Synthesising Liberal Normative Systems

Javier Morales[1,2], Maite Lopez-Sanchez[1]
[1]Universitat de Barcelona
Barcelona, Spain
jmorales@iiia.csic.es
maite@maia.ub.es

Juan A. Rodriguez-Aguilar
[2]Artificial Intelligence Research Institute (IIIA-CSIC)
Campus UAB. Bellaterra, Spain
jar@iiia.csic.es

Michael Wooldridge
Dept. of Computer Science
University of Oxford
Oxford, UK
mjw@cs.ox.ac.uk

Wamberto Vasconcelos
Dept. of Computing Science
University of Aberdeen
Aberdeen, UK
wvasconcelos@acm.org

ABSTRACT

Norms have been extensively studied to coordinate multi-agent systems, and the literature has investigated two general approaches to norm synthesis: off-line (synthesising norms at design-time) and on-line (run-time synthesis). On-line synthesis is generally recognised to be appropriate for open systems, where aspects of the system remain unknown at design-time. In this paper we present LION, an algorithm aimed at synthesising *liberal* normative systems. LION's normative systems respect the agents' autonomy to the greatest possible extent, constraining their behaviour when only necessary to avoid undesirable system states. LION's norm synthesis is also driven by the need to construct *compact* normative systems. The key to the success of LION in this multi-objective synthesis process is that it learns about and exploits norm synergies. More precisely, LION can learn when norms are either substitutable or complementary. We show empirically that LION significantly outperforms the state of the art by synthesising normative systems that are more liberal while maintaining representation compactness.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence —*Multiagent Systems*

General Terms

Algorithms

Keywords

Norms - Normative Systems - On-line Norm Synthesis

1. INTRODUCTION

Norms have been extensively studied as a technique for coordinating interactions within multi-agent systems (MAS) [6, 4]. However, computing a normative system (i.e., a set of norms) that will effectively coordinate a multi-agent system has been identified as a computationally complex (NP-hard) problem [21]. Two general approaches to the design of norms have been investigated in the literature: off-line [21, 7, 1] (design time synthesis) and on-line [17, 18, 8, 15, 3] (run-time synthesis). It is generally recognised that on-line synthesis is better suited than off-line synthesis for open systems, where aspects of the system (such as the system state space or the range of possible agent behaviours) are unknown at design-time.

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

Recent research on on-line synthesis has focused on norm emergence [20, 19, 9, 23], which considers that agents collaborate to synthesise their own norms. However, this line of research makes several strong assumptions, namely that agents will participate in the norm synthesis process, and will have the appropriate computational capabilities to synthesise norms. These assumptions are not required in a family of alternative on-line norm synthesis approaches [11, 12, 13] that synthesise normative systems from the observation of agents' interactions. On the one hand, they synthesise norms that are *effective* to avoid undesirable states of a MAS (i.e., conflicts) as long as agents comply with them. On the other hand, they aim at avoiding over-regulation by preserving only those norms that are *necessary* to avoid such conflicts. Based on [11], the IRON [12] and SIMON [13] systems focus on synthesising *compact* normative systems. Compactness is concerned with minimising both the number of norms that are given to the agents as well as their norm reasoning computational effort. With this aim, they perform norm generalisations that represent several (specific) norms by means of general (abstract) norms. The normative systems that result have fewer and more general norms (and thus, easier to reason about) than [11]. In particular, SIMON proved to be the *best-in-class* approach, outperforming IRON in terms of compactness.

However, even though IRON and SIMON minimise the number of norms, they do not take into account agents' actual freedom, a key synthesis criterion that considers how norms constrain their behaviour. This is so because a general norm (such as, for example, "give way to any approaching car" in a traffic scenario), represents several specific norms ("give way to left", "give way to right", etc.) which together restrict agents' behaviour to a high extent. Therefore, the smaller the number of constraints in a normative system, the greater freedom for the agents. Along this line of preserving autonomy, the work in [7] proposes to minimise the number of constraints that agents are required to obey. Additionally, the work in [1] introduces *liberality* as a relationship between normative systems. It states that a normative system Ω is *more liberal* (less restrictive) than another one Ω' if it places fewer constraints on the agents. Thus, following the traffic example, we can state that a normative system with a "give way to left" norm is *more liberal* than one with a norm "give way to any approaching car".

Aiming at synthesising liberal normative systems without compromising compactness, we present LION (*LI*beral *O*n-line *N*orm synthesis). Based on [13], LION aims at synthesising normative systems that (i) avoid undesired states (i.e., conflicts) in a system; (ii) minimise the number of norms; and (iii) maximise agents' freedom (i.e., autonomy) by minimising the number of imposed constraints. The key to the success of LION in this multi-objective synthesis process is that it learns about and exploits norm synergies. More precisely, LION can learn when norms are either *substitutable* or *complementary*. Substitutability and complementarity [16, 10] are well-known concepts in economics that apply to a wide range of domains [2, 25, 22]. While two substitute goods may replace one another in use, complementary goods are better used together. By considering such relationships, LION outperforms SIMON in terms of liberality without compromising criteria (i) and (ii). In 96% of our simulations using a simple traffic domain LION synthesised normative systems that contained, on average, 90% fewer substitutability relationships than those synthesised by SIMON.

The remainder of this paper is organised as follows. Section 2 provides the necessary background about previous norm synthesis approaches. Next, Section 3 introduces concepts used in LION, which is presented in Section 4. Section 5 reports our empirical evaluation and Section 6 concludes the paper.

2. BACKGROUND

As previously introduced, the work reported in [11, 12, 13] constitutes a family of on-line norm synthesis approaches that synthesise normative systems from the observation of agents' interactions. In what follows, we present the key concepts of SIMON [13], which proved to be the best-in-class approach.

Briefly, SIMON synthesises norms by continuously monitoring a system at runtime, searching for undesirable states (which we refer to as *conflicts*). Whenever a conflict arises, it proposes a new norm (aimed at avoiding the conflict in the future) and provides the updated set of active norms to the agents. Agents can then decide whether to comply or not with those norms that apply to them at each time step. Norm fulfilment/infringement decisions may have effects in terms of future conflicts. SIMON gathers this information as norm performance is evidenced (i.e., if they where effective/necessary to avoid conflicts). Norm evaluation can then be used to refine the normative system by (i) generalising well-performing norms that can be represented in a more compact manner, and (ii) discarding (or specialising) under-performing norms.

SIMON considers a system populated by a set of agents Ag that can perform a set of actions $Ac \subseteq \mathcal{L}_{Ag}$. The *context* of an agent is a formula of a first-order logic language $\mathcal{L}_{Ag}$ that describes the local perception of that agent (i.e., the agent's own point of view). $\mathcal{L}_{Ag}$ is composed of first-order predicates $p(\tau_1, \ldots, \tau_n)$, where p is a predicate symbol and $\tau_1, \ldots, \tau_n$ are terms of $\mathcal{L}_{Ag}$. A *norm* is then defined so that it will apply to an agent whenever the context of that agent satisfies the precondition of the norm. Specifically, norms are of the form $\langle \varphi, \theta(ac) \rangle$, where φ is the norm's precondition and $\theta(ac)$ its post-condition (where φ is a formula of $\mathcal{L}_{Ag}$, θ is a deontic operator such as a *prohibition*, and $ac \in Ac$). We say $\theta(ac)$ holds for any agent whose context satisfies φ. Furthermore, we say a norm is *specific* iff all its terms are grounded.

SIMON keeps synthesised norms as nodes in a directed graph data structure called the *Normative Network* (*NN*), so that it represents current normative system Ω as its set of active norms. Moreover, edges in SIMON's *NN* stand for norm *generalisation* relationships. SIMON creates general norms that compactly represent specific norms. Generalisation may lead to over-generalisations that will be backtracked by specialization whenever one of the specific norms under-performs.

In order to illustrate these concepts, we introduce a traffic example scenario in which $ambulance, police$ represent different types of $emergency$ vehicles; $car, motorbike$ describe $private$ vehicles; and any represents both $emergency$ and $private$ terms. Figure 1a depicts a normative network composed of four specific active norms n_1 : *"Give way to ambulances"*, n_2 : *"Give way to police"*, n_3 : *"Give way to cars"*, and n_4 : *"Give way to motorbikes"*. Since *ambulances* and *police cars* are both *emergency* vehicles, SIMON

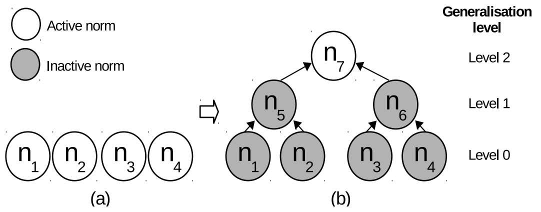

Figure 1: Generalisation of norms $n_1, n_2, n_3, n_4, n_5, n_6$.

will create a general norm n_5 : *"Give way to emergency vehicles"* to represent (and establish a generalisation relationship with) n_1 and n_2. Similarly, a new general norm n_6 : *"Give way to private vehicles"* will generalise norms n_3, n_4, which in turn will lead to the creation of a new norm n_7 : *"Give way to any vehicle"* with the highest generalisation level. Figure 1b shows the resulting normative network, where $\Omega = \{n_7\}$. Afterwards, in case n_1 underperforms, SIMON will deactivate n_1, n_5, n_7 and will activate n_2, n_6 so that the normative system becomes updated ($\Omega = \{n_2, n_6\}$).

Generalisation thus provides representation compactness but, as introduced in Section 1, general norms can represent an arbitrary number of constraints (i.e., specific norms) and, in particular, some of them may not be required to avoid conflicts. Thus, generalisations may over-constrain the system, restricting the freedom of agents. Indeed, in [13] SIMON was employed to synthesise norms and this was the case. There, a slightly different traffic scenario was used, where each car described its perceptions by means of predicates $left, front, right$, each predicate containing one term out of $\{car\text{-}to\text{-}left, car\text{-}to\text{-}right, car\text{-}same\text{-}heading, wall, nil, any\}$. The first three terms represent a car heading in different directions with respect to the reference car, term $wall$ represents the perception of a wall, nil means that a car does not perceive anything, and any stands for car, $wall$ or nil. Table 1 shows SIMON's most-frequently synthesised normative system, which was obtained by executing SIMON's publicly available code [14]. This normative system resulted in 90% of the tests and proved to avoid collisions when norms were fulfilled just by having five explicit general norms. Nevertheless, this compact normative system implicitly represents (i.e., includes) 18 different constraints[1] and not all of them are really required. In fact, n_a represents a general norm of giving way to the left (that is, a car should stop whenever it perceives a car to its left, and no matter what it perceives to its front and right positions) and n_b is a general norm to give way to the right[2]. It is worth noticing that n_a does not generalise n_b nor the other way around, and thus they are not related from SIMON's point of view. However, although they are neither syntactically related, if we could go a step further and consider semantics, then we could infer that the normative system could still prevent collisions by just having one of these two norms. Therefore, there is room for finding an alternative normative system that is more liberal (poses fewer constraints to agents) than that obtained by SIMON.

Nevertheless, removing constraints from a normative system is not straightforward, since norm synergies within the normative system need to be taken into account. In the previous example from Table 1, specific norms n_a and n_b can replace one another, and thus we say they are *substitutable*. However, the proper performance of some specific norms may depend on the existence of other specific norms. If this is the case, then we say that these specific norms complement each other: they are *complementary*. Furthermore,

[1] Notice that the addition of #Constr. column in Table 1 results in 28, but several general norms generalise to the same specific norm.

[2] The remaining norms n_c, n_d, n_e ensure a car stops whenever it perceives a car ahead with different headings.

Norm	Pre-condition (θ)	Norm target	#Constr.
n_a	*left(car-to-right), front(any), right(any)*	*prh(Go)*	6
n_b	*left(any), front(any), right(car-to-left)*	*prh(Go)*	6
n_c	*left(any), front(car-same-heading), right(any)*	*prh(Go)*	8
n_d	*left(any), front(car-to-left), right(any)*	*prh(Go)*	4
n_e	*left(any), front(car-to-right), right(any)*	*prh(Go)*	4

Table 1: A typical SIMON's normative system upon convergence, along with the number of constraints each norm represents.

there may be substitutable norms that are complementary to others, and this should be considered when discarding norms if we want to preserve the effectiveness of the normative system (i.e., its performance in avoiding undesired states).

We thus conclude that SIMON's normative systems are overconstrained, but that aiming at synthesising more liberal normative systems amounts to managing norm synergies in this multi-objective synthesis process. More precisely, substitutability and complementarity relationships between specific norms need to be successfully learned and exploited.

3. CHARACTERISING NORM SYNERGIES

This section formally characterises the notion of synergies between norms. In particular, we focus on two types of norm synergies: substitutability and complementarity. On the one hand, we consider that two norms are substitutable if they satisfy the same regulatory requirements and, therefore, can be used to replace one another. On the other hand, we consider that two norms are complementary if they are better when used together instead of separately. Next, in section 3.1 we provide some preliminary definitions, while in section 3.2 we provide the formal characterisation of both norm relationships. Finally, we also discuss the relationship between complementarity, substitutability, and generalisation.

3.1 Preliminary definitions

Our basic definitions follow those of [13] presented in Section 2. Thus, we consider a multi-agent system composed of a set of agents Ag; a set of actions $Ac = \{ac_1, \ldots, ac_m\}$ available to the agents; a set S of states of the system, where a state s is a set of ground atomic formulae from $\mathcal{L}_{Ag}$; and a set $C \subset S$ of undesirable states. The set of active norms is made available to the agents as a normative system (e.g., Table 1).

Furthermore, we consider a *state transition function* $\mathbb{T} : S \times Ac^{|Ag|} \rightarrow S$ that leads the MAS to a state s' from a state s after the agents perform a collection of actions $A \subseteq Ac^{|Ag|}$. We will denote a transition from a state s to a state s' after the performance of A by $\langle s, A, s' \rangle$. For convenience, Ac includes a special action nil that stands for not performing any action. Then, given a transition $\langle s, A, s' \rangle$, function $action : Ag \times S \times S \rightarrow Ac$ returns the action that an agent performed during the transition. We assume each agent has its own perception of the MAS it is part of. Such perception is a partial representation of the MAS from the agent's local point of view. Thus, we consider that an agent ag perceives the MAS at a state s by means of its local function *context* $c : Ag \times S \rightarrow \mathcal{P}(\mathcal{L}_{Ag})$, which returns a set of predicates with terms of language $\mathcal{L}_{Ag}$. Besides, we will also consider that an agent can tell what other agents it can perceive. This is captured by the scope function $sc : Ag \times S \rightarrow \mathcal{P}(Ag)$, which returns the set of agents that an agent perceives at a given state. When an agent ag' is in the scope of agent ag, we say that ag' is detected by ag.

We now introduce the core concepts of norm fulfilment and norm infringement. We first establish when a norm applies: $\langle \varphi, \theta(ac) \rangle$ *applies* to agent ag at state s if the agent's local context $c(ag, s)$ satisfies the precondition of the norm, namely iff $c(ag, s) \models \varphi$. Let there be $\langle \varphi, \theta(ac) \rangle$ which applies to agent ag at state s, and a transition $\langle s, A, s' \rangle$; we then say that the agent *fulfilled* the norm during the state transition if either it did not perform an action prohibited by the norm or it performed an action obliged by the norm.

DEF. 1 (NORM FULFILMENT). *Let $s \in S$ be a state, $ag \in Ag$ an agent, and $n = \langle \varphi, \theta(ac) \rangle$ be a norm that applies to ag at s. We say that ag fulfilled norm n during a transition $\langle s, A, s' \rangle$ when either: (i) $\theta = prh$ and $action(ag, s, s') \neq ac$; or (ii) $\theta = obl$ and $action(ag, s, s') = ac$.*

As a dual concept to norm fulfilment, we consider that an agent *infringed* a norm during a state transition if it performed an action that the norm prohibited or it did not perform an action that the norm enforced.

DEF. 2 (NORM INFRINGEMENT). *Let $s \in S$ be a state, $ag \in Ag$ an agent, and $n = \langle \varphi, \theta(ac) \rangle$ a norm that applies to ag at s. We say that ag infringed norm n during a transition $\langle s, A, s' \rangle$ when either: (i) $\theta = prh$ and $action(ag, s, s') = ac$; or (ii) $\theta = obl$ and $action(ag, s, s') \neq ac$.*

Next, we provide our formal characterisation of norm synergies.

3.2 Substitutability and complementarity

As mentioned above, two norms are substitutable if they satisfy the same regulatory needs and, therefore, can be used to replace one another, whereas two norms are complementary if they perform better when used together. Therefore, in order to assess whether any of these two relationships holds between a pair of norms, we will have to assess the difference in outcome between the fulfilment of the two norms at the same time and their individual fulfilments. Thus, substitutability between two norms will hold when the concurrent fulfilment of the two norms avoids undesired states, but also does the individual fulfilment of any of the two norms (while the other one is infringed). By contrast, complementarity between two norms will hold when only the concurrent fulfilment of both norms avoids undesired states, whereas the individual fulfilment of any of the two norms does not.

However, before formalising substitutability and complementarity we need to establish when two norms concurrently apply. First, we consider the notion of joint context. We say that two agents share a joint context in a given state if they can detect (perceive) one another. Formally,

DEF. 3 (JOINT CONTEXT). *Let $s \in S$ be a state, and $ag, ag' \in Ag$ two agents with contexts $c(ag, s)$, $c(ag', s)$ and scopes $sc(ag, s)$, $sc(ag', s)$. We say that $\langle c(ag, s), c(ag', s) \rangle$ is a joint context shared by the agents iff $ag' \in sc(ag, s)$ and $ag \in sc(ag', s)$.*

Then, when two agents share a joint context and there are two norms such that there is one norm that applies to each agent, we say that the norms concurrently apply. Formally,

DEF. 4 (CONCURRENT APPLICABILITY). *Let $s \in S$ be a state, $ag, ag' \in Ag$ two agents that share a joint context $\langle c(ag, s), c(ag', s) \rangle$, and $n = \langle \varphi, \theta(ac) \rangle$, $n' = \langle \varphi', \theta'(ac') \rangle$ two different norms. We say that n, n' concurrently apply in the joint context iff $c(ag, s) \models \varphi$ and $c(ag', s) \models \varphi'$.*

We now formalise the notion of substitutability. We notice though that we will consider that substitutability only holds between *specific norms* as defined in Section 2. Thus, for example, $\langle \{left(car\text{-}to\text{-}right), front(nil), right(nil)\}, prh(Go) \rangle$ is a specific norm generalised (represented) by n_a in Table 1. Then, we say that two specific norms that concurrently apply are substitutable in a particular state when only one of them is required to avoid a transition to an undesired state. Formally,

DEF. 5 (SUBSTITUTABILITY). *Let $s \in S$ be a state, $C \subseteq S$ a set of undesired states, and $ag, ag' \in Ag$ two agents that share a joint context $\langle c(ag, s), c(ag', s)\rangle$. Let n, n' be two different, specific norms such that n applies to ag and n' applies to ag', and they concurrently apply in the agents' joint context. Let $\langle s, A, s'\rangle$ be the transition that results after ag and ag' perform actions $ac \in A$ and $ac' \in A$ respectively. Norms n, n' are substitutable in s iff the following conditions hold: (i) a conflict occurs ($s' \in C$) when both agents infringe their norms; and (ii) no conflict occurs ($s' \notin C$) when at least one of the agents fulfils its applicable norm.*

In Section 2 we illustrated substitutability with two norms n_a: "give way to left" and n_b: "give way to right". Notice that both norms concurrently apply together in a situation in which (i) two cars perceive each other; and (ii) each norm applies to only one of the cars. In this situation, collisions can be avoided by employing only one of the norms. Therefore, employing both norms would over-constrain the situation. We say then that both norms substitute one another, namely they are substitutable, since any of them could satisfactorily regulate the situation.

From the definition above, we say that two norms are substitutable in a set of states if they are substitutable at each state. Consider now all the states of a system where two norms concurrently apply, we say that the two norms are *fully substitutable* if they are substitutable in all those states. We notice that the substitutability relationship is irreflexive, and symmetric, but non-transitive.

Next we formalise the notion of complementarity. We say that two norms that concurrently apply are complementary in a particular state when the two of them are required to avoid a transition to an undesired state. Formally,

DEF. 6 (COMPLEMENTARITY). *Let $s \in S$ be a state, $C \subseteq S$ a set of undesired states, and $ag, ag' \in Ag$ two agents that share a joint context $\langle c(ag, s), c(ag', s)\rangle$. Let n, n' be two different, specific norms such that n applies to ag and n' applies to ag', and they concurrently apply in the agents' joint context. Say that $\langle s, A, s'\rangle$ is the transition that results after ag and ag' perform actions $ac \in A$ and $ac' \in A$ respectively. Norms n, n' are complementary in s iff the following conditions hold: (i) a conflict occurs ($s' \in C$) when at least one of the agents infringes its applicable norm; and (ii) no conflict occurs ($s' \notin C$) when both agents fulfil their norms.*

From the definition above, we say that two norms are complementary in a set of states if they are complementary at each state. We say that the two norms are *fully complementary* if they are complementary in all the states of the system where they concurrently apply. Like substitutability, the complementarity relationship is irreflexive, and symmetric, but non-transitive.

OBSERVATION 1. *Generalisation, substitutability, and complementarity are mutually exclusive relationships.*

To see this, first observe that substitutability and complementarity are mutually exclusive relationships: two norms that are substitutable cannot be complementary at the same time and the other way around. This directly follows from the conditions in definitions 5 and 6. Furthermore, generalisation and substitutability are mutually exclusive. This also comes from the definition of generalisation and substitutability. According to definition 5, substitutability only holds between specific norms, whereas generalisation always requires at least a non-specific norm. From this it follows that generalisation and substitutability cannot hold at the very same time between two norms. Following the same line of reasoning, since complementarity only holds between specific norms, generalisation and complementarity are mutually exclusive.

This observation tells us that the norm synthesis algorithm described in [13], which only learned generalisation relationships, never learned any substitutability relationship nor any complementarity relationship. In the next section we show how a norm synthesis process can take advantage of substitutability and complementarity relationships. The aim of our normative synthesis process will be: (i) to exploit generalisation relationships to yield a compact normative system; (ii) to exploit substitutability to disregard over-constraining norms; and (iii) to exploit complementarity to keep together norms that are better off together.

4. SYNTHESISING LIBERAL NORMATIVE SYSTEMS

We now describe LION (*LI*beral *O*n-line *N*orm synthesis), our novel multi-objective norm synthesis mechanism. LION aims to synthesise compact normative systems that avoid undesirable states (i.e., conflicts) while respecting the autonomy of agents to the greatest possible extent. The main feature of LION is that it is capable of learning and exploiting (benefitting from) norm synergies during the norm synthesis process. More precisely, LION identifies when norms are either substitutable or complementary, book-keeps these relationships between norms, and uses this knowledge (together with the generalisation relationships) to synthesise liberal normative systems that perform well and do not compromise compactness. This requires carefully handling the interplay between all norm relationships at synthesis time.

Specifically, LION continuously monitors agents' interactions at run-time, searching for conflicts and creating norms to avoid them. LION publishes the normative system that it currently handles (the current active norms) so that agents make their decision considering whether to fulfil or infringe its norms. The outcome of agents' decisions may (or may not) lead to conflicts. LION captures such outcomes to evaluate how norms perform individually and to detect pair-wise relationships between specific norms. From this, LION updates the current normative system as follows. On the one hand, LION employs the individual performance of norms to either perform generalisations (when norms perform well) or specialisations (when a norm performs poorly). On the other hand, LION exploits norm relationships to discard (deactivate) norms involved in substitutability relationships. The heuristic employed to make this decision is simple: given two substitutable norms, choose to discard the one that causes less *compactness loss* provided that it is not part of any complementarity relationships. Since deactivating a norm leads to specialisation, and hence to losing compactness, LION chooses to discard the norm that minimises such loss. Overall, generalisation pursues compactness, complementarity safeguards performance, and discarding substitutable norms provides liberality.

This section is thus organised as follows. Section 4.1 details how LION represents substitutability and complementarity relationships, Section 4.2 explains how it detects these relationships and Section 4.3 details how it exploits them. Finally, Section 4.4 presents LION's multi-objective norm synthesis strategy, and Section 4.5 defines metrics to evaluate normative systems.

4.1 Representing norm relationships

First, LION generalises the notion of *Normative Network* introduced in [12], and referred to in Section 2, to represent norm relationships as follows:

DEF. 7 (GENERALISED NORMATIVE NETWORK). *A generalised normative network (GNN) is a tuple* $\langle \mathcal{N}, \mathcal{E}_{\mathcal{G}}, \mathcal{E}_{\mathcal{S}}, \mathcal{E}_{\mathcal{C}}, \Delta, \delta \rangle$ *where:* (i) $(\mathcal{N}, \mathcal{E}_{\mathcal{G}} \cup \mathcal{E}_{\mathcal{S}} \cup \mathcal{E}_{\mathcal{C}})$ *is a graph such that:* $\mathcal{N}$ *is a set of norms that correspond to the nodes in the graph;* $\mathcal{E}_{\mathcal{G}} \subseteq \mathcal{N} \times \mathcal{N}$ *is a set of directed edges representing generalisation relationships* ($(n, n') \in \mathcal{E}_{\mathcal{G}}$ *means that* n' *generalises* n*);* $\mathcal{E}_{\mathcal{S}} \subseteq \mathcal{N} \times \mathcal{N}$ *is a set of undirected edges standing for substitutability relationships;* $\mathcal{E}_{\mathcal{C}} \subseteq \mathcal{N} \times \mathcal{N}$ *is a set of undirected edges standing for complementarity relationships; and* $\mathcal{E}_{\mathcal{G}}, \mathcal{E}_{\mathcal{S}}, \mathcal{E}_{\mathcal{C}}$ *are pair-wise disjoint.* (vi) $\Delta = \{Active, Inactive\}$ *is the set of possible states of a norm in* $\mathcal{N}$*;* (vii) $\delta : \mathcal{N} \rightarrow \Delta$ *yields the state of a norm in* $\mathcal{N}$.

Figure 2b shows a GNN built by LION after detecting substitutability and complementarity relationships between norms. We notice that specific norms n_9 and n_1 hold a substitutability relationship; specific norms n_9 and n_{11} hold a complementarity relationship; and the rest of relationships (directed edges) stand for generalisations (n_{10} generalises n_8, n_9 and n_7 generalises norms n_1–n_6 at different generalisation levels). Since norms n_{11}, n_{10}, n_7 are the only ones active (denoted by white circles), the GNN represents the normative system $\Omega = \{n_{11}, n_{10}, n_7\}$.

4.2 Detecting norm relationships

Next we describe how LION learns, at run-time, substitutability and complementarity relationships between specific norms in the GNN. We recall from Section 3.2 that a pair of concurrently applicable norms are *substitutable* if no conflicts arise whenever at least one of them is fulfilled. Similarly, they are considered *complementary* if any conflicts arise whenever at least one of them is infringed. LION is an on-line process that progressively gathers evidences, as they occur, to support whether a pair of norms are either substitutable or complementary. Thus, given a pair of norms, if the substitutability conditions in Def. 5 occur a sufficient number of times, namely for a sufficient number of states, LION considers that the relationship holds. The same applies to complementarity. When LION learns that either a substitutability or complementarity relationship holds, it does represent it in the GNN.

LION proceeds as follows. Whenever it detects a new pair of concurrently applicable norms (n, n'), it creates three utility series

$$\langle \mathcal{U}^{FF}_{n,n'}, \mathcal{U}^{FI}_{n,n'}, \mathcal{U}^{IF}_{n,n'} \rangle$$

that accumulate the frequency in avoiding conflicts along time whenever n, n' are fulfilled (F) or infringed (I). $\mathcal{U}^{FF}_{n,n'}$ is a binary series $\langle u^{FF}_1, \ldots, u^{FF}_m \rangle$, where u^{FF}_i gathers the evidence related to the i-th time that n and n' were concurrently applicable and were both fulfilled (FF). Specifically, $u^{FF}_i = 1$ when both fulfilments did not lead to an undesired state, and 0 otherwise. Analogously, we define $\mathcal{U}^{FI}_{n,n'}$ and $\mathcal{U}^{IF}_{n,n'}$ for series where n is fulfilled and n' is infringed, and n is infringed and n' is fulfilled, respectively.

Gathering pair-wise evidences does not cover dependencies with other norms in the current normative system, although accounted undesired states may in fact be caused by third norms. This "noisy" evidences cause fluctuations in the binary series. Therefore, as it is usually the case for data streams, LION employs the *cumulative moving average* [5] to smooth out short-term fluctuations and highlight longer-term trends when detecting substitutability and complementarity. The cumulative average of a series $\mathcal{U}^{FF}_{n,n'}$ is another series $\hat{\mathcal{U}}^{FF}_{n,n'} = \langle \hat{u}_1, \ldots \hat{u}_m \rangle$, where $\hat{u}^{FF}_i$ is computed as follows:

$$\hat{u}^{FF}_i = \frac{\sum_{u_j \in \mathcal{U}^{FF}_{n,n'}, 0 \le j \le i} u_j}{i} \tag{1}$$

the elements in $\hat{\mathcal{U}}^{FI}_{n,n'}$ and $\hat{\mathcal{U}}^{IF}_{n,n'}$ are computed analogously.

We consider that two concurrently applicable norms n, n' are substitutable whenever their cumulative averaged series *are similar*. Since the Euclidean distance is one of the most used and efficient time series (dis)similarity measures [24], we assess the similarity between our series by means of the *averaged Euclidean distance*, computed as follows:

$$Distance(\hat{\mathcal{U}}^{FF}_{n,n'}, \hat{\mathcal{U}}^{FI}_{n,n'}) = \frac{\sqrt{\sum_{i=1}^{l_{min}} (\hat{u}^{FF}_i - \hat{u}^{FI}_i)^2}}{l_{min}} \tag{2}$$

being l_{min} the minimum length of series $\hat{\mathcal{U}}^{FF}_{n,n'}$ and $\hat{\mathcal{U}}^{FI}_{n,n'}$, namely $l_{min} = min(|\hat{\mathcal{U}}^{FF}_{n,n'}|, |\hat{\mathcal{U}}^{FI}_{n,n'}|)$.

Thus, LION determines that two concurrently applicable norms n and n' are substitutable iff: (i) it accumulates *enough* evidence regarding the outcomes of the concurrent application of both norms; and (ii) the distance between series $\hat{\mathcal{U}}^{FF}_{n,n'}$ and $\hat{\mathcal{U}}^{FI}_{n,n'}$ as well as the distance between series $\hat{\mathcal{U}}^{FF}_{n,n'}$ and $\hat{\mathcal{U}}^{IF}_{n,n'}$ are both below a given threshold $\alpha_{sim} > 0$. This amounts to verifying whether the following conditions hold:

$$|\hat{\mathcal{U}}^{FF}_{n,n'}| \ge eval_{min}, |\hat{\mathcal{U}}^{FI}_{n,n'}| \ge eval_{min}, |\hat{\mathcal{U}}^{FI}_{n,n'}| \ge eval_{min} \tag{3}$$

$$Distance(\hat{\mathcal{U}}^{FF}_{n,n'}, \hat{\mathcal{U}}^{FI}_{n,n'}) \le \alpha_{sim} \tag{4}$$

$$Distance(\hat{\mathcal{U}}^{FF}_{n,n'}, \hat{\mathcal{U}}^{IF}_{n,n'}) \le \alpha_{sim} \tag{5}$$

Along the same lines, LION considers that two concurrently applicable norms are complementary whenever their series $\hat{\mathcal{U}}^{FF}_{n,n'}$ is greater than series $\hat{\mathcal{U}}^{FI}_{n,n'}$ and $\hat{\mathcal{U}}^{IF}_{n,n'}$, and all them have a minimum number of evidences. Formally, it corresponds to satisfying equations 3, 6 and 7:

$$Distance(\hat{\mathcal{U}}^{FF}_{n,n'}, \hat{\mathcal{U}}^{FI}_{n,n'}) > \alpha_{sim} \tag{6}$$

$$Distance(\hat{\mathcal{U}}^{FF}_{n,n'}, \hat{\mathcal{U}}^{IF}_{n,n'}) > \alpha_{sim} \tag{7}$$

Whenever LION detects that two norms are substitutable or complementary, it establishes the corresponding relationship in the GNN.

4.3 Exploiting norm relationships

As previously discussed, generalisation pursues compactness, complementarity safeguards performance and discarding substitutable norms provides liberality. Hence, at run-time, when LION detects two substitutable norms, it should discard one of them, since it is over-constraining the agents in the MAS. Nevertheless, LION does not proceed directly, on the contrary, it exploits the relationships of both norms so that the less generalised substitutable norm is discarded provided that it is not complementary with another norm.

We recall from Section 2 that deactivating specific norms implies backtracking over (that is, undoing) norm generalisations, hence leading to less compact normative systems. Therefore, before discarding a substitutable norm, LION considers the ancestors of each norm to determine its corresponding *compactness loss* and to choose the norm having the lowest value. As detailed below, the compactness loss of a norm n is related to the decrement in compactness that the normative system will suffer in case norm n is deactivated, or, in other words, its associated specialisation cost.

As an example, consider the GNN in Figure 2a representing the normative system $\{n_7, n_{10}\}$. It contains two substitutable norms n_1, n_9 that are generalised by other norms. In particular, n_9 is represented by an active norm n_{10} whose generalisation level is 1, and n_1 is represented by an active norm n_7 whose generalisation level is 2. Because of being more general, n_7 compactly represents

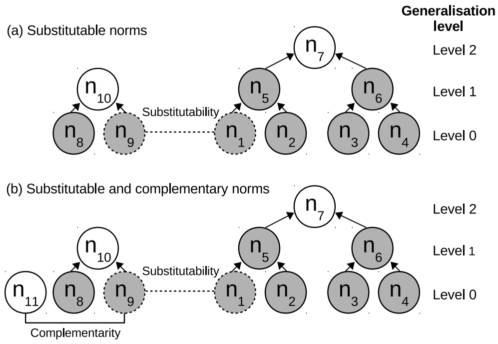

Figure 2: Norm relationships: solid arrows stand for (non-symmetric) generalisation; dashed lines for substitutability (symmetric) and solid lines for complementarity (symmetric).

a greater number of norms than n_{10}. At this point, LION must decide whether to discard n_9 or n_1 because they are substitutable. Since deactivating n_1 would imply in specialising n_7, we consider that n_1 has a higher specialisation cost than n_9, and thus, LION will deactivate norm n_9 instead.

Specifically, LION uses equation 8 to compute the compactness loss of a norm n as the sum of the generalisation degrees of its ancestors in the GNN graph.

$$\mathcal{C}_{loss}(n, GNN) = \sum_{n' \in \, ancestors(n)} g_{deg}(n', GNN) \qquad (8)$$

where $g_{deg}(n', GNN)$ is the generalisation degree of a norm n' in the normative network, which is computed by means of equation 9:

$$g_{deg}(n, GNN) = |children(n)| \cdot k^{level(n)} \qquad (9)$$

where: $children(n)$ is a function that assesses the norms that n directly generalises (those norms n has an incoming generalisation relationship with); $level(n)$ is the generalisation level of n; and $k > 1$ a constant factor.

Figure 2b helps to illustrate these computations. The compactness loss of norm n_1 is the sum of the generalisation degree of its ancestors n_5, n_7. In particular, since n_5 has a generalisation level of 1 and has two children (n_1, n_2), its generalisation degree is $g_{deg}(n_5, GNN) = 2k$. Analogously, norm n_7 has a generalisation degree $g_{deg}(n_5, GNN) = 2k^2$ because its generalisation level is 2 and it has 2 children (n_5 and n_6). As a result, $\mathcal{C}_{loss}(n_1, GNN) = 2k + 2k^2$, which is greater than $\mathcal{C}_{loss}(n_9, GNN) = 2k$. Therefore, LION chooses to deactivate n_9.

Regarding complementarity relationships, Figure 2b depicts an alternative situation during the synthesis process where the normative network contains, not only substitutable norms n_1, n_9, but also a complementarity relationship between norm n_9 and norm n_{11}. In this case, since LION prioritizes the preservation of complementary norms, it will keep n_9 active and will choose to deactivate n_1.

Algorithm 1 details how LION exploits norm relationships to choose which norm to discard (i.e., deactivate) from a pair of substitutable norms. Given two substitutable norms n_A, n_B, it first checks if any of them is complementary to other norms in the normative network (lines 3-4) in order to preserve complementary norms (lines 5-8). If none of them is complementary, then it marks to substitute the norm with the lowest compactness loss (lines 9-13). Otherwise, in case there are no complementarity relationships with other norms and their compactness losses are equal, then LION randomly marks to deactivate one of them (line 15). It then returns (line 16) the norm that has been chosen to be substituted.

Algorithm 1 Choosing a substitutable norm to discard

1: **function** CHOOSETODISCARD(n_A, n_B, GNN)
2: $\quad toDiscard \leftarrow null$
3: $\quad nACompl \leftarrow hasComplementarity(n_A, GNN)$
4: $\quad nBCompl \leftarrow hasComplementarity(n_B, GNN)$
5: $\quad$ **if** $nACompl$ **and not** $nBCompl$ **then**
6: $\quad\quad toDiscard \leftarrow n_B$
7: $\quad$ **if** $nBCompl$ **and not** $nACompl$ **then**
8: $\quad\quad toDiscard \leftarrow n_A$
9: $\quad$ **if not** $nACompl$ **and not** $nBCompl$ **then**
10: $\quad\quad$ **if** $\mathcal{C}_{loss}(n_A, GNN) > \mathcal{C}_{loss}(n_B, GNN)$ **then**
11: $\quad\quad\quad toDiscard \leftarrow n_B$
12: $\quad\quad$ **else if** $\mathcal{C}_{loss}(n_A, GNN) < \mathcal{C}_{loss}(n_B, GNN)$ **then**
13: $\quad\quad\quad toDiscard \leftarrow n_A$
14: $\quad\quad$ **else**
15: $\quad\quad\quad toDiscard \leftarrow random(n_A, n_B)$
16: $\quad$ **return** $toDiscard$

4.4 Norm synthesis strategy

We have so far described how to detect substitutability and complementarity relationships between norms. Furthermore, we have detailed how these relationships can be exploited to minimise the number of constraints imposed on the agents, while preserving the compactness of normative systems to the greatest possible extent. We now introduce a novel norm synthesis strategy that detects and exploits substitutability and complementarity relationships.

Algorithm 2 describes LION's norm synthesis strategy, which receives as input a transition $\langle s, A, s' \rangle$ with the system states before (s) and after (s') the performance of A; the set $C \in S$ of undesired states of the system; and the generalised normative network GNN. Initially, the strategy performs norm generation if the current state s' is undesired and LION has never generated norms aimed at avoiding it (line 2). If this is the case, it creates norms aimed at avoiding s' in the future (line 3). We borrow $normCreation$ and other basic operations such as $normApplicability$, $normEvaluation$, $generaliseUp$ or $specialiseDown$, from the SIMON algorithm [13]. Next, it retrieves those norms that have been applicable to the agents in state s (line 4), and computes concurrently applicable norms (line 5). Then, the strategy evaluates if applicable norms have succeeded in avoiding conflicts in s' (a norm is considered to perform well if it was applicable at the previous state s, and the current state s' is not undesired) (line 6). Afterwards, it classifies each applicable norm (line 9) to assess if (i) it is has performed well so far; (ii) it is *generalisable* with another norm in the normative system; (iii) it is *substitutable* with another norm in the normative network; and (iv) it is *complementary* with another norm in the normative network. Thereafter, the GNN is updated with the newly discovered substitutability and complementarity relationships (line 12). Next, the algorithm exploits norm classifications to refine the normative system. Different operations are carried out depending on the outcome of the classification:

- *Underperforming norm.* If a norm underperforms in terms of effectiveness or necessity, it must be specialised. Thus, the algorithm deactivates the norm and specialises down the norm's children in the GNN (lines 14-15). This undoes (backtracks over) all the generalisations involving the underperforming norm.
- *Generalisable norm.* If a norm is generalisable with another norm, it generalises them up in the GNN (lines 17-18).
- *Substitutable norm.* If a norm is substitutable with other norms, the algorithm retrieves all the norms it is substitutable with (lines 19-20). For each norm it is substitutable with, it invokes Algorithm 1 to choose which of them must be discarded (line 22). The chosen norm is subsequently deactivated and specialised down in the GNN (line 24).

Algorithm 2 LION's norm synthesis strategy

```
function LIONSTRATEGY(⟨s, A, s'⟩, C, GNN)
    if s' ∈ C and not regulated(s', GNN) then
        GNN ← normCreation(⟨s, A, s'⟩, GNN)
    applicableNorms ← normApplicability(s, GNN)
    concurrentNorms ← normConcurrency(s, GNN)
    performances ← normEvaluation(s', applicableNorms)
    classifiedNorms ← ∅
    for all n ∈ applicableNorms do
        n ← classify(n, performances, concurrentNorms)
        if isClassified(n) then
            classifiedNorms ← classifiedNorms ⋃{n}
            GNN ← update(GNN, n)
    for all n ∈ classifiedNorms do
        if underPerforms(n) then
            GNN ← specialiseDown(n, GNN)
        else
            if isGeneralisable(n) then
                GNN ← generaliseUp(n, GNN)
            if isSubstitutable(n) then
                substitutableNorms ← getSubstitutable(n, GNN)
                for all n' ∈ substitutableNorms do
                    discard ← chooseToDiscard(n, n', GNN)
                    if discard ≠ null then
                        GNN ← specialiseDown(discard, GNN)
    return GNN
```

The algorithm ends by returning the possibly updated GNN, which contains the normative system that will be communicated to the agents.

4.5 Analysing normative systems

Next we establish how to measure and compare normative systems in terms of the synthesis objectives pursued by LION, namely in terms of *compactness*, *liberality*, and *regulative performance*.

The literature in norms research has considered different metrics to evaluate normative systems. In particular, [7] introduces *minimality* and *simplicity* as criteria for the off-line synthesis of norms. While minimality measures the number of *constraints* in a normative system, simplicity measures the cost of agent reasoning with its norms. Based on these concepts, in [13], normative systems are evaluated in terms of their minimality and simplicity. There, the minimality of a normative system is computed as the number of norms it contains, and simplicity is assessed as the total number of norm predicates. However, we recall that a norm may compactly represent several specific norms (e.g. in Figure 2a, norm n_7 represents four specific norms n_1, n_2, n_3, n_4). Therefore, the interpretation of minimality in [13] disregards the actual number of specific norms that a normative system represents. This requires a more fine-grained measure of minimality. Thus, we define the representation minimality of a normative system Ω as $\mathcal{R}(\Omega) = |\cup_{n\in\Omega}\,\mathit{specific}(n)|$, where $\mathit{specific}(n)$ stands for the specific norms represented by norm n. From this definition it follows that the less the representation minimality of a normative system, namely the fewer the number of constraining (specific) norms, the greater the freedom for the agents. We consider that the notion of representation minimality more accurately captures the concept of minimality originally introduced in [7].

We notice that minimality and simplicity provide *compactness measures*. In order to compare whether a normative system is more liberal than another, we rely on the *"more liberal than"* relationship between normative systems introduced by Ågotnes et al. in [1]. Hence, in this work we say that a normative system Ω is more liberal than another Ω' if the norms represented by Ω are included in those represented by Ω', namely if $\cup_{n\in\Omega}\mathit{specific}(n) \subset \cup_{n'\in\Omega'}\mathit{specific}(n')$. We will also assess the substitutability of a normative system as the number of substitutability relationships that it contains. This provides a measure of the lack of liberality of a normative system: the more substitutability of a normative system, the bigger the opportunity to synthesise a more liberal one.

Finally, regarding regulative performance, we resort to the measures of effectiveness and necessity defined in [13]. These allow us to quantify how good a normative system is at regulating a MAS and how necessary its norms are for regulating.

5. EMPIRICAL EVALUATION

We now compare LION with SIMON [13] along several dimensions. We first analyse the liberality of the normative systems synthesised by both approaches, and show that the normative systems synthesised by LION are more liberal than those synthesised by SIMON. We observe that the key to LION being more liberal than SIMON is that LION manages to detect and discard, on average, 90% of the substitutability relationships within SIMON's most-frequently synthesised normative system. Thereafter, we perform a further analysis to quantify the performance of SIMON's multi-objective synthesis. We observe that the representation compactness and performance obtained by LION is similar to SIMON's. This comes at some extra cost of LION's synthesis process.

Empirical settings. Our experiments use the same scenario and experimental settings described in [13]. We run a discrete-event simulation of a traffic junction, whose agents are autonomous cars, and undesired states are those where cars collided. Each simulation employs some norm synthesis mechanism (either LION or SIMON) that monitors the simulation and synthesises norms for the cars. Every time the norm synthesis mechanism at use changes the normative system, it sends the new normative system to the cars. At each tick, each car decides whether to fulfil or infringe norms according to some norm infringement probability, which is fixed to 0.3 and is the same for all cars. To detect substitutability and complementarity, we set the minimum number of samples for the utility series of a pair of norms ($\hat{\mathcal{U}}^{FF}_{n,n'}$, $\hat{\mathcal{U}}^{FI}_{n,n'}$ and $\hat{\mathcal{U}}^{IF}_{n,n'}$) to 20 ($eval_{min} = 20$). Moreover, we set α_{sim}, the threshold employed to detect substitutability and complementarity in equations 4, 5, 6, 7, to 0.05. Finally, we set to 10 the constant factor k employed to compute the generalisation degree of a norm in equation 9.

We performed 200 simulations with LION and 200 simulations with SIMON. Each simulation started with an empty normative system. The syntax of norms is the one described in Section 2: three predicates describe the perceptions of a reference car from its three front positions (relative heading of another car, a wall, nothing, or anything). We considered that a simulation converged to a normative system if during a 5000-tick period: (i) the normative system remained unchanged; and (ii) no new conflicts (those that had not been regulated by any norm yet) occurred. Thus, conflicts arising from norm violations are disregarded when assessing convergence.

Liberality analysis. Our first comparison focuses on the liberality of the normative systems that LION and SIMON managed to synthesise upon convergence. We notice that both LION and SIMON converged in all simulations. Figure 3 graphically represents the relationship *"more liberal than"* between LION's and SIMON's normative systems. Each circle represents a different normative system. The squared figure on top of each circle stands for the number of times (out of 200 simulations) it was synthesised. White circles represent LION's normative systems, while gray circles represent SIMON's normative systems. For instance, Ω_1 is a normative system that was synthesised by SIMON 179 times. The "more liberal than" relationship is represented by the subset relationship between circles. For instance, since Ω_7 is contained in Ω_1 then we say that Ω_7 is more liberal than Ω_1.

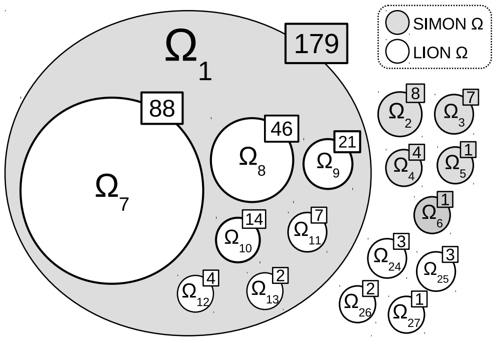

Figure 3: Normative systems synthesised by LION and SIMON.

We observe that SIMON converged to 6 different normative systems ($\Omega_1, \ldots, \Omega_6$). Specifically, 90% of the times (179 out of 200 simulations), it converged to normative system Ω_1, which corresponds to the one previously depicted in Table 1. As to LION, it synthesised 21 different normative systems ($\Omega_7, \ldots, \Omega_{27}$). Nevertheless, those normative systems where not evenly distributed. Thus, 90% of the simulations (180 out of 200) synthesised just 6 normative systems ($\Omega_7, \ldots, \Omega_{12}$), whereas the remaining normative systems ($\Omega_{13}, \ldots, \Omega_{27}$) where only synthesised by 10% of the simulations. As shown in Figure 3, 81% of LION's normative systems (from Ω_7 to Ω_{23}) are more liberal than (are contained in) Ω_1, namely SIMON's most frequent normative system[3]. 81% of LION's normative systems were synthesised in 96% of simulations, and thus we can state that LION converged to normative systems more liberal than Ω_1 for 96% of simulations. The remaining normative systems have similar (slightly better) metrics than SIMON's Ω_1.

Now we analyse why 81% of LION's normative systems are more liberal than Ω_1. With this aim, we compute the number of substitutability relationships in LION's and SIMON's normative systems. We carry out this computation by means of simulation. For each normative system, we run a simulation that performs pairwise comparison between its norms to check if they are substitutable. For each pair of norms in the normative system, the simulation proceeds by having two cars fulfil/infringe the norms, checking whether conflicts arise or not after fulfilments/infringements, and hence detecting substitutability according to Definition 5. As depicted in Figure 4, on average, LION's normative systems manage to get rid of 90% of the substitutability relationships that SIMON's Ω_1 contains. Furthermore, it managed to detect and preserve the 100% of complementary norms in SIMON's Ω_1.

Multi-objective synthesis analysis. Figure 4 summarises the average savings obtained by LION's multi-objective synthesis process with respect to SIMON. First, following the observation above, the success in detecting and discarding substitutable norms translates into a reduction in representation minimality, which is our measure to quantify the gain in liberality. More precisely, the normative systems synthesised by LION contain 21% less specific norms than those synthesised by SIMON. Therefore, LION's normative systems are less restrictive than SIMON's normative systems. Second, regarding effectiveness and necessity, LION's normative systems are similarly effective in avoiding conflicts (in fact, slightly, 2.73%, more effective), while they are more necessary (11% higher). This comes at no surprise because SIMON's normative systems contain substitutable norms. We recall that two substitutable norms satisfy the same regulatory needs, and hence only one of them (but at least one of them) is actually necessary. Third, as to compactness, although LION's and SIMON's normative systems can be considered to be equally minimal (LION is 0.1% more minimal), the first ones are 28% less simple. In other words, LION's normative systems have as many norms as SIMON's, but the norms in LION's normative systems have more predicates. This is reasonable if we consider that discarding substitutable norms involves specialising general norms, and hence leads to normative systems whose norms are more specific (and thus have a larger number of predicates). Notice though that these benefits come at some extra cost: LION requires 27.81% extra time to converge, since it performs extra tasks to capture norm synergies, as well as to deactivate substitutable norms.

[3]For clarity, Figure 3 only shows 7 out of the 17 LION's normative systems that are more liberal than Ω_1. Normative systems $\Omega_{14}, \ldots, \Omega_{23}$ are all more liberal than Ω_1 but each one is only synthesised once by LION.

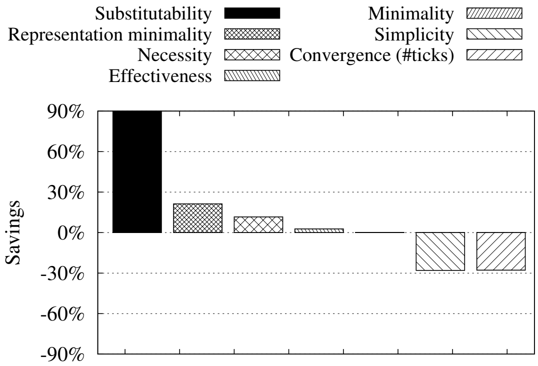

Figure 4: LION savings with respect to SIMON's most frequently synthesised normative system.

6. CONCLUSIONS AND FUTURE WORK

The aim of our work is to automatically synthesise compact, well-performing and liberal normative systems for open multi-agent systems that respect the autonomy of agents to the greatest extent possible. We argue that, on the one hand, the resulting normative system that is synthesised on-line should: i) be able to regulate agent interactions in a seamless way; ii) be as compact as possible to limit agents' norm reasoning effort; and iii) impose as few restrictions as possible, to respect the autonomy of agents to the greatest extent possible. On the other hand, we also highlight that norms within normative systems can present non-apparent synergies that have been hitherto unexplored in the literature. In this paper, in addition to considering generalisation relationships that have been proven to help to synthesise compact normative systems, we propose LION, a *LI*beral *O*n-line *N*orm mechanism that includes *substitutability* and *complementarity* norm relationships to pursue liberality without compromising efficiency.

We provided experimental evidence to assesses the quality and relevance of our proposal. In particular, we reported on experiments in which 96% of the time LION synthesised normative systems that are more liberal than those produced by the best-of-class approach, SIMON [13]. Specifically, they remove 90% of its substitutability relationships. This is accomplished without compromising minimality, effectiveness, or necessity. In future work we plan to study its application to other scenarios such as self-regulation of on-line communities as well as to employee-driven creation of best-practices for professional bodies.

Acknowledgements

Work funded by projects AT (CSD2007-0022), COR (TIN2012-38876-C02-01, TIN2012-38876-C02-02), and 2009-SGR-1434. Mike Wooldridge was supported by the ERC under Advanced Grant 291528 ("RACE").

REFERENCES

[1] T. Ågotnes, W. Van Der Hoek, C. Sierra, and M. Wooldridge. On the logic of normative systems. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI '07, pages 1175–1180, 2007.

[2] G. M. Ahlfeldt and W. Maennig. Substitutability and complementarity of urban amenities: External effects of built heritage in berlin. *Real Estate Economics*, 38(2):285–323, 2010.

[3] N. Alechina, M. Dastani, and B. Logan. Norm approximation for imperfect monitors. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '14, pages 117–124, Richland, SC, 2014. International Foundation for Autonomous Agents and Multiagent Systems.

[4] G. Boella, L. van der Torre, and H. Verhagen. Introduction to normative multiagent systems. *Computational & Mathematical Organization Theory*, 12(2-3):71–79, 2006.

[5] Y. Chou. *Statistical analysis: with business and economic applications*. Holt, Rinehart and Winston Quantitative Methods Series. Holt, Rinehart and Winston, 1969.

[6] F. Dignum. Autonomous agents with norms. *Artif. Intell. Law*, 7(1):69–79, 1999.

[7] D. Fitoussi and M. Tennenholtz. Minimal social laws. In *AAAI*, pages 26–31, 1998.

[8] H. Franks, N. Griffiths, and S. Anand. Learning agent influence in mas with complex social networks. *Autonomous Agents and Multi-Agent Systems*, 28(5):836–866, 2014.

[9] N. Griffiths and M. Luck. Norm Emergence in Tag-Based Cooperation. In *Proceedings of COIN*, 2010.

[10] G. McKenzie. Complementarity, substitutability, and independence. *Oxford Economic Papers*, 29(3):430–441, 1977.

[11] J. Morales, M. López-Sánchez, and M. Esteva. Using Experience to Generate New Regulations. In *IJCAI*, pages 307–312, 2011.

[12] J. Morales, M. Lopez-Sanchez, J. A. Rodriguez-Aguilar, M. Wooldridge, and W. Vasconcelos. Automated synthesis of normative systems. In *AAMAS 2013*, pages 483–490, 2013.

[13] J. Morales, M. Lopez-Sanchez, J. A. Rodriguez-Aguilar, M. Wooldridge, and W. Vasconcelos. Minimality and simplicity in the on-line automated synthesis of normative systems. In *AAMAS 2014*, pages 109–116, Richland, SC, 2014. IFAAMAS.

[14] J. Morales, M. Lopez-Sanchez, J. A. Rodriguez-Aguilar, M. Wooldridge, and W. Vasconcelos. Source code of minimality and simplicity in the on-line automated synthesis of normative systems. `https://github.com/NormSynthesis/NormSynthesisMachine`, 2014.

[15] A. Perotti, G. Boella, and A. d. Garcez. A framework for monitoring and learning in multiagent systems. 2014.

[16] P. A. Samuelson. Complementarity: An essay on the 40th anniversary of the hicks-allen revolution in demand theory. *Journal of Economic Literature*, pages 1255–1289, 1974.

[17] B. Savarimuthu, S. Cranefield, M. Purvis, and M. Purvis. Identifying prohibition norms in agent societies. *Artificial Intelligence and Law*, 21(1):1–46, 2013.

[18] T. Savarimuthu, S. Cranefield, M. A. Purvis, and M. K. Purvis. Obligation norm identification in agent societies. *Journal of Artificial Societies and Social Simulation*, 13(4):3, 2010.

[19] O. Sen and S. Sen. Effects of social network topology and options on norm emergence. In *Proceedings of COIN*, pages 211–222, 2010.

[20] S. Sen and S. Airiau. Emergence of norms through social learning. In *IJCAI*, pages 1507–1512, 2007.

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

[22] M. van Smoorenburg and R. van der Velden. The training of school-leavers: Complementarity or substitution? *Economics of Education Review*, 19(2):207 – 217, 2000.

[23] D. Villatoro, J. Sabater-Mir, and S. Sen. Social instruments for robust convention emergence. In *IJCAI*, pages 420–425, 2011.

[24] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2):275–309, 2013.

[25] A. J. Ward, J. A. Brown, and D. Rodriguez. Governance bundles, firm performance, and the substitutability and complementarity of governance mechanisms. *Corporate Governance: An International Review*, 17(5):646–660, 2009.