Lemma 1] For a given 1-MAS $\mathcal{G}$, its accessory game $\mathcal{G}'$, and a state $\pi \in 2^{\mathrm{AP}}$, the following two statements hold:*

- *For every strategy σ_0 of agent 0 in $\mathcal{G}'$, it holds that* $\mathsf{out}_{\mathcal{G}'}(\sigma_0, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}} = \mathsf{Paths}_{\mathcal{G} \oplus \mathcal{N}_{\sigma_0}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}}$*;*
- *For every norm $\mathcal{N}$ on $\mathcal{G}$, it holds that* $\mathsf{Paths}_{\mathcal{G} \oplus \mathcal{N}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}} = \mathsf{out}_{\mathcal{G}'}(\sigma_{\mathcal{N}}, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}}$.

The lemma turns the problem of synthesizing a norm for $\mathcal{G}$ to satisfy an objective φ into checking for an existence of a strategy for agent 0. The latter can be done by model-checking $\mathcal{G}'$ against an $\mathrm{ATL}^*_{\mathsf{sc}}$ formula $\langle \cdot 0 \cdot \rangle \varphi$. Checking for the existence of a strategy can be done in $(h + 1)$-EXPTIME for $\mathrm{ATL}^*_{\mathsf{sc}}$ formulas φ starting with an existential quantifier and $(h + 2)$-EXPTIME for $\mathrm{ATL}^*_{\mathsf{sc}}$ formulas starting with a universal quantifier (because $\langle \cdot 0 \cdot \rangle$ adds an extra alternation for those formulas).

Using a variant of this construction, we can solve the maximum single step norm optimization problem as follows.

Theorem 2. *The Maximum Single Step Norm Optimisation problem can be solved in time $(h + 1)$-EXPTIME for an existential $\mathrm{ATL}^*_{\mathsf{sc}}$ formula with quantifier alternation number h, and in $(h + 2)$-EXPTIME for a universal $\mathrm{ATL}^*_{\mathsf{sc}}$ formula with quantifier alternation number h.*

Proof. Consider the accessory game $\mathcal{G}'$ defined above and the variant $\mathcal{G}'_m$ that differs from $\mathcal{G}'$ on the capacity function, which is defined for the normative agent 0 as follows:

$$\mathsf{Cap}'_m(0, \pi') = \{X \in 2^{\mathrm{Ac} \times \mathrm{Ag}} : cost(X) \leq m\}$$

Clearly, the normative agent in $\mathcal{G}'_m$ is allowed to prevent actions only up to cost m at every single iteration.

Therefore, every solution to the normative synthesis in $\mathcal{G}'_m$ corresponds to a solution of the single step norm optimization in $\mathcal{G}$, and vice versa. □

Another interesting problem related to maximum single step norm optimization consists in finding the *optimal* cost, that is, the minimum value m^* for which it is possible to solve the maximum single step norm optimization problem.

Definition 5 (Function Form of Maximum Single Step Optimization)**.** *For a given k-MAS $\mathcal{G}$ and an $\mathrm{ATL}^*_{\mathsf{sc}}$ formula φ, find the minimum integer m^* for which the maximum single step cost optimization problem admits a solution.*

If the cost function $cost : 2^{\mathrm{Ac}\times\mathrm{Ag}} \longrightarrow \mathbb{N}$ is monotone, i.e., $A \subseteq B$ implies that $cost(A) \leq cost(B)$, by exploiting the solution provided in Theorem 2, we can compute m^* and the following theorem holds.

Theorem 3 (Optimal Cost Computation)**.** *For a given k-MAS $\mathcal{G}$ and an ATL$^*_{\mathsf{sc}}$ formula φ of quantifier alternation h, computing the optimal cost of the single step norm optimization problem can be solved in time $\log_2(M)\cdot(h+1)$-EXPTIME if φ is an existential formula and $\log_2(M)\cdot(h+2)$-EXPTIME if φ is a universal formula, where $M = cost(2^{\mathrm{Ac}\times\mathrm{Ag}})$.*

Proof. If the cost function is monotone, the maximum cost in $\mathcal{G}$ is given when the norm makes all actions for each agent illegal (except *noop*), which is computed as $M = cost(2^{\mathrm{Ac}\times\mathrm{Ag}})$.

We can therefore employ binary search in the range $[0, M]$ where, at each recursive step, with current interval $[a, b]$, we check for the solution of a single step norm optimization with $m = \frac{a+b}{2}$. If such a solution exists, we call the binary search over the range $[a, m]$, otherwise, we search over the range $[m + 1, b]$, and return the value m^* obtained at the end of the search procedure.

Clearly, the value m^* is optimal. Regarding the complexity analysis, notice that at each step of the binary search we run a maximum single step norm optimization procedure, which is of complexity $(h + 1)$-EXPTIME for existential formulas and $(h+2)$-EXPTIME for universal formulas. As the number of iterations is at most $\log_2(M)$, the overall complexity is as stated in the theorem. ☐

For non-monotone cost functions, m^* can be computed by first enumerating costs for all $X \subseteq 2^{\mathrm{Ac}\times\mathrm{Ag}}$ and then solving a single step norm optimization for each of those costs in increasing order.

5 Average Step Cost Optimisation

The maximal single step cost focuses on the greatest restriction imposed by a norm. In many applications, the average degree of restriction is also of interest. In this section, we consider the synthesis of norms that minimize the *average cost* of the prohibited set of actions over an infinite run.

Example 7. *Consider a slightly different scenario to Example 1 where simultaneously executing go only leads to a crash in the initial state of the environment. A norm that makes $(go, 2)$ illegal in the initial state and returns an empty illegal set subsequently ensures the objective in this scenario. Considering the maximal single step cost, this norm is no less prohibitive than the norm from Example 1. However, in the long run (apart from the initial state) the cost of the second norm tends to 0.*

We can model such a notion of cost using *mean-payoff games*. For a sequence $r \in \mathbb{R}^\omega$, the mean-payoff value of r, denoted $\mathsf{mp}(r)$ is defined as

$$\mathsf{mp}(r) = \liminf_{n\to\infty} \tfrac{1}{n}\textstyle\sum_{j=0}^{n-1} r_j.$$

We can use the mean-payoff value to define the *average step cost* of a norm. For a given game $\mathcal{G}$ a norm $\mathcal{N}_{k+1}$, a legal run $r = (\pi_0, \vec{q}_0), (\pi_1, \vec{q}_1), \ldots$ in $\mathcal{G}\oplus\mathcal{N}_{k+1}$ defines a sequence of costs for $\mathcal{N}_{k+1}$ defined as

$$cost(r) = cost(\mathsf{illegal}_{k+1}(q_0^{k+1}, \pi_0)), cost(\mathsf{illegal}_{k+1}(q_1^{k+1}, \pi_1)), \ldots,$$

which is a sequence of real numbers on which we can consider the mean-payoff value $\mathsf{mp}(cost(r))$.

Consider a game $\mathcal{G}$, a norm $\mathcal{N}_{k+1}$, and an ATL$^*_{\mathsf{sc}}$ formula of the form $\langle\cdot A\cdot\rangle\psi$ with ψ being a purely temporal formula [1]. Let σ_A be a strategy profile for the set A of agents such that $r \models_{\sigma_A} \psi$ for each $r \in \mathsf{out}_{\mathcal{G}\oplus\mathcal{N}_{k+1}}(c_0, \sigma_A)$, with c_0 being the initial configuration of $\mathcal{G} \oplus \mathcal{N}_{k+1}$. Note that σ_A exists if, and only if, $\mathcal{G} \oplus \mathcal{N}_{k+1}, c_0 \models \langle\langle A\rangle\rangle\psi$. Denote by $Win(\mathcal{G} \oplus \mathcal{N}_{k+1}, \langle\cdot A\cdot\rangle\psi)$ the set of strategy profiles for A that are a solution (winning) to the model-checking of $\langle\cdot A\cdot\rangle\psi$ against $\mathcal{G} \oplus \mathcal{N}_{k+1}$. Assuming that the set of solution strategy profiles is nonempty, we can define the average step cost of $\mathcal{N}_{k+1}$ for $\mathcal{G}$ and $\langle\cdot A\cdot\rangle\psi$ as follows:

$$cost_{mp}(\mathcal{N}_{k+1}, \langle\cdot A\cdot\rangle\psi) = \inf_{\sigma_A\in Win(\mathcal{G}\oplus\mathcal{N}_{k+1})} \sup_{r\in\mathsf{out}_{\mathcal{G}\oplus\mathcal{N}_{k+1}}(\sigma_A, c_0)} \mathsf{mp}(r)$$

Intuitively, the average step cost of $\mathcal{N}_{k+1}$ on the formula $\langle\cdot A\cdot\rangle\psi$ depends both on the best behavior of agents in A and the worst behavior of the antagonist agents, i.e., $\mathrm{Ag} \setminus A$.

Definition 6 (Average Step Cost Optimization)**.** *The* Average Step Cost Optimization *problem amounts to finding a norm $\mathcal{N}_{k+1}$ such that*

$$cost_{mp}(\mathcal{N}_{k+1}, \langle\cdot A\cdot\rangle\psi) \leq cost_{mp}(\mathcal{N}'_{k+1}, \langle\cdot A\cdot\rangle\psi)$$

for every other norm $\mathcal{N}'_{k+1}$, if such a norm exists.

Theorem 4. *There exists a game $\mathcal{G}$ and an ATL$^*_{\mathsf{sc}}$ formula φ for which the Average Step Cost Optimization problem cannot be solved.*

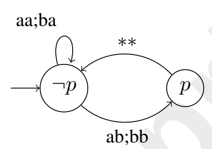

Figure 1: A game with unsolvable Average Step Cost Optimization.

Proof sketch. The proof proceeds by considering the game $\mathcal{G}$ depicted in Figure 1 and showing two things. The first is that, for every natural number n, there exists a norm $\mathcal{N}^n$ with exactly n internal states such that $\mathcal{G} \circ \mathcal{N}^n \models_\emptyset \langle\cdot 0\cdot\rangle \mathtt{GF}p$ with an average step cost of $\frac{1}{n}$. The second is that there not exist a finite norm $\mathcal{N}$ such that $\mathcal{G}\circ\mathcal{N} \models \langle\cdot 0\cdot\rangle \mathtt{GF}p$ whose average step cost is 0. Therefore, it shows an example where the average step cost cannot be optimized. ☐

Theorem **??** states that, in certain cases, we cannot solve the Average Step Cost Optimization problem *optimally*. By closely analyzing the counterexample in the proof, this is because we can always increase the memory of a given norm and so decrease its mean-payoff cost by, essentially, arbitrarily enlarging the loop on which the periodic costs are applied.

[1] Notice that this can also be seen as a ATL* formula

However, we can stop adding memory to the norm, as soon as the cost is within ε of the optimal value. We say that a norm $\mathcal{N}_{k+1}$ is ε-optimal if $cost_{mp}(\mathcal{N}_{k+1}, \langle\langle A\rangle\rangle\psi) \leq cost_{mp}(\mathcal{N}'_{k+1}, \langle\langle A\rangle\rangle\psi) + \varepsilon$, for every norm $\mathcal{N}'_{k+1}$ such that $\mathcal{G} \oplus \mathcal{N}'_{k+1} \models_{\emptyset} \langle\cdot A\cdot\rangle\psi$.

In the remainder of this section we show that for each ATL^*_{sc} formula of the form $\langle\cdot A\cdot\rangle\psi$ with ψ being a purely temporal formula, we can find an ε-optimal norm that solves the Average Step Cost Optimization problem. In order to do so, we first recall the notion of mean-payoff parity games, introduced in [Chatterjee *et al.*, 2005].

Definition 7 (Mean-Payoff Parity Games)**.** *A game graph $G = ((V, E), (V_0, V_1))$ consists of a directed graph (V, E) and a partition (V_0, V_1) of the set V of vertices. All game graphs have the property that every vertex has at least one out-going edge.*

A mean-payoff parity game $MP = (G, p, c)$ consists of a game graph G, a priority function $p : V \to [d]$ for some $d \in \mathbb{N}$ and a cost function $c : V \to \mathbb{R}$.

In [Chatterjee *et al.*, 2005], it is shown that an optimal strategy for solving a mean-payoff parity game may require infinite memory. However, for each $\varepsilon > 0$ there always exists a finite memory ε-optimal strategy for the maximizing agent [Chatterjee *et al.*, 2005, Lemma 3].

We can use this result to prove the following.

Theorem 5. *For every game $\mathcal{G}$, ATL^*_{sc} formula $\langle\cdot A\cdot\rangle\psi$ with ψ being a purely temporal formula, and $\varepsilon > 0$, there exists a ε-optimal norm $\mathcal{N}_{k+1}$ such that $\mathcal{G} \oplus \mathcal{N}_{k+1} \models_{\emptyset} \langle\cdot A\cdot\rangle\psi$.*

Proof sketch. The proof constructs the accessory game as defined in Construction 1 and considers the structural product with the Deterministic Parity Automaton $\mathcal{A}_\psi$ recognizing all and only the executions that satisfy the LTL formula ψ. This defines a mean-payoff parity game, whose optimal strategy for the maximizer corresponds to the combination of strategies for the normative agent 0 together with those for the agents in coalition A, that is, those that are existentially quantified. Therefore, as shown in [Chatterjee *et al.*, 2005, Lemma 3], we can always find a finite memory ε-optimal strategy for the maximizing agent, which then corresponds to a ε-optimal norm for $\mathcal{G}$ over $\langle\cdot A\cdot\rangle\psi$. □

6 Related Work

Although there has been a significant amount of work on synthesis with quantitative objectives, e.g. [Gutierrez *et al.*, 2017], there has been much less work on quantitative approaches to norm synthesis.

Previous work on the synthesis of dynamic norms has considered either less expressive objectives, or does not consider costs. [Huang *et al.*, 2016] considers the synthesis of dynamic norms to satisfy Computation Tree Logic (CTL) objectives, and [Perelli, 2019] considers the synthesis of dynamic norms for LTL objectives and Nash equilibria. [Perelli, 2019] addressed LTL objectives and a somewhat different notion of minimality of norms in energy games, where some resource ('energy') required for norm enforcement is periodically replenished. The work closest to our approach is [Alechina *et al.*, 2022] who consider the synthesis of dynamic norms for ATL* objectives. Given an ATL* objective and a concurrent game structure, a norm is synthesised so that the joint system (with some actions prohibited by the norm) satisfies the objective. However, [Alechina *et al.*, 2022] do not consider the cost of the synthesized norm, and use a less expressive logic as the specification language.

There also exists previous work on generating norms that are not overly restrictive or complex, e.g., [Morales *et al.*, 2015a; Christelis and Rovatsos, 2009]. For example, in [Morales *et al.*, 2015b], a synthesis procedure for state based norms that do not impose unneccessary restrictions on the agents (liberal norms) has been proposed. Issues related to costs have also been addressed in the literature. For example, [Alechina *et al.*, 2013] investigated the problem of whether a given desirable agent behaviour can still take place after a norm is imposed while incurring at most r sanctions (punishments for violating the norm). This can be seen as a cost of the norm. [Cao and Naumov, 2022] considered a logic where 'doing the right thing' (which can be interpreted as complying with a norm) was assigned a cost to the agent (e.g. diving in to save a drowning person clearly has a higher cost than throwing them a lifebuoy).

There exist logics of strategic ability where it is possible to express bounds on costs of agent strategies: RB-ATL [Nguyen *et al.*, 2018] and RB-ATL* [Alechina *et al.*, 2018]. However these logics cannot be used to express the bound on the cost of the normative agent's strategy, as, to date, no extensions of these logics with strategy contexts have been proposed. Moreover, RB-ATL and RB-ATL* consider the cost of achieving an ATL objective on all infinite runs of the system generated by a strategy. Hence without production of resources by the normative organisation they can express only reachability objectives, and invariant objectives that have a 0 cost on an infinite suffix. In contrast, we make no assumptions about the normative organisation's ability to produce resources, and only consider the cost of the most expensive step on a run and the average cost over an infinite run, which we believe are more appropriate in the context of norm synthesis.

The dynamic norms we consider are effectively implemented in a MAS through regimentation [Grossi *et al.*, 2006]. A *regimented* norm is impossible to violate due to the design of the MAS. For example, only authorised users can login to the system. An interesting direction for future work is considering the *enforcement* approach to implementing norms, which imposes a sanction on an agent when a norm is violated, e.g., a fine or social disapproval [Grossi *et al.*, 2006].

7 Conclusions

We presented a formal approach to synthesis of minimal cost dynamic norms to ATL^*_{sc} objectives. We showed that synthesis of minimum cost dynamic norms satisfying ATL^*_{sc} objectives is possible when considering the maximum single step cost, and does not result in increased complexity compared to synthesis of arbitrary norms. We also showed that for non-nested ATL^*_{sc} objectives, it is possible to synthesize ϵ-optimal norms w.r.t. to the average step cost. In future work, we plan to explore tradeoffs between the system-level objective and the cost of the norm.

Acknowledgments

Giuseppe Perelli received funding support from the PNRR MUR project PE0000013-FAIR, the PRIN 2020 projects PINPOINT and Sapienza University of Rome under the "*Progetti Grandi di Ateneo*" programme, grant RG123188B3F7414A (ASGARD - Autonomous and Self-Governing Agent-Based Rule Design).

References

[Ågotnes *et al.*, 2007] Thomas Ågotnes, Wiebe van der Hoek, Juan A. Rodríguez-Aguilar, Carles Sierra, and Michael J. Wooldridge. On the logic of normative systems. In Manuela M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1175–1180, 2007.

[Ågotnes *et al.*, 2010] Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge. Robust normative systems and a logic of norm compliance. *Logic Journal of the IGPL*, 18(1):4–30, 2010.

[Alechina *et al.*, 2013] Natasha Alechina, Mehdi Dastani, and Brian Logan. Reasoning about normative update. In *Proceedings of the Twenty Third International Joint Conference on Artificial Intelligence (IJCAI 2013)*, pages 20–26. AAAI Press, 2013.

[Alechina *et al.*, 2018] Natasha Alechina, Nils Bulling, Stéphane Demri, and Brian Logan. On the complexity of resource-bounded logics. *Theor. Comput. Sci.*, 750:69–100, 2018.

[Alechina *et al.*, 2022] Natasha Alechina, Giuseppe De Giacomo, Brian Logan, and Giuseppe Perelli. Automatic synthesis of dynamic norms for multi-agent systems. In Gabriele Kern-Isberner, Gerhard Lakemeyer, and Thomas Meyer, editors, *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning, KR 2022*, 2022.

[Alur *et al.*, 2002] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002.

[Astefanoaei *et al.*, 2009] L. Astefanoaei, M. Dastani, J.J. Meyer, and F. de Boer. On the semantics and verification of normative multi-agent systems. *International Journal of Universal Computer Science*, 15(13):2629–2652, 2009.

[Boella and van der Torre, 2004] Guido Boella and Leendert van der Torre. Regulative and constitutive norms in normative multiagent systems. In *Proceedings of the Ninth International Conference on Principles of Knowledge Representation and Reasoning (KR'04)*, pages 255–266, 2004.

[Boella *et al.*, 2008] G. Boella, J. Broersen, and L. van der Torre. Reasoning about constitutive norms, counts-as conditionals, institutions, deadlines and violations. In *Proceedings of the International Conference on Principles and Practice of Multi-Agent Systems (PRIMA)*, pages 86–97, 2008.

[Bulling and Dastani, 2016] Nils Bulling and Mehdi Dastani. Norm-based mechanism design. *Artif. Intell.*, 239:97–142, 2016.

[Bulling *et al.*, 2013] Nils Bulling, Mehdi Dastani, and Max Knobbout. Monitoring norm violations in multi-agent systems. In *Twelfth International conference on Autonomous Agents and Multi-Agent Systems (AAMAS'13)*, pages 491–498, 2013.

[Cao and Naumov, 2022] Rui Cao and Pavel Naumov. The limits of morality in strategic games. In Luc De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022*, pages 2561–2567. ijcai.org, 2022.

[Chatterjee *et al.*, 2005] Krishnendu Chatterjee, Thomas A. Henzinger, and Marcin Jurdzinski. Mean-payoff parity games. In *20th IEEE Symposium on Logic in Computer Science (LICS 2005)*, pages 178–187. IEEE Computer Society, 2005.

[Chopra *et al.*, 2018] Amit Chopra, Leendert van der Torre, Harko Verhagen, and Serena Villata, editors. *Handbook of Normative Multiagent Systems*. College Publications, 2018.

[Christelis and Rovatsos, 2009] George Christelis and Michael Rovatsos. Automated norm synthesis in an agent-based planning environment. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 161–168, 2009.

[Corapi *et al.*, 2011] Domenico Corapi, Alessandra Russo, Marina De Vos, Julian A. Padget, and Ken Satoh. Normative design using inductive learning. *Theory Pract. Log. Program.*, 11(4-5):783–799, 2011.

[Da Costa Lopes *et al.*, 2010] Arnaud Da Costa Lopes, François Laroussinie, and Nicolas Markey. ATL with strategy contexts: Expressiveness and model checking. In Kamal Lodaya and Meena Mahajan, editors, *FSTTCS 2010*, volume 8 of *LIPIcs*, pages 120–132. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010.

[Dastani *et al.*, 2013] Mehdi Dastani, Davide Grossi, and John-Jules Meyer. A logic for normative multi-agent programs. *Journal of Logic and Computation, special issue on Normative Multiagent Systems*, 23(2):335–354, 2013.

[Dennis *et al.*, 2010] Louise A. Dennis, Nick A. M. Tinnemeier, and John-Jules Ch. Meyer. Model checking normative agent organisations. In Jürgen Dix, Michael Fisher, and Peter Novák, editors, *Computational Logic in Multi-Agent Systems - 10th International Workshop, CLIMA X, Revised Selected and Invited Papers*, volume 6214 of *Lecture Notes in Computer Science*, pages 64–82. Springer, 2010.

[Dignum *et al.*, 2004] F. Dignum, J. Broersen, V. Dignum, and J.-J. C. Meyer. Meeting the deadline: Why, when and how. In *Proceedings of the International Workshop on Formal Approaches to Agent-Based Systems (FAABS)*, pages 30–40, 2004.

[Fitoussi and Tennenholtz, 2000] David Fitoussi and Moshe Tennenholtz. Choosing social laws for multi-agent systems: Minimality and simplicity. *Artificial Intelligence*, 119(1):61–101, 2000.

[Grossi *et al.*, 2006] Davide Grossi, Huib Aldewereld, and Frank Dignum. *Ubi Lex, Ibi Poena* : Designing norm enforcement in e-institutions. In Pablo Noriega, Javier Vázquez-Salceda, Guido Boella, Olivier Boissier, Virginia Dignum, Nicoletta Fornara, and Eric Matson, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems II - AAMAS 2006 and ECAI 2006 International Workshops, COIN 2006, Revised Selected Papers*, volume 4386 of *Lecture Notes in Computer Science*, pages 101–114. Springer, 2006.

[Gutierrez *et al.*, 2017] Julian Gutierrez, Aniello Murano, Giuseppe Perelli, Sasha Rubin, and Michael J. Wooldridge. Nash equilibria in concurrent games with lexicographic preferences. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017*, pages 1067–1073. ijcai.org, 2017.

[Huang *et al.*, 2016] X. Huang, J. Ruan, Q. Chen, and K. Su. Normative Multiagent Systems: The Dynamic Generalization. pages 1123–1129, 2016.

[Laroussinie and Markey, 2015] François Laroussinie and Nicolas Markey. Augmenting ATL with strategy contexts. *Inf. Comput.*, 245:98–123, 2015.

[Meyer and Wieringa, 1993] J.-J. Ch. Meyer and R. J. Wieringa. Deontic logic: A concise overview. In J.-J. Ch. Meyer and R.J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 3–16. John Wiley & Sons, 1993.

[Morales *et al.*, 2015a] Javier Morales, Maite López-Sánchez, Juan Antonio Rodríguez-Aguilar, Wamberto Weber Vasconcelos, and Michael J. Wooldridge. Online automated synthesis of compact normative systems. *ACM Trans. Auton. Adapt. Syst.*, 10(1):2:1–2:33, 2015.

[Morales *et al.*, 2015b] Javier Morales, Maite López-Sánchez, Juan Antonio Rodríguez-Aguilar, Michael J. Wooldridge, and Wamberto Weber Vasconcelos. Synthesising liberal normative systems. In Gerhard Weiss, Pinar Yolum, Rafael H. Bordini, and Edith Elkind, editors, *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015*, pages 433–441. ACM, 2015.

[Morales *et al.*, 2018] Javier Morales, Michael J. Wooldridge, Juan A. Rodríguez-Aguilar, and Maite López-Sánchez. Off-line synthesis of evolutionarily stable normative systems. *Auton. Agents Multi Agent Syst.*, 32(5):635–671, 2018.

[Nguyen *et al.*, 2018] Hoang Nga Nguyen, Natasha Alechina, Brian Logan, and Abdur Rakib. Alternating-time temporal logic with resource bounds. *J. Log. Comput.*, 28(4):631–663, 2018.

[Perelli, 2019] Giuseppe Perelli. Enforcing Equilibria in Multi-Agent Systems. pages 188–196, 2019.

[Pnueli and Rosner, 1989] Amir Pnueli and Roni Rosner. On the synthesis of a reactive module. In *Conference Record of the Sixteenth Annual ACM Symposium on Principles of Programming Languages, Austin, Texas, USA, January 11-13, 1989*, pages 179–190. ACM Press, 1989.

[Shoham and Tennenholtz, 1995] Yoav Shoham and Moshe Tennenholtz. On social laws for artificial agent societies: off-line design. *Artificial Intelligence*, 73(1-2):231–252, 1995.

[Tinnemeier *et al.*, 2009] Nick Tinnemeier, Mehdi Dastani, J-J. Ch. Meyer, and Leon van der Torre. Programming normative artifacts with declarative obligations and prohibitions. In *Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, WI-IAT'09*, volume 2, pages 145 –152, September 2009.

[van der Hoek *et al.*, 2007] Wiebe van der Hoek, Mark Roberts, and Michael J. Wooldridge. Social laws in alternating time: effectiveness, feasibility, and synthesis. *Synthese*, 156(1):1–19, 2007.