Imperfect norm enforcement in stochastic environments: an analysis of efficiency and cost tradeoffs

Moser Silva Fagundes[1], Sascha Ossowski[2], and Felipe Meneguzzi[3]

[1] Federal Institute of Education, Science and Technology
Sul-Rio-Grandense (IFSul), Charqueadas, RS, Brazil
moserfagundes@charqueadas.ifsul.edu.br
[2] Centre for Intelligent Information Technologies (CETINIA),
University Rey Juan Carlos (URJC), Móstoles, Madrid, Spain
sascha.ossowski@urjc.es
[3] School of Informatics, Pontifical Catholic University
of Rio Grande do Sul (PUCRS), Porto Alegre, Brazil
felipe.meneguzzi@pucrs.br

Abstract. In heterogeneous multiagent systems, agents might interfere with each other either intentionally or unintentionally, as a side-effect of their activities. One approach to coordinating these agents is to restrict their activities by means of social norms whose compliance ensures certain system properties, or otherwise results in sanctions to violating agents. While most research on normative systems assumes a deterministic environment and norm enforcement mechanism, we formalize a normative system within an environment whereby agent actions have stochastic outcomes and norm enforcement follows a stochastic model in which stricter enforcement entails higher cost. Within this type of system, we analyze the tradeoff between norm enforcement efficiency (measured in number of norm violations) and its cost considering a population of norm-aware self-interested agents capable of building plans to maximize their expected utilities. Finally, we validate our analysis empirically through simulations in a representative scenario.

Keywords: NMDP, MDP, Stochastic, Norm, Enforcement.

1 INTRODUCTION

Autonomous selfish agents in heterogenous societies must act in order to accomplish their individual objectives while taking into account potential disruption caused by other members of the society. The key challenge in designing these open societies is in ensuring that the agents are able to achieve their own goals while minimizing the impact of negative interference between their actions. When dealing with a small number of agents this problem is typically modeled using game theory, with algorithms designed to find strategies in some kind of equilibrium. However, finding a Nash equilibrium [11] with a lower bound on the expected payoff within a stochastic multiplayer game is undecidable in general, and typically contained in the PSPACE complexity class even with simplifications on the types of acceptable strategy in the equilibrium [16].

An alternative approach to achieve desirable properties within a multiagent society is to define societal norms [9] that regulate agent behavior. Norms use deontic concepts of obligations and prohibitions in the specification of soft-constraints on agent behavior, with failure to comply with the norms resulting in sanctions designed to restore a society to a desirable state. To ensure that a norm-regulated society operates as expected, it must provide enforcement mechanisms that monitor agent behavior and apply sanctions when transgressions are detected. As stated in [10], most existing work on norm-regulated agent societies assume a deterministic environment and disregard the norm enforcement cost, which limits the applicability of these techniques.

In this paper, we define the dynamics of the world and the norm enforcement mechanism in Section 2 without the assumption of determinism, and we formalize the decision making process of our agents with *Normative Markov Decision Processes* (NMDPs) [6, 7, 5] in Section 3. We consider a stochastic norm-based coordination mechanism aiming at ensuring that the multiagent system as a whole runs according to the properties specified in a set of social norms. Thus, agents are subjected to social norms, and failure to comply with these norms brings about some kind of sanction. Norms are enforced on the basis of the observation of the current world-state by a mechanism that detects violations with a certain probability. Within this model, we associate a cost such that stricter enforcement mechanisms incur a higher cost to the enforcement authority. Such an enforcement costing model mirrors the real world, where mechanisms with a higher probability of detecting (and sanctioning) violations are more expensive. We develop a simulation-based method to calculate this tradeoff (norm enforcement efficiency measured in number of norm violations and its cost), and show results for a representative scenario in Section 4. Finally, we conclude the paper pointing towards future developments in Section 5. In summary, this paper makes two major contributions. First, we define a rich stochastic norm enforcement mechanism in stochastic environments populated with self-interested rational agents, and second, we provide insights into the tradeoffs involved in norm enforcement.

2 NORMATIVE STOCHASTIC ENVIRONMENT

2.1 World model and norms

Let $\mathcal{G} = \{a_1, \ldots, a_n\}$ be the set of agents operating in the multiagent system. The state space of an agent $a_i \in \mathcal{G}$ is defined as follows. Let $\mathcal{F}$ be the set of features that characterize different aspects of the states of the world. A feature $f_j \in \mathcal{F}$ can take on a finite number of values and $\mathcal{V}_{f_j}$ corresponds to the finite set of possible values of f_j. The state of an agent is a complete assignment of values to its features, and the state space $\mathcal{S}$ is the cross product of the value spaces for the features, i.e.: $\mathcal{S} = \times_{i=1}^{|\mathcal{F}|} \mathcal{V}_{f_j}$. The *current state* of the agents determines the system's current state and it is represented as a vector $\{s_1, \ldots, s_n\}$ in which s_i is the current state of the agent $a_i \in \mathcal{G}$. In our environment, the outcome of the actions is *stochastic*, that is, the intended resulting state of executing an action occurs with a given probability. For each state-transition, an agent receives an *immediate reward* and the sum of rewards received by this agent determines its *utility*. The current utility of the agents in $\mathcal{G}$ is represented as a vector $\mathcal{U} = \{u_1, \ldots, u_n\}$ in which u_i is the utility of $a_i \in \mathcal{G}$.

Although the various efforts to develop normative multiagent systems proposed in the literature differ on technical details, they all share the intuition that norms are constraints on the behaviour of agents, by means of which some global goals can be achieved [1]. Our approach is related to strands of research on governing environments [14] and coordination infrastructures [12], in that we assume the existence of coherent set of norms, allowing the agents to be fully compliant with the norms if they decide to do so. We assume that the norms are expressed in a sufficiently expressive language, but with associated mechanisms which are tractable, regulating the behavior of autonomous agents in an otherwise open multiagent system. In what follows, we formalize the key notions that characterize our model of norms.

Definition 1 (Norm). *A norm is a tuple* $\langle \delta, \mathcal{X}, \mathcal{E}, \sigma \rangle$ *where:* $\delta \in \{\mathsf{OBLIGATION}, \mathsf{PROHIBITION}\}$ *is the deontic modality;* $\mathcal{X} \subseteq \mathcal{S}$ *is the set of states (normative context) in which the norm applies;* $\mathcal{E} \subseteq \mathcal{X}$ *is the subset of states in the normative context which are obliged or prohibited (target states); and* σ *is a sanction represented by a tuple* $\langle \rho, \phi \rangle$*:*

- $\rho : \mathcal{S} \mapsto \mathbb{R}$ *is a function that gives the penalty for violating this norm in a given state* ($\rho(s)$ *yields the penalty to be paid in* s*);*
- $\phi : \mathcal{S} \mapsto \mathcal{S}$ *is a function that calculates the state resulting from an enforced state-transition in response to the violation of this norm* ($\phi(s)$ *yields the outcome of an enforced state-transition in* s).

Broadly speaking, the intended semantics of a norm is as follows: if the norm is a *prohibition*, in the set of states $\mathcal{X}$ where the norm applies, the agents are prohibited to be in any state in $\mathcal{E}$; if it is an *obligation*, in the set of states $\mathcal{X}$ where the norm applies, the agents are obliged to be in some state in $\mathcal{E}$. A *sanction* consists of a penalty and an enforced state-transition aimed at updating the current state of agents that have transgressed a norm. The underlying intent of the sanctions is to punish the transgressors by decreasing their utility, and moving them from violating states to states where the norms are obeyed and/or their capabilities are limited. A set of states is relevant to a norm $\langle \delta, \mathcal{X}, \mathcal{E}, \sigma \rangle$ if this set of states is a subset of $\mathcal{X}$, which indicates the context where the norm applies. Given a set of states that are relevant to a norm, we can determine which of them violate it. In Definition 2 we formalize the norm violating states.

Definition 2 (Violating states). *Let* q *be a norm* $\langle \delta, \mathcal{X}, \mathcal{E}, \sigma \rangle$*, the set of states that violate* q*, denoted as* $\mathcal{S}_q^{\triangledown}$*, is defined as:*

$$\mathcal{S}_q^{\triangledown} = \begin{cases} \mathcal{E} & if \;\; (\delta = \mathsf{PROHIBITION}) \\ \mathcal{X} \setminus \mathcal{E} & if \;\; (\delta = \mathsf{OBLIGATION}). \end{cases}$$

Previous research [3] has shown that *factored* representations can be used to design efficient MDP solution algorithms that exploit *structures* in the state space. Using this factored representation, the norm in Definition 1 can be expressed in a compact way as in [7] (an example is given in Section 4.1). In the systems that we envision, norms are assumed to be common knowledge, represented as the following set.

Definition 3 (Set of norms). *A set of norms* $\mathcal{N}$ *is a totally ordered set* $\{q_1, q_2, \ldots q_m\}$ *where each norm* $q_k \in \mathcal{N}$ *is defined according to Definition 1.*

2.2 Norm enforcement mechanism

The model of norm enforcement mechanism developed in this paper is based on the detection of violating states in terms of observations of the agents' current state in the world. Such observations are assumed to be imperfect[4], so that the mechanism only detects violations with a certain probability, as stated in Definition 4.

Definition 4 (Detection model). *Let $\mathcal{N}$ be the set of norms and $\mathcal{S}$ be the state space. A probabilistic detection model consists of a function $\mathcal{D} : \mathcal{N} \times \mathcal{S} \rightarrow [0,1]$ such that $\mathcal{D}(q, s)$ returns the detection probability of the violation of the norm $q \in \mathcal{N}$ in the state $s \in \mathcal{S}$.*

Besides being imperfect, the enforcement mechanism is *resource-bounded* so that monitoring the environment has an associated cost. This cost is a function of the accuracy of observations and the size of the population of agents. We formalize this function as $\mathcal{MK}(\mathcal{D}, \mu)$, which returns the cost per time step of detecting norm violations according to the model $\mathcal{D}$ in a multiagent system with population size μ. Note that at this point we do not provide a specific formulation for this function since the degree in which parameters affect the costs depends on the environment to be created or simulated.

If a norm violation is detected by the norm enforcement mechanism, the respective sanction is applied immediately. According to Definition 5, which formalizes the application of a single sanction, the mechanism has the power to change the current state s_i (enforced state-transition) and the utility u_i (penalization) of an agent a_i. Multiple sanctions can be applied where multiple norms have been violated – one sanction per norm violation that has been detected. This is formalized in Definition 6.

Definition 5 (Application of sanction). *Let $q \in \mathcal{N}$ be a norm violated by a given agent in a state $s \in \mathcal{S}$, and let $\sigma = \langle \rho, \phi \rangle$ be the sanction of q to be applied in this state. Then the application of σ in the state s results in a state-transition of the agent to the state $\phi(s)$ and a penalty of $\rho(s)$ utility units.*

Definition 6 (Application of sanctions). *Let $\{q_1, q_2, \ldots q_m\} \subseteq \mathcal{N}$ be an ordered set of norms which violation has been detected in a given state $s \in \mathcal{S}$. Let $\{\sigma_1, \sigma_2, \ldots \sigma_m\}$ be the sanctions to be applied in this state, such that $\sigma_k = \langle \rho_k, \phi_k \rangle$ corresponds to the sanction of the norm q_k. The application of $\{\sigma_1, \sigma_2, \ldots \sigma_m\}$ in s results in a transition to $\phi_1(\phi_2(\ldots(\phi_m(s))))$ and a penalization of $\rho_1(s) + \rho_2(s) + \ldots + \rho_m(s)$ utility units.*

Our approach consists of merging multiple state-transitions into one in which the initial state is the violating state, and the outcome state is calculated by applying the function ϕ of the sanctions sequentially. Similarly to detecting norm violations, the application of sanctions implies costs for the norm enforcement mechanism. Thus, the sanctioning cost $\mathcal{SK}(t)$ for a given time step t, is:

$$\mathcal{SK}(t) = \sum_{k=1}^{|\mathcal{N}|} \mathcal{I}(\sigma_k, t)\ \mathcal{SSK}(\sigma_k)$$

[4] Given the number of state features of some complex domains (large state spaces), it can be computationally costly for the norm enforcer to keep track of all potentially relevant features. It happens in real life situations as well. For example, when a police district has to distribute a limited number of officers in a region. Some norm violations may not be detected given that the police cannot cover all public spaces of the region.

where the function $\mathcal{I}(\sigma_k, t)$ gives the number of times that the sanction σ_k has been applied in the time step t, and the function $\mathcal{SSK}(\sigma_k)$ returns the cost of applying the sanction σ_k. In summary, the total norm enforcement cost at a given time step t, denoted as $\mathcal{K}(t)$, is determined as follows:

$$\mathcal{K}(t) = \mathcal{MK}(\mathcal{D}, \mu) + \mathcal{SK}(t) \tag{1}$$

where $\mathcal{MK}(\mathcal{D}, \mu)$ is detection cost per time step employing the detection model $\mathcal{D}$ in a population μ, and $\mathcal{SK}(t)$ is the sanctioning cost in the time step t.

3 NORMATIVE AGENT REASONING

We use *Normative Markov Decision Processes* (NMDPs) to generate the policies followed by our selfish agents operating under the norm enforcement mechanism. An NMDP is a formal model that integrates the normative framework detailed in Section 2 with the widely known MDPs [2]. The result of such an integration is a framework to develop rational agents that comply with a norm only if the expected utility of doing so exceeds the expected utility of violating this norm (for details about NMDPs and a comparison with related work, including normative multi-agent systems and electronic institutions, see Fagundes [5]).

Definition 7 (Normative Markov Decision Process). *An* NMDP *is a tuple* $\langle \mathcal{S}, \mathcal{A}, \mathcal{C}, \mathcal{T}, \mathcal{R}, \mathcal{N}, \mathcal{D} \rangle$ *where:* $\mathcal{S}$ *is the finite set of states of the world;* $\mathcal{A}$ *is a finite set of actions;* $\mathcal{C} : \mathcal{S} \to \mathcal{A}$ *is a capability function that denotes the set of admissible actions in a given state (*$\mathcal{C}(s)$ *corresponds to the set of admissible actions in the state* s*);* $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ *is a state-transition function (*$\mathcal{T}(s, a, s')$ *indicates the probability of executing* a *at* s *and ending at* s'*);* $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ *is a reward function that determines the reward (*$\mathcal{R}(s, a, s')$ *corresponds to the one gained by the agent for executing* a *at* s *and ending at the state* s'*),* $\mathcal{N}$ *is a set of norms specified according to Definition 3; and* $\mathcal{D} : \mathcal{N} \times \mathcal{S} \to \mathbb{R}$ *is a detection function specified according to Definition 4.*

In order to generate a behavior policy for an NDMP, agents use the control flow of Figure 1. In this model, agents perform *normative reasoning* to identify which states violate which norms and represent the respective sanctions within the violating states. The outcome of this process is a standard MDP, which can be solved by well known algorithms [2, 8, 13]. Finally, the agent executes the resulting policy π.

Hardwiring normative constraints into agent behavior [15] can be problematic since it constrains the agent's ability to adapt to changes in the system's norms [4]. In consequence, we develop a technique to generate an MDP at runtime that incorporates sanctions from an NMDP. This technique consists of identifying, for all states, which norms they violate (if any), following Definition 2. Then, for each violating state $s \in \mathcal{S}_q^{\triangledown}$, the sanction of the norm q is represented with s by modifying the transition probabilities $\mathcal{T}$ and rewards $\mathcal{R}$ based on the norms $\mathcal{N}$ and detection probabilities $\mathcal{D}$. The result of the *normative reasoning*, as shown in Figure 1, is a standard MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{C}, \mathcal{T}', \mathcal{R}' \rangle$ in which $\mathcal{T}'$ is the new state-transition function and $\mathcal{R}'$ is the new reward function. In what follows, we describe how $\mathcal{T}'$ and $\mathcal{R}'$ are computed.

Let $\mathcal{Q}_s$ be the set of norms violated in a given state $s \in \mathcal{S}$. The combinations of sanctions in s are provided by the power set of $\mathcal{Q}_s$ denoted as $\mathsf{P}(\mathcal{Q}_s)$. For instance, if $\mathcal{Q}_s = \{q_1, q_2\}$ then $\mathsf{P}(\mathcal{Q}_s) = \{\emptyset, \{q_1\}, \{q_2\}, \{q_1, q_2\}\}$. The outcome resulting from a combination of sanctions $\mathcal{B} \in \mathsf{P}(\mathcal{Q}_s)$ in s is computed by the function below[5]:

$$\mathsf{Out}(s, \mathcal{B}) = \begin{cases} \mathsf{Head}(\mathcal{B}).\sigma.\phi(\mathsf{Out}(s, \mathcal{B} \setminus \mathsf{Head}(\mathcal{B}))) & if \ \ |\mathcal{B}| > 1 \\ \mathsf{Head}(\mathcal{B}).\sigma.\phi(s) & if \ \ |\mathcal{B}| = 1 \end{cases}$$

where $\mathsf{Head}(\mathcal{B})$ is a function that returns the first element in $\mathcal{B}$ and $\mathsf{Head}(\mathcal{B}).\sigma.\phi$ refers to the function ϕ of the sanction σ of the norm $\mathsf{Head}(\mathcal{B})$. Note that function $\mathsf{Out}(s, \mathcal{B})$ computes the outcome state resulting from a combination of sanctions $\mathcal{B}$ according to Definition 6, ensuring that norm enforcement mechanism and agents calculate the same outcome state for any combination of sanctions.

In order to represent the effect of sanctions in the transition and reward functions of the new MDP, we determine the set of all combinations of sanctioned norm violations $\mathsf{W}(s, s') \subseteq \mathsf{P}(\mathcal{Q}_s)$ which, if executed in $s \in \mathcal{S}$, bring about $s' \in \mathcal{S}$:

$$\mathsf{W}(s, s') = \{\mathcal{B} \in \mathsf{P}(\mathcal{Q}_s) \mid (\mathcal{B} \neq \emptyset) \wedge \mathsf{Out}(s, \mathcal{B}) = s'\}.$$

Using the detection probabilities from function $\mathcal{D}$, we compute the probability that a combination $\mathcal{B} \in \mathsf{P}(\mathcal{Q}_s)$ occurs in $s \in \mathcal{S}$ as follows:

$$\mathsf{Pro}(\mathcal{B}, s) = \prod_{q \in \mathcal{B}} \mathcal{D}(q, s) \prod_{q \in \mathcal{Q}_s \setminus \mathcal{B}} (1 - \mathcal{D}(q, s)).$$

Bringing it all together, Formula (2) calculates $\mathcal{T}'(s, a, s')$: the probability of executing action a at s and ending up at s' taking into account the sanctions to be applied in the initial state s. First, take the probability of a transition if no sanction takes place. This is determined by multiplying the transition probability $\mathcal{T}(s, a, s')$ by the probability $\mathsf{Pro}(\emptyset, s)$ that no sanctioning happens. Then, we take the probability of occurrence for each combination of sanctions $\mathcal{B} \in \mathsf{W}(s, s')$, which if executed in s ends in s'.

$$\mathcal{T}'(s, a, s') = \mathcal{T}(s, a, s')\,\mathsf{Pro}(\emptyset, s) + \sum_{\mathcal{B} \in \mathsf{W}(s, s')} \mathsf{Pro}(\mathcal{B}, s) \qquad (2)$$

[5] In an abuse of notation, we refer to sub-components of tuples using an object-oriented programming idiom, so that $\sigma.\phi$ refers to the component ϕ of tuple σ.

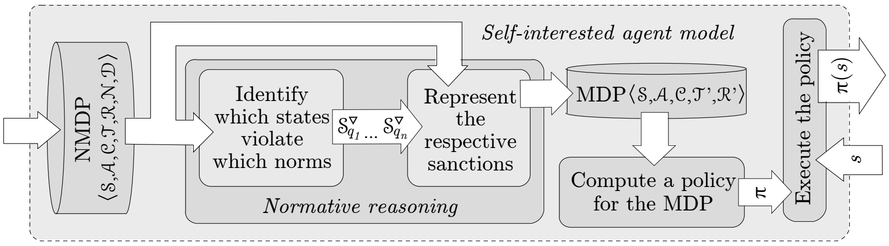
Self-interested agent model
NMDP
⟨𝒮,𝒜,𝒞,𝒯,ℛ,𝒩,𝒟⟩
Identify which states violate which norms
$\mathcal{S}^{\triangledown}_{q_1} \ldots \mathcal{S}^{\triangledown}_{q_n}$
Represent the respective sanctions
Normative reasoning
MDP⟨𝒮,𝒜,𝒞,𝒯',ℛ'⟩
Compute a policy for the MDP
π
Execute the policy
π(s)
s

Fig. 1. Control flow of the self-interested agent model.

Finally, Formula (3) describes how to calculate $\mathcal{R}'(s,a,s')$, the immediate reward of executing action a at the origin state s and ending at s' taking into account the sanctions to be applied at s. To calculate $\mathcal{R}'(s,a,s')$ we use $\mathcal{R}(s,a,s')$, the immediate reward for this state-transition if no sanction is applied, and $\mathsf{Pen}(\mathcal{B},s)$, the penalty of a combination of sanctions $\mathcal{B} \in \mathsf{W}(s,s')$ at s. As these combinations of sanctions are mutually exclusive and happen with different probabilities, we calculate $\mathcal{R}'(s,a,s')$ as a weighted average reward on which the probabilities are the weights, where the penalty of $\mathcal{B} \in \mathsf{P}(\mathcal{Q}_s)$ is : $\mathsf{Pen}(\mathcal{B},s) = \sum_{q \in \mathcal{B}} q.\sigma.\rho(s)$.

$$\mathcal{R}'(s,a,s') = \frac{\mathsf{Pro}(\emptyset,s)\,\mathcal{T}(s,a,s')\,\mathcal{R}(s,a,s') + \sum_{\mathcal{B}\in\mathsf{W}(s,s')} \mathsf{Pro}(\mathcal{B},s)\mathsf{Pen}(\mathcal{B},s)}{\mathcal{T}'(s,a,s')} \tag{3}$$

4 EXPERIMENTS

We now turn our attention to evaluating stochastic norm-enforcement mechanisms. The goal of our evaluation is to determine effective norm enforcement intensities that balance the ability of the mechanism to self-support itself and at the same time to ensure that coordination problems do not exceed a maximum acceptable level.

4.1 Motion environment

The motion environment, shown in Figure 2, is a stochastic environment as defined in Section 2.1, made up of discrete cells, where the agents (depicted as triangles indicating their moving direction) are able to move one cell at a time. There are 8 lanes and each lane contains 24 contiguous cells. There are also 8 gateways through which the agents enter and leave the environment. The position of each agent is determined by features LANE and CELL, its orientation is defined by DIRECTION, and STATUS indicates whether it is holding a position or moving through the cells.

Each norm-aware agent $a_i \in \mathcal{G}$ operating in the motion environment is modeled as an NMDP that computes its policy following the reasoning process described in Section 3. The agents' current state (c.f. Section 3) determines the system's current state and it is represented as a vector $\{s_1,\ldots,s_n\}$ in which s_i, the current state of the agent $a_i \in \mathcal{G}$, is a complete assignment of values to its features.

Once released in a gateway, an agent can choose between two adjoining lanes. Nondeterminism is modeled by the fact the move action is "unreliable": its intended outcome occurs with probability 0.99, but with probability 0.01 the agent remains in the same position. With this we intend to model the fact that the "engine" of our agent can fail (i.e. out of gas, mechanical problems). The agents' actions are executed synchronously within the motion environment, that is, each agent executes one action per time-step. For every state-transition implied by an action, the agent receives a reward, which may be positive or negative. For this environment, the reward is $-0.01\mathsf{R_N}$ for all transitions ($\mathsf{R_N}$ is the reward unit), except when the agent reaches the assigned destination gateway, which results in a reward of $+0.4\mathsf{R_N}$ (an incentive to reach this gateway quickly). Note that leaving the environment via any other gateway gives $-0.01\mathsf{R_N}$.

These state-transition probabilities and rewards have been arbitrarily chosen. In this section, we focus on the experimental analysis in simulated environments, and our technique can be applied with different probabilities and reward levels.

In our experiments, the agents know the norms and the detection probabilities. That is, although they do not know whether individual norm violations will be detected, they are aware of the probability of violations being detected in general. The agents' perception is incomplete, so while traveling across the environment, the agents always know their own position, but not the position of other agents. This assumption about the agents' perception may cause coordination problems: two agents that come from *opposite directions* crash into each other if they try to occupy the same cell or try to cross each other (two examples of imminent crashes are highlighted in Figure 2 with circles). When agents crash, they are removed from the environment. To cope with these coordination problems, we introduce a set of norms to regulate the traveling direction in each lane (see the arrows in Figure 2). If a violation is detected, the agent is driven to the obliged direction and loses $0.1\mathsf{R_N}$. These norms are specified using the following template, where X is a particular lane and Y is the obliged moving direction:

$$\langle \text{OBLIGATION}, (\text{LANE=X}), (\text{DIRECTION=Y}), \\ \langle \{ \top \rightarrow -1.0\mathsf{R_N} \},\{ \top \rightarrow \{(\text{DIRECTION=Y})\} \} \rangle\rangle$$

The main purpose of this environment is to serve as a testbed for studying the tradeoff between norm enforcement efficiency and its cost, taking into account a population of norm-aware self-interested agents capable of complex planning to maximize their expected utilities. Origin and destination gateways of each agent were randomly assigned using a uniform distribution. Each simulation ran for 10^6 time steps, resulting in a 95% confidence interval for the results shown in this section, allowing us to identify the parameters and value ranges in which the societal behavior changes significantly:

- β is the detection probability of violations. In our simulations, $\mathcal{D}(q, s) = \beta$ for all norms and all states, and β ranges from 0.01 to 0.13 in intervals of 0.01.
- μ is the population setting, corresponding to the number of agents in the environment at any given time throughout the entire simulation. In our simulations $\mu = 10$.

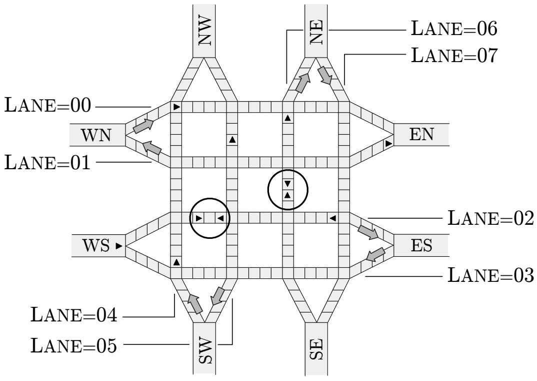

Fig. 2. Motion environment. Circles indicate imminent crashes.

The norm enforcement cost is computed using Formula 1 where $\mathcal{MK}(\mathcal{D}, \mu)$ is the cost of monitoring violations, and $\mathcal{SK}(t)$ is the sanctioning cost. Here, the cost of monitoring norm violations is a function of the accuracy of observations and the number of agents. Monitoring and sanctioning cost in this environment are defined, respectively, as:

$$\mathcal{MK}(\mathcal{D}, \mu) = \mathcal{MK}(\beta, \mu) = 1.1^{10\mu\beta} \times 10^{-3}\mathsf{R_N}$$

$$\mathcal{SK}(t) = \sum_{k=1}^{8} \mathcal{I}(\sigma_k, t)\ 0.05\mathsf{R_N}$$

where $\mathcal{I}(\sigma_k, t)$ yields the number of times that the sanction σ_k has been imposed in the time step t, and $0.05\mathsf{R_N}$ is the cost of applying any sanction in the motion environment. There are 8 norms in our normative system: one obligation per lane and the cost of sanctioning in each of them is the same. Using these functions, we can calculate the average norm enforcement cost per time step, denoted as $\overline{\mathcal{K}}$, in 10^6 time steps of simulation:

$$\overline{\mathcal{K}} = \frac{1}{10^6} \sum_{t=1}^{10^6} \mathcal{K}(t).$$

Note that each cost function is specific to the domain being simulated. For example, in simulated environment used in this paper, the monitoring cost increases exponentially with the detection probability β and the size of the population μ, and the cost of applying a sanction is fixed.

4.2 Results

In the absence of norms regulating the motion environment, the optimal choice would be taking the shortest path from the origin gateway to the destination gateway. However, in a norm regulated setting, agents adapt their behavior taking the sanctions into account. As the detection probability β rises, the agents tend to commit fewer violations. When $\beta > 0.09$, the agents start to comply with the norms because the expected utility gain of compliance outweighs that of violating them.

From the macro perspective, we estimate the effectiveness of various detection probabilities of norm violations β using the sum of penalties paid by the agents that violate norms (i.e. norm enforcement income). As we shall see, this income is used to fund the norm enforcement costs. For each β, Figure 3 shows the average norm enforcement income per time step (dotted line, left y-axis). We can see that high detection probabilities ($\beta > 0.09$) result in no income as these settings inhibit any violation of norms (starving the enforcement mechanism of funds). As the detection probability β decreases, the number of norm violations increases monotonically (solid line, right y-axis), whereas the enforcement income varies non-monotonically. This happens because the norm enforcement income is determined not only by β, but also by the number of norm violations, which in turn, is determined by the agents' policies.

We now look at an example of calculation of the norm enforcement income. Our experiments have shown that under detection probabilities $\{0.02, 0.03, 0.04\}$, agents stay on average 30.8%, 30.1% and 13.9% of their lifetime in violating states, respectively.

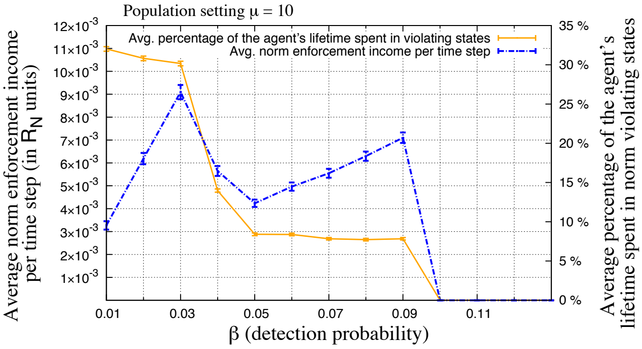

Fig. 3. Average norm enforcement income and percentage of time spent in violating states.

By multiplying these percentages by their respective β we find how many times, on average, the violations are detected: $\{0.00616, 0.00903, 0.00556\}$. Finally, if we multiply these values by the penalty value ($0.1\mathsf{R_N}$) and by the number of agents running simultaneously in the multiagent system ($\mu = 10$), we obtain the respective average enforcement income values shown in Figure 3:

$$(\beta = 0.02,\ 6.16 \times 10^{-3}\mathsf{R_N}),\ \ (\beta = 0.03,\ 9.03 \times 10^{-3}\mathsf{R_N}),\ \ (\beta = 0.04,\ 5.56 \times 10^{-3}\mathsf{R_N}).$$

If we increase the detection probability β from 0.02 to 0.03, the average income per time step increases from $6.16 \times 10^{-3}\mathsf{R_N}$ to $9.03 \times 10^{-3}\mathsf{R_N}$. However, if we increase β from 0.03 to 0.04, the agents change their policies (they perform less norm violations) and the income decreases to $5.56 \times 10^{-3}\mathsf{R_N}$.

The norm enforcement cost per time step $\overline{\mathcal{K}}$ is drawn in Figure 4 as a solid line with circles. If we subtract this cost from the average enforcement income (dotted line), we have the average profit (solid line with triangles). In our experiments, the system profits when $\beta \lessapprox 0.09$. Notice that the enforcement income increases as μ (number of agents simultaneously running in the environment) increases. Assuming that the detection cost does not depend on the value of μ, certain normative solutions may become profitable if we allow more agents to join the system.

Although Figure 4 shows that the highest profit occurs when $\beta = 0.03$, in multiagent systems with self-interested agents, such a low detection probability of norm violations may entail a significant amount of coordination problems, nullifying the effectiveness of the norms as a multiagent coordination device. For each detection probability β, Figure 5 shows the average percentage of agents that crash (dotted line, right y-axis) and the average profit of enforcing norms (solid line with triangles, left y-axis) when $\mu = 10$. In a system in which crashes (coordination problems) must not occur, any $\beta \geq 0.10$ can be employed, and $\beta = 0.10$ achieves norm compliance with the lowest cost (i.e. a larger enforcement intensity wastes resources).

If some coordination problems are acceptable, then the norm enforcement mechanism can consider a wider range of settings. For instance, any $\beta \geq 0.05$ can be used if a 20% maximum average percentage of crashes is acceptable, with the best option being $\beta = 0.09$ since it results in the highest income with a crash rate lower than 20%.

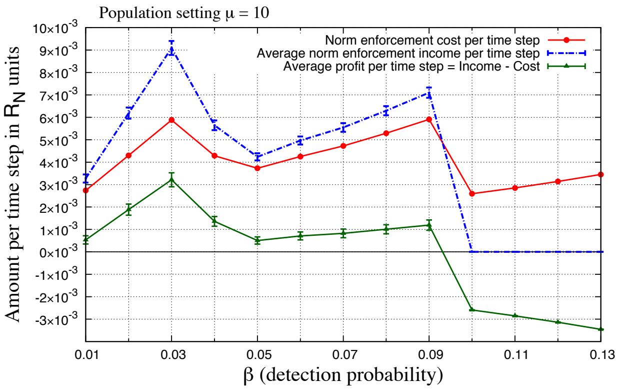

Fig. 4. Average norm enforcement cost, income and profit per time step.

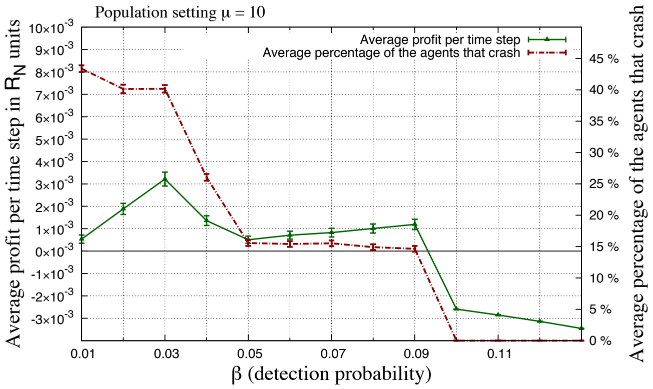

Fig. 5. Average percentage of agents that crash and average enforcement profit per time step.

5 CONCLUSION AND FUTURE WORK

In this paper we have defined a stochastic norm enforcement mechanism for agents operating in stochastic environments extending our earlier work and evaluated the tradeoffs involved when the effectiveness of the norm enforcement mechanism incurs a cost. Our evaluation led to a number of insights. From the agent's perspective, our experiments show that adequate norms can help shape the behavior of rational normative agents with an MDP-based world model, fostering coordination in a multiagent setting. From a macro perspective, we empirically show how our approach can estimate the global benefits of norms together with their enforcement cost in order to design economically viable norm enforcement mechanisms for systems populated by self-interested rational agents. In complex stochastic environments, our approach can be used to identify settings that make monitoring independent external funding (self-supporting).

We aim to extend this research in several promising ways. First, we can extend the agent model to account for the behavior of other agents or construct policies when the detection probabilities are unknown. Finally, although we have discussed insights in the context of one specific scenario, we aim to study ways to generalize the estimation of norm enforcement intensities.

Acknowledgements: This research is partially supported by the Spanish Ministry of Economy and Competitiveness through the projects AT (grant CSD2007-0022; CONSOLIDER-INGENIO 2010) and iHAS (TIN2012-36586-C03-02). The second and third authors acknowledge the support of the Brazilian National Council for Scientific and Technological Development (CNPq) under project Universal (grant 482156/2013-9), and PQ (grant 306864/2013-4).

References

1. Ågotnes, T., van der Hoek, W., Wooldridge, M.: Normative system games. In: Durfee, E.H., Yokoo, M., Huhns, M.N., Shehory, O. (eds.) AAMAS. pp. 881–888. IFAAMAS (2007)
2. Bellman, R.E.: Dynamic Programming. Dover Publications, Incorporated (2003)
3. Boutilier, C., Dean, T., Hanks, S.: Decision-Theoretic Planning: Structural Assumptions and Computational Leverage. J. Artif. Intell. Res. (JAIR) 11, 1–94 (1999)
4. Castelfranchi, C., Dignum, F., Jonker, C.M., Treur, J.: Deliberative Normative Agents: Principles and Architecture. In: Jennings, N.R., Lespérance, Y. (eds.) ATAL. Lecture Notes in Computer Science, vol. 1757, pp. 364–378. Springer (1999)
5. Fagundes, M.S.: Sequential Decision Making in Normative Environments. Ph.D. thesis, Universidad Rey Juan Carlos (2012)
6. Fagundes, M.S., Billhardt, H., Ossowski, S.: Reasoning about Norm Compliance with Rational Agents. In: Coelho, H., Studer, R., Wooldridge, M. (eds.) ECAI. Frontiers in Artificial Intelligence and Applications, vol. 215, pp. 1027–1028. IOS Press (2010)
7. Fagundes, M.S., Ossowski, S., Luck, M., Miles, S.: Using Normative Markov Decision Processes for evaluating electronic contracts. AI Commun. 25(1), 1–17 (2012)
8. Howard, R.A.: Dynamic Programming and Markov Processes. The M.I.T. Press (1960)
9. Jones, A.J.I., Sergot, M.: Deontic Logic in Computer Science: Normative System Specification, chap. On the characterisation of law and computer systems: the normative systems perspective, pp. 275–307. Wiley Professional Computing Series, Wiley (1993)
10. Modgil, S., Faci, N., Meneguzzi, F.R., Oren, N., Miles, S., Luck, M.: A framework for monitoring agent-based normative systems. In: Sierra, C., Castelfranchi, C., Decker, K.S., Sichman, J.S. (eds.) AAMAS (1). pp. 153–160. IFAAMAS (2009)
11. Nash Jr, J.F.: Equilibrium points in n-person games. Proceedings of the National Academy of Sciences 36, 48–49 (1950)
12. Omicini, A., Ossowski, S., Ricci, A.: Coordination infrastructures in the engineering of multiagent systems. In: Bergenti, F., Gleizes, M.P., Zambonelli, F. (eds.) Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook, Multiagent Systems, Artificial Societies, and Simulated Organizations, vol. 11, chap. 14, pp. 273–296. Kluwer Academic Publishers (2004)
13. Puterman, M.L., Shin, M.C.: Modified Policy Iteration Algorithms for Discounted Markov Decision Problems. Management Science 24, 1127–1137 (1978)
14. Schumacher, M., Ossowski, S.: The governing environment. In: Weyns, D., Parunak, H.V.D., Michel, F. (eds.) E4MAS. LNCS, vol. 3830, pp. 88–104. Springer (2005)
15. Tennenholtz, M.: On social constraints for rational agents. Computational Intelligence 15(4), 367–383 (1999)
16. Ummels, M., Wojtczak, D.: The complexity of nash equilibria in stochastic multiplayer games. Logical Methods in Computer Science 7(3) (2011)